

МИНОБРНАУКИ РОССИИ

ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО  
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ  
ВЫСШЕГО ОБРАЗОВАНИЯ «ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ЭКОНОМИКИ И СЕРВИСА» В Г. АРТЕМЕ



УТВЕРЖДАЮ  
Заместитель директора  
филиала  
О.И. ИВАНЮГА

## **РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ПРАКТИКИ**

### **К ПРОФЕССИОНАЛЬНЫМ МОДУЛЯМ:**

- ПМ.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ;**
- ПМ.02 РАЗРАБОТКА И АДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ**
- ПМ.03 УЧАСТИЕ В ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ**
- ПМ.04 ВЫПОЛНЕНИЕ РАБОТ ПО ПРОФЕССИИ РАБОЧЕГО «ОПЕРАТОР ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНЫХ И ВЫЧИСЛИТЕЛЬНЫХ МАШИН»**

программы подготовки специалистов среднего звена  
*по специальности*

**09.02.03 Программирование в компьютерных системах**


Год набора на ООП  
2020

Уровень подготовки: базовый


Артем 2020

Рабочая учебная программа учебной практики к профессиональным модулям **ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем; ПМ.02 Разработка и администрирование баз данных; ПМ.03 Участие в интеграции программных модулей; ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»** разработана в соответствии с Разъяснениями по формированию примерных программ начального профессионального и среднего профессионального образования на основе Федеральных государственных образовательных стандартов НПО и СПО, утвержденными Департаментом государственной политики и нормативно - правового регулирования в сфере образования Минобрнауки РФ от 27 августа 2009 года, с учетом требований Федерального государственного образовательного стандарта среднего профессионального образования (далее – СПО), утвержденного приказом Минобрнауки РФ от 28 июля 2014 г. № 804, для освоения программы подготовки специалистов среднего звена по специальности **09.02.03 Программирование в компьютерных системах** базовой подготовки, реализуемой колледжем Филиала федерального государственного бюджетного образовательного учреждения высшего образования «Владивостокский государственный университет экономики и сервиса» в г. Артеме (далее Филиал ФГБОУ ВО «ВГУЭС» в г. Артеме).

**Разработчик:**

Место работы	Занимаемая должность, ученая степень и ученое (почетное) звание, квалификационная категория	Инициалы, фамилия	Подпись
Филиал ФГБОУ ВО «ВГУЭС» в г. Артеме	Преподаватель кафедры экономики, управления и информационных технологий	Е.В.Волошин	

**Эксперты:**

Место работы	Занимаемая должность, ученая степень и ученое (почетное) звание, квалификационная категория	Инициалы, фамилия	Подпись
ООО «СКС – Сервис», г. Артем	Директор	О.В. Бажин	
Филиал ФГБОУ ВО «ВГУЭС» в г. Артеме	Руководитель информационно-технического центра	В.В. Неслюзов	

Рабочая программа учебной практики к ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем; ПМ.02 Разработка и администрирование баз данных; ПМ.03 Участие в интеграции программных модулей; ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»

**ОДОБРЕНА**

на заседании кафедры экономики, управления и информационных технологий филиала ФГБОУ ВО «ВГУЭС» в г. Артеме

Протокол №14 от 06 мая 2020 года

И.о. зав. кафедрой ЭУИТ  А.А.Власенко

СОГЛАСОВАНА  
Зав. отделением  М.С.Словикова

Методист учебно-методической части  Т.И.Теплякова

## СОДЕРЖАНИЕ

<b>1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ</b>	стр. 4
<b>2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ УЧЕБНОЙ ПРАКТИКИ</b>	8
<b>3. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ПРАКТИКИ</b>	10
<b>4. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ</b>	17
<b>5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ПРАКТИКИ</b>	20
<b>ПРИЛОЖЕНИЕ А ГЛОССАРИЙ</b>	27
<b>ПРИЛОЖЕНИЕ Б ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ, ВНЕСЕННЫХ В РАБОЧУЮ ПРОГРАММУ</b>	34

# 1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ

## 1.1. Область применения программы

Рабочая программа учебной практики по профессиональным модулям: **ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем; ПМ.02 Разработка и администрирование баз данных; ПМ.03 Участие в интеграции программных модулей; ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»** является частью основной профессиональной образовательной программы базовой подготовки по специальности **09.02.03 Программирование в компьютерных системах** в части освоения основных видов профессиональной деятельности (ВПД): **Разработка программных модулей программного обеспечения компьютерных систем, Разработка и администрирование баз данных, Участие в интеграции программных модулей; Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»** и соответствующих профессиональных компетенций (ПК):

- ПК 1.1 Выполнять разработку спецификаций отдельных компонент.
- ПК 1.2 Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.
- ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств.
- ПК 1.4 Выполнять тестирование программных модулей.
- ПК 1.5 Осуществлять оптимизацию программного кода модуля.
- ПК 1.6 Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.
- ПК 2.1 Разрабатывать объекты базы данных
- ПК 2.2 Реализовывать базу данных в конкретной СУБД
- ПК 2.3 Решать вопросы администрирования базы данных
- ПК 2.4 Реализовывать методы и технологии защиты информации в базах данных
- ПК 3.1 Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.
- ПК 3.2 Выполнять интеграцию модулей в программную систему.
- ПК 3.3 Выполнять отладку программного продукта с использованием специализированных программных средств.
- ПК 3.4 Осуществлять разработку тестовых наборов и тестовых сценариев.
- ПК 3.5 Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования.
- ПК 3.6 Разрабатывать технологическую документацию.

Рабочая программа учебной практики может быть использована при разработке программ профессиональной подготовки и переподготовки работников в области программирования в компьютерных системах при наличии среднего (полного) общего образования, среднего или высшего образования, не отвечающего профилю данной подготовки. Опыт работы не требуется.



## **1.2. Цели и задачи учебной практики – требования к результатам освоения:**

В целях овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения программы учебной практики должен:

### **по ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем:**

#### **иметь практический опыт:**

ПО 1. Разработка проекта программного продукта согласно постановке задачи пользователя.

ПО 2. Разработки кода программного продукта на основе готовой спецификации на уровне модуля.

ПО 3. Использования инструментальных средств на этапе отладки программного продукта.

ПО 4. Проведения тестирования программного модуля по определенному сценарию.

#### **уметь:**

У 1. Осуществлять разработку кода программного модуля на современных языках программирования.

У 2. Создавать программу по разработанному алгоритму как отдельный модуль.

У 3. Выполнять отладку и тестирование программы на уровне модуля.

У 4. Оформлять документацию на программные средства.

У 5. Использовать инструментальные средства для автоматизации оформления документации.

#### **знать:**

З 1. Основные этапы разработки программного обеспечения.

З 2. Основные принципы технологии структурного и объектно-ориентированного программирования.

З 3. Основные принципы отладки и тестирования программных продуктов.

З 4. Методы и средства разработки технической документации

### **по ПМ.02 Разработка и администрирование баз данных:**

#### **иметь практический опыт:**

– работы с объектами базы данных в конкретной СУБД;

– использования средств заполнения базы данных;

– использования стандартных методов защиты объектов базы данных;

#### **уметь:**

– создавать объекты баз данных в современных СУБД и управлять доступом к этим объектам;

– работать с современными Case-средствами проектирования баз данных;

– формировать и настраивать схему базы данных;

– разрабатывать прикладные программы с использованием языка SQL;

– создавать хранимые процедуры и триггеры на базах данных;

– применять стандартные методы для защиты объектов базы данных;

#### **знать:**

– основные положения теории баз данных, хранилищ данных, баз знаний.

– основные принципы построения концептуальной, логической и физической модели данных.

– современные инструментальные средства разработки схемы базы данных.

– методы описания схем баз данных в современных СУБД.

– структуры данных СУБД, общий подход к организации представлений, таблиц, индексов и кластеров.

- методы организации целостности данных.
- способы контроля доступа к данным и управления привилегиями.
- основные методы и средства защиты данных в базах данных.
- модели и структуры информационных систем.
- основные типы сетевых топологий, приемы работы в компьютерных сетях.
- информационные ресурсы компьютерных сетей.
- технологии передачи и обмена данными в компьютерных сетях.
- основы разработки приложений баз данных.

**по ПМ.03 Участие в интеграции программных модулей:**

**иметь практический опыт:**

- участия в выработке требований к программному обеспечению;
- участия в проектировании программного обеспечения с использованием специализированных программных пакетов;

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;
- использовать методы для получения кода с заданной функциональностью и степенью качества;

**знать:**

- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;
- основные методы и средства эффективной разработки;
- основы верификации и аттестации программного обеспечения;
- концепции и реализации программных процессов;
- принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения;
- методы организации работы в коллективах разработчиков программного обеспечения;
- основные положения метрологии программных продуктов, принципы построения, проектирования и использования средств для измерений характеристик и параметров программ, программных систем и комплексов;
- стандарты качества программного обеспечения;
- методы и средства разработки программной документации.

**по ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин» :**

**иметь практический опыт:**

- работы в операционной системе WINDOWS;
- работы в основных приложениях OFFICE;

**знать:**

- основы работы в операционных системах и сервисных оболочках;
- устройство персонального компьютера, работу с его основными и периферийными устройствами;
- структуру основных папок операционной системы WINDOWS;
- основные антивирусные программы;
- основные приёмы работы с папками и файлами;
- стандартные программы операционной системы WINDOWS;
- основные программы - архиваторы;

- основные приёмы работы в локальной и глобальной сети;

**уметь:**

- запускать программы, установленные в операционной системе;
- выполнять основные операции над папками и файлами;
- выполнять поиск информации в компьютере, флеш-картах, картах памяти, оптических носителях, локальной и глобальной сети;
- удалять и устанавливать программное обеспечение;
- обновлять антивирусную программу, проверять диски на вирусы;
- сканировать и форматировать информацию;
- создавать и форматировать презентации;
- устранять сбои и ошибки, возникающие в работе программного обеспечения;
- создавать тесты в электронных оболочках;
- создавать, сохранять, модифицировать, выводить на печать документы, созданные в приложениях WINDOWS;
- изменять настройки окон приложений WINDOWS;
- осуществлять настройку операционной системы WINDOWS

**1.3. Рекомендуемое количество часов на освоение программы учебной практики:**

всего: 360 часов, в том числе:

ПМ.01 – 108 часов;

ПМ.02 – 108 часов;

ПМ.03 – 72 часа;

ПМ.04 – 108 часов;

## 2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ

Результатом освоения программы учебной практики является овладение обучающимися видами профессиональной деятельности (ВПД): **Разработка программных модулей программного обеспечения компьютерных систем, Разработка и администрирование баз данных, Участие в интеграции программных модулей; Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»** , в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	Наименование результата обучения
ПК 1.1.	Выполнять разработку спецификаций отдельных компонент.
ПК 1.2.	Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.
ПК 1.3.	Выполнять отладку программных модулей с использованием специализированных программных средств.
ПК 1.4.	Выполнять тестирование программных модулей.
ПК 1.5.	Осуществлять оптимизацию программного кода модуля.
ПК 1.6.	Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.
ПК 2.1	Разрабатывать объекты базы данных.
ПК 2.2	Реализовывать базу данных в конкретной СУБД.
ПК 2.3	Решать вопросы администрирования базы данных.
ПК 2.4	Реализовывать методы и технологии защиты информации в базах данных.
ПК 3.1	Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.
ПК 3.2	Выполнять интеграцию модулей в программную систему.
ПК 3.3	Выполнять отладку программного продукта с использованием специализированных программных средств.
ПК 3.4	Осуществлять разработку тестовых наборов и тестовых сценариев.
ПК 3.5	Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования.
ПК 3.6	Разрабатывать технологическую документацию.
ОК 1.	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес
ОК 2.	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
ОК 3.	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
ОК 4.	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
ОК 5.	Владеть информационной культурой, анализировать и оценивать информацию с использованием информационно-коммуникационных технологий.
ОК 6.	Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7.	Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.
ОК 8.	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
ОК 9.	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

### 3. СТРУКТУРА И СОДЕРЖАНИЕ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ

#### 3.1. Тематический план программы учебной практики

<b>Коды профессиональных компетенций</b>	<b>Наименование профессиональных модулей</b>	<b>Всего часов</b>	<b>Распределение часов по семестрам</b>
1	2	3	4
ПК 1.1, ПК 1.4, ПК 3.3	ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»	108	4
ПК 1.1. – ПК 1.6.	ПМ.01. Разработка программных модулей программного обеспечения компьютерных систем	108	5
ПК 2.1. – ПК 2.4.	ПМ.02 Разработка и администрирование баз данных	108	5
ПК 3.1 – ПК 3.6	ПМ.03 Участие в интеграции программных модулей	72	7
	<b>Всего:</b>	<b>396</b>	

### 3.2. Содержание обучения по учебной практике

Наименование профессионального модуля (ПМ), МДК и тем учебной практики	Содержание учебного материала	Объем часов (с указанием их распределения по семестрам)
1	2	3
	<b>УЧЕБНАЯ ПРАКТИКА</b>	<b>396</b>
	<b>ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»</b> <b>УП.04.01. Учебная практика к МДК.04.01 Программное обеспечение ЭВМ</b> <i>4 семестр</i>	<b>108</b>
<b>Тема 1. Вводное занятие</b>	<b>Виды работ</b> Инструктаж о прохождении практики. Знакомство с программой практики и порядком её проведения; изучение правил внутреннего распорядка; знакомство с графиком работы студентов, рекомендациями по ведению дневника практики, составлению отчета. Инструктаж по технике безопасности, пожаробезопасности, производственной санитарии под роспись в журнале. Правила безопасности при работе с компьютером.	6
<b>Тема 2. Текстовый редактор Microsoft Word</b>	<b>Виды работ</b> Установка офисного программного обеспечения, настройка интерфейса текстового редактора Microsoft Word	6
<b>Тема 3. Панели инструментов</b>	<b>Виды работ</b> Подключение основных панелей инструментов, их настройка, описание и назначение, форматирование текстовых, табличных, графических и смешанных документов	6
<b>Тема 4. Обмен информацией</b>	<b>Виды работ</b> Передача информации в другие программы, получение информации из внешних источников	6
<b>Тема 5. Ссылки</b>	<b>Виды работ</b> Выполнение пробных работ. Создание гиперссылок. Связь с другими документами.	6
<b>Тема 6. Приобретение опыта по обслуживанию очереди вывода на печать в WINDOWS, удаление из очереди ненужных файлов. Сканер</b>	<b>Виды работ</b> Выполнение практического задания. Подготовка и распечатка документов на принтере. Сканирование документов.	6

Тема 7. Работа с текстовыми документами	<b>Виды работ:</b> Установка офисного программного обеспечения.	6
	<b>Виды работ:</b> Настройка интерфейса текстового редактора Microsoft Word.	12
	<b>Виды работ:</b> Подключение основных панелей инструментов, их настройка, описание и назначение.	18
Тема 8. Работа со смешанными документами	<b>Виды работ:</b> Форматирование текстовых, табличных, графических и смешанных документов. Передача информации в другие программы.	24
Тема 9. Подготовка отчётов о прохождении практики. Защита практики.	<b>Виды работ:</b> Выполнение индивидуального задания по итогам практики	6
	<b>Виды работ:</b> Подготовка отчётов прохождения практики.	4
	<b>Виды работ:</b> Защита практики.	2
	<b>ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем</b>	<b>108</b>
	<b>УП.01.01. Учебная практика к МДК.01.01. Системное программирование</b>	
	<b>5 семестр</b>	
Тема 1. Вводное занятие	<b>Виды работ:</b> Инструктаж о прохождении практики. Знакомство с программой практики и порядок её проведения, изучение правил внутреннего распорядка, знакомство с графиком работы студентов, ведения дневника практики, составление отчета. Инструктаж по технике безопасности, пожаробезопасности, производственной санитарии под роспись в журнале. Правила безопасности при работе с компьютером.	6
Тема 2. Разработка алгоритма поставленной задачи и реализация его средствами автоматизированного проектирования.	<b>Виды работ:</b> Анализ поставленной задачи. Выбор методов и разработка основных алгоритмов решения задачи. Разработка технического задания.  Примерные темы заданий для выполнения практической работы: <ul style="list-style-type: none"><li>– Обработка сообщений</li><li>– Рисование геометрических фигур в окне</li><li>– Вывод текста</li></ul>	6



	<ul style="list-style-type: none"> <li>– Диалог с пользователем</li> <li>– Чтение и запись файлов в библиотеке Win32 API</li> <li>– Диалоговые окна</li> <li>– Растровая графика</li> <li>– Анимация</li> <li>– Библиотеки динамической компоновки DLL</li> </ul>	
	<p><b>Виды работ:</b> Разработка структуры и конкретных компонент разрабатываемого программного обеспечения, в том числе схемы алгоритмов, их общее описание. Определение свойств входных и выходных данных поставленной задачи.</p>	6
	<p><b>Виды работ:</b> Анализ процесса обработки информации и выбор структур данных для её хранения</p>	6
<b>Тема 3. Разработка кода программного продукта на основе готовой спецификации на уровне модуля.</b>	<p><b>Виды работ:</b> Выбор технологии и среды программирования. Проектирование интерфейса пользователя. Проектирование классов предметной области. Организация обработки сообщений.</p>	6
	<p><b>Виды работ:</b> Использование директив препроцессора. Организация диалога с пользователем. Тестирование элементов управления.</p>	6
	<p><b>Виды работ:</b> Работа с панелями инструментов. Разработка форм ввода-вывода информации. Чтение и запись файлов в библиотеке. Организация работы с файлами.</p>	6
	<p><b>Виды работ:</b> Разработка кода программного продукта на языке C++ на уровне модуля.</p>	6
<b>Тема 4. Использование инструментальных средств на этапе отладки программного продукта.</b>	<p><b>Виды работ:</b> Выбор стратегии тестирования и разработка тестов. Отладка кода программного продукта, используя возможности отладчика.</p>	6
	<p><b>Виды работ:</b> Использование средств отладки, предоставляемых интерфейсом пользователя. Определение мест программы, в которых необходимо установить точки останова.</p>	6
	<p><b>Виды работ:</b> Использование команд меню Debug, Go для анализа значения переменных.</p>	6

<b>Тема 5. Проведение тестирования программного модуля по определенному сценарию.</b>	<b>Виды работ:</b> Ручной контроль программного модуля. Проверка структуры программного модуля.	6
	<b>Виды работ:</b> Структурное тестирование. Устранение утечки памяти. Исследование возможных причин утечки памяти.	6
	<b>Виды работ:</b> Функциональное тестирование. Особенности отладки приложений, использующих шаблоны функций и классов.	6
	<b>Виды работ:</b> Оценочное тестирование программного продукта. Анализ соответствия разработанного программного продукта постановке задачи.	6
<b>Тема 6. Оформление документации на программные средства.</b>	<b>Виды работ:</b> Составление программной документации. Определение сведений, необходимых для сопровождения и эксплуатации программного продукта. Разработка пояснительной записки, содержащей информацию о структуре и конкретных компонентах программного обеспечения, в том числе схемы алгоритмов, их общее описание, обоснование принятых технических решений.	6
	<b>Виды работ:</b> Разработка руководства системного программиста, содержащего сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения.	6
	<b>Виды работ:</b> Разработка руководства по техническому обслуживанию, содержащего сведения для применения тестовых и диагностических программ при обслуживании технических средств.	6
	<b>ПМ.02 Разработка и администрирование баз данных</b>	<b>108</b>
	<b>УП.02.01. Учебная практика к МДК 02.01. Инфокоммуникационные системы и сети</b>	
	<b>5 семестр</b>	
<b>Тема 1. Вводное занятие</b>	<b>Виды работ:</b> Инструктаж о прохождении практики. Знакомство с программой практики и порядок её проведения, изучение правил внутреннего распорядка, знакомство с графиком работы студентов, ведения дневника практики, составление отчета. Инструктаж по технике безопасности, пожаробезопасности, производственной санитарии под роспись в журнале. Правила безопасности	6

	при работе с компьютером.	
<b>Тема 2. Основы построения сети.</b>	<b>Виды работ:</b> Выполнение работ по изучению различных видов кабелей и их подсоединение (витая пара, оптоволоконный). Изучение топологии локальной организации. Работа с аппаратурой передачи данных (сетевые адаптеры, модемы). Организация и настройка локальной сети отдела. Удаление сетевого программного обеспечения. Своевременное обновление сетевого программного обеспечения. Выполнение пробных практических работ	12
<b>Тема 3. Беспроводные технологии передачи данных</b>	<b>Виды работ:</b> Работа по организации беспроводной связи по стандарту Bluetooth. Работа в беспроводных локальных сетях Выполнение пробных практических работ	18
<b>Тема 4. Стек коммуникационных протоколов TCP/IP</b>	<b>Виды работ:</b> Работа по реализации межсетевого взаимодействия средствами TCP/IP. Установка и настройка программного обеспечения для работы локальной сети. Выполнение пробных практических работ	12
<b>Тема 5. Локальные вычислительные сети Проектирование компьютерных сетей</b>	<b>Виды работ:</b> Маршрутизация пакетов в IP сетях. Проектирование локальной сети отдела Работы по созданию общих ресурсов в локальной сети и управление ими Выполнение пробных практических работ	12
<b>Тема 7. Настройка домена и его безопасность</b>	<b>Виды работ:</b> Работа с системой доменных имен DNS. Настройка сетевых принтеров и факсов. Проектирование учетной политики организации. Назначение прав доступа. Выполнение пробных практических работ	12
<b>Тема 8. Обеспечение компьютерной безопасности в информационных системах и сетях. Защита информации в инфокоммуникационных системах и сетях.</b>	<b>Виды работ:</b> Работа с портами. Составление отчетов по сетевой активности в организации. Работа по антивирусной защите. Выполнение пробных практических работ	18
<b>Тема 9. Заключительное занятие</b>	<b>Виды работ:</b> 1. Выполнение итоговых индивидуальных практических работ	4
	2. Составление отчета по практике. Заполнение дневника по практике. Работа с литературой по профилю специальности. Заполнение дневника по практике. Заполнение дневника по практике. Работа с литературой по профилю специальности. Защита практики	14

	<b>ПМ.03 Участие в интеграции программных модулей</b>	<b>72</b>
	<b>УП.03.01. Учебная практика к МДК 03.01. Технология разработки программного обеспечения</b>	
	<i>7 семестр</i>	
<b>Тема 1. Вводное занятие</b>	<b>Виды работ:</b> Инструктаж о прохождении практики. Знакомство с программой практики и порядок её проведения, изучение правил внутреннего распорядка, знакомство с графиком работы студентов, ведения дневника практики, составление отчета. Ознакомление с порядком сбора информации. Инструктаж по технике безопасности, пожаробезопасности, производственной санитарии под роспись в журнале. Правила безопасности при работе с компьютером. Выбор задания.	<b>6</b>
<b>Тема 2. Работа приложения с рабочим столом</b>	<b>Виды работ:</b> Область уведомлений. Виджет экрана. Класс сервиса рабочего стола.	<b>18</b>
<b>Тема 3. Работа с файлами, каталогами и потоками ввода-вывода</b>	<b>Виды работ:</b> Наблюдение за файлами и каталогами. Потоки ввода-вывода. Класс QTextStream. Класс QDataStream.	<b>24</b>
<b>Тема 4. Создание гибридных приложений. Браузер WebKit.</b>	<b>Виды работ:</b> Написание простого Web-браузера. Ввод адресов. Управление историей. Загрузка страниц и ресурсов.	<b>24</b>
<b>Всего:</b>		<b>396</b>

## 4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ

### 4.1. Требования к минимальному материально-техническому обеспечению

Реализация программы учебной практики к модулям **ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем, ПМ.02 Разработка и администрирование баз данных, ПМ.03 Участие в интеграции программных модулей, ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»** предполагает наличие учебных лабораторий: «Системного и прикладного программирования», «Информационно-коммуникационных систем».

#### **Оборудование лаборатории и рабочих мест лаборатории «Системного и прикладного программирования»:**

- компьютеры по количеству обучающихся;
- интерактивная доска или мультимедиа проектор;
- локальная сеть.

#### **Технические и программные средства обучения:**

- операционная система Windows XP, Vista,7;
- среда Visual Studio 2010 Release Candidate или Visual Studio 2005/2008;
- язык программирования Microsoft VisualC++;
- средств разработки Qt by Digiav4.8.5. (MinGW OpenSource);
- пакет прикладных программ Microsoft Office;
- программы-архиваторы.

#### **Оборудование лаборатории и рабочих мест лаборатории «Информационно-коммуникационных систем»:**

- компьютеры по количеству обучающихся;
- интерактивная доска или мультимедиа проектор.
- локальная сеть.

#### **Технические и программные средства обучения:**

- Операционная система Windows XP, Vista,7;
- программное обеспечение общего назначения;
- пакет прикладных программ Microsoft Office;
- программы-архиваторы.

### 4.2 Информационное обеспечение обучения

#### **Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы**

##### **Основная литература:**

1. Брайен, Керниган В. Язык программирования C : учебник / Брайен Керниган В., Деннис Ричи М. — Москва : Интуит НОУ, 2016. — 313 с. — URL: <https://book.ru/book/918294>
2. Александров, Э.Э. Программирование на языке C в Microsoft Visual Studio 2010 : курс лекций / Александров Э.Э., Афонин В.В. — Москва : Интуит НОУ, 2016. — 570 с. — URL: <https://book.ru/book/918122>
3. Алексеев, Е.Р. Программирование на языке C++ в среде Qt Creator : курс лекций / Алексеев Е.Р., Злобин Г.Г., Костюк Д.А., Чеснокова О.В., Чмыхало А.С. — Москва : Интуит НОУ, 2016. — 715 с. — URL: <https://book.ru/book/918128>
4. Построение сетей на базе коммутаторов и маршрутизаторов : курс лекций / — Москва : Интуит НОУ, 2016. — 330 с. — URL: <https://book.ru/book/917981>

5. Смирнова, Е.В. Построение коммутируемых компьютерных сетей : курс лекций / Смирнова Е.В., Баскаков И.В., Пролетарский А.В., Федотов Р.А. — Москва : Интуит НОУ, 2016. — 428 с. — URL: <https://book.ru/book/917979>
6. Кумскова, И.А. Базы данных : учебник / Кумскова И.А. — Москва : КноРус, 2020. — 400 с. — (СПО). — ISBN 978-5-406-07467-1. — URL: <https://book.ru/book/932493>
7. Грошев, А.С. Основы работы с базами данных : курс лекций / Грошев А.С. — Москва : Интуит НОУ, 2016. — 255 с. — URL: <https://book.ru/book/917933>
8. Пакулин, В.Н. 1С:Бухгалтерия 8. 1 : учебное пособие / Пакулин В.Н. — Москва : Интуит НОУ, 2016. — 67 с. — URL: <https://book.ru/book/917496>
9. Заика, А.А. Практика бухгалтерского учета в 1С:Бухгалтерии 8 : курс лекций / Заика А.А. — Москва : Интуит НОУ, 2016. — 526 с. — URL: <https://book.ru/book/917985>
10. Угринович, Н.Д. Информатика : учебник / Угринович Н.Д. — Москва : КноРус, 2020. — 377 с. — (СПО). — ISBN 978-5-406-07314-8. — URL: <https://book.ru/book/932057>
11. Прохорский, Г.В. Информатика : учебное пособие / Прохорский Г.В. — Москва : КноРус, 2020. — 240 с. — (СПО). — ISBN 978-5-406-07612-5. — URL: <https://book.ru/book/936152>

***Электронные ресурсы:***

1. ЭБС ИЗДАТЕЛЬСТВА "BOOK.RU" КОЛЛЕКЦИЯ СПО <https://www.book.ru/>
2. ЭБС ИЗДАТЕЛЬСТВА "ЮРАЙТ" <https://urait.ru>
3. ЭБС ИЗДАТЕЛЬСТВА "ЛАНЬ" <https://e.lanbook.com>

***Дополнительная литература:***

1. Макарова, Н.В. Основы программирования : учебник / Макарова Н.В., Нилова Ю.Н., Зеленина С.Б., Лебедева Е.В. — Москва : КноРус, 2021. — 451 с. — ISBN 978-5-406-03394-4. — URL: <https://book.ru/book/936582>
2. Фридман, А.Л. Язык программирования С : курс лекций / Фридман А.Л. — Москва : Интуит НОУ, 2016. — 218 с. — ISBN 978-5-9556-0017-8. — URL: <https://book.ru/book/918295>
3. Златопольский, Д.М. Программирование: типовые задачи, алгоритмы, методы : учебное пособие / Златопольский Д.М. 4-е изд. — Москва : Лаборатория знаний, 2020. — 224 с. — ISBN 978-5-00101-789-9. — URL: <https://book.ru/book/936428>
4. Основы построения объединенных сетей по технологиям CISCO : курс лекций / — Москва : Интуит НОУ, 2016. — 285 с. — URL: <https://book.ru/book/917875>
5. Осипов, Д.Л. Системы управления базами данных : практикум / Осипов Д.Л., Огур М.Г., сост. — Ставрополь : Северо-Кавказский федеральный университет, 2017. — 148 с. — URL: <https://book.ru/book/929919>
6. Златопольский, Д.М. Программирование: типовые задачи, алгоритмы, методы : учебное пособие / Златопольский Д.М. 4-е изд. — Москва : Лаборатория знаний, 2020. — 224 с. — ISBN 978-5-00101-789-9. — URL: <https://book.ru/book/936428>
7. Попов, А.А. Эргономика пользовательских интерфейсов в информационных системах : учебное пособие / Попов А.А. — Москва : КноРус, 2020. — 304 с. — ISBN 978-5-406-07634-7. — URL: <https://book.ru/book/935936>
8. 1С: Бухгалтерия предприятия 8. 2 : практическое пособие / Селищев Н.В., под ред. — Москва : КноРус, 2020. — 385 с. — ISBN 978-5-406-07219-6. — URL: <https://book.ru/book/932749>
9. Операционная система Microsoft Windows XP : курс лекций / — Москва : Интуит НОУ, 2016. — 374 с. — URL: <https://book.ru/book/917813>
10. Назаров, С.В. Современные операционные системы : курс лекций / Назаров С.В., Широков А.И. — Москва : Интуит НОУ, 2016. — 351 с. — ISBN 978-5-9963-0416-5. — URL: <https://book.ru/book/918225>

11. Угринович, Н.Д. Информатика. Практикум : учебное пособие / Угринович Н.Д. — Москва : КноРус, 2020. — 264 с. — (СПО). — ISBN 978-5-406-07320-9. — URL: <https://book.ru/book/932058>

**Информационное обеспечение обучения требует наличия залов:** библиотеки, читального зала с выходом в сеть Интернет.

#### **4.3. Общие требования к организации образовательного процесса учебной практики**

Освоение обучающимися профессионального модуля проходит в условиях созданной образовательной среды как в учебном заведении, так и в организациях, соответствующих профилю специальности изучаемых модулей ПМ.01, ПМ.02, ПМ.03, ПМ.04

Освоению программы учебной практики предшествует изучение общепрофессиональных дисциплин: «Системное программирование», «Инфокоммуникационные системы и сети», «Технология разработки программного обеспечения», «Программное обеспечение ЭВМ», МДК профессиональных модулей:

**ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем,**

**ПМ.02 Разработка и администрирование баз данных,**

**ПМ.03 Участие в интеграции программных модулей,**

**ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин» .**

#### **4.4. Кадровое обеспечение учебной практики образовательного процесса**

**Требования к квалификации педагогических кадров, обеспечивающих руководство учебной практикой по профессиональным модулям ПМ.01, ПМ.02, ПМ.03, ПМ.04:**

- наличие высшего образования;
- опыт деятельности в соответствующей профессиональной сфере;
- стажировка – 1 раз в три года.

Волошин Евгений Владимирович окончил ФГБОУ ВО «ВГУЭС» в г. Артеме в 2016 году, бакалавр по направлению 09.03.03 Прикладная информатика. Стаж работы – 2 г.9 мес. Преподаватель кафедры экономики, управления и информационных технологий.

Ематина Надежда Игоревна: окончила Владивостокский государственный университет экономики и сервиса в 2005 году, квалификация: техник. В 2010 году окончила Владивостокский государственный университет экономики и сервиса, квалификация: менеджер. Прошла курсы повышения квалификации 02.05.17-10.05.17 - по теме «Профессиональная деятельность преподавателя среднего профессионального образования в условиях внедрения ФГОС четвертого поколения», г. Владивосток, ГАУ ДПО «Приморский краевой институт развития образования». Стаж работы – 14 лет. Преподаватель кафедры экономики, управления и информационных технологий.

## 5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ПРАКТИКИ

### 5.1 Контроль и оценка сформированности профессиональных компетенций обучающихся

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
ПК 1.1. Выполнять разработку спецификаций отдельных компонент.	Точность определения основных этапов разработки программного обеспечения. Правильность применения основных принципов технологии структурного и объектно-ориентированного программирования. Правильность оформления документации на программные средства. Правильность и точность разработки алгоритма поставленной задачи.	Текущий контроль в форме: - защиты лабораторных заданий; - контрольных работ по темам МДК. Зачеты по производственной практике и по каждому из разделов профессионального модуля.
ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.	Правильность применения основных принципов технологии структурного и объектно-ориентированного программирования. Правильность и точность разработки кода программного модуля на современных языках программирования. Точность создания программы по разработанному алгоритму как отдельного модуля. Правильность разработки кода программного продукта на основе готовой спецификации на уровне модуля.	Текущий контроль в форме: - защиты лабораторных и практических занятий; - контрольных работ по темам МДК. Зачеты по каждому из разделов междисциплинарного курса.
ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.	Правильность применения основных принципов отладки и тестирования программных продуктов. Точность использования инструментальных средств на этапе отладки программного продукта. Правильность отладки и тестирования программы на уровне модуля.	Текущий контроль в форме: - защиты лабораторных и практических занятий; - контрольных работ по темам МДК. Зачеты по производственной практике и по каждому из разделов МДК.
ПК 1.4. Выполнять тестирование программных модулей.	Проведение тестирования программного модуля по разработанному сценарию. Правильность выполнения отладки и тестирование программы на	Текущий контроль в форме: - защиты лабораторных и практических занятий; - контрольных работ по темам МДК.



	уровне модуля.	.
ПК 1.5. Осуществлять оптимизацию программного кода модуля.	Точность проведения оптимизации программного кода модуля по определенному сценарию. Правильность выполнения отладки и тестирование программы на уровне модуля. Правильность использования инструментальных средств на этапе отладки программного продукта.	Текущий контроль в форме: - защиты лабораторных занятий; - контрольных работ по темам МДК.
ПК 1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.	Правильность использования инструментальных средств для автоматизации оформления документации. Правильность определения и использования методов и средств разработки технической документации.	Текущий контроль в форме: - защиты лабораторных занятий; - контрольных работ по темам МДК. .
ПК 2.1 Разрабатывать объекты базы данных	Точность определения основных этапов разработки программного обеспечения. Правильность применения основных принципов технологии структурного и объектно-ориентированного программирования. Правильность оформления документации на программные средства. Правильность и точность разработки алгоритма поставленной задачи	Текущий контроль в форме: защиты лабораторных заданий, контрольных работ, внеаудиторных самостоятельных работ по темам МДК модуля; Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);
ПК 2.2 Реализовывать базу данных в конкретной СУБД	Правильность применения основных принципов технологии структурного и объектно-ориентированного программирования. Правильность и точность разработки кода программного модуля на современных языках программирования. Точность создания программы по разработанному алгоритму как отдельного модуля. Правильность разработки кода программного продукта на основе готовой спецификации на уровне модуля.	Текущий контроль в форме: защиты лабораторных заданий, контрольных работ, внеаудиторных самостоятельных работ по темам МДК модуля; Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);
ПК 2.3 Решать вопросы администрирования базы данных	Правильность применения основных принципов отладки и тестирования программных	Текущий контроль в форме: защиты лабораторных заданий, контрольных работ,

	<p>продуктов.</p> <p>Точность использования инструментальных средств на этапе отладки программного продукта.</p> <p>Правильность отладки и тестирования программы на уровне модуля.</p>	<p>внеаудиторных самостоятельных работ по темам МДК модуля;</p> <p>Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);</p>
<p>ПК 2.4 Реализовывать методы и технологии защиты информации в базах данных</p>	<p>Проведение тестирования программного модуля по разработанному сценарию.</p> <p>Правильность выполнения отладки и тестирование программы на уровне модуля.</p>	<p>Текущий контроль в форме: защиты лабораторных заданий, контрольных работ, внеаудиторных самостоятельных работ по темам МДК модуля;</p> <p>Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);</p>
<p>ПК 3.1 Разрабатывать объекты базы данных</p>	<p>Точность определения основных этапов разработки программного обеспечения.</p> <p>Правильность применения основных принципов технологии структурного и объектно-ориентированного программирования.</p> <p>Правильность оформления документации на программные средства.</p> <p>Правильность и точность разработки алгоритма поставленной задачи.</p>	<p>Текущий контроль в форме: защиты лабораторных заданий, контрольных работ, внеаудиторных самостоятельных работ по темам МДК модуля;</p> <p>Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);</p>
<p>ПК 3.2 Реализовывать базу данных в конкретной СУБД</p>	<p>Правильность применения основных принципов технологии структурного и объектно-ориентированного программирования.</p> <p>Правильность и точность разработки кода программного модуля на современных языках программирования.</p> <p>Точность создания программы по разработанному алгоритму как отдельного модуля.</p> <p>Правильность разработки кода программного продукта на основе готовой спецификации на уровне модуля.</p>	<p>Текущий контроль в форме: защиты лабораторных заданий, контрольных работ, внеаудиторных самостоятельных работ по темам МДК модуля;</p> <p>Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);</p>
<p>ПК 3.3 Решать вопросы администрирования</p>	<p>Правильность применения основных принципов отладки и тестирования программных</p>	<p>Текущий контроль в форме: защиты лабораторных заданий, контрольных работ,</p>

базы данных	продуктов. Точность использования инструментальных средств на этапе отладки программного продукта. Правильность отладки и тестирования программы на уровне модуля.	внеаудиторных самостоятельных работ по темам МДК модуля; Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);
ПК 3.4 Реализовывать методы и технологии защиты информации в базах данных	Проведение тестирования программного модуля по разработанному сценарию. Правильность выполнения отладки и тестирование программы на уровне модуля.	Текущий контроль в форме: защиты лабораторных заданий, контрольных работ, внеаудиторных самостоятельных работ по темам МДК модуля; Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);
ПК 3.5 Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования.	Проверка навыков инспектирования компонент программного продукта на предмет соответствия стандартам кодирования.	Текущий контроль в форме: защиты лабораторных заданий, контрольных работ, внеаудиторных самостоятельных работ по темам МДК модуля; Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);
ПК 3.6 Разрабатывать технологическую документацию	Проверка умения разрабатывать технологическую документацию	Текущий контроль в форме: защиты лабораторных заданий, контрольных работ, внеаудиторных самостоятельных работ по темам МДК модуля; Проведения зачётных пробных заданий в период прохождения учебной и производственной практики (по профилю специальности);

## 5.2. Контроль и оценка результатов развития общих компетенций и обеспечивающих их умений.

Таблица 5.2 – Контроль и оценка результатов развития общих компетенций и обеспечивающих их умений

Результаты (освоенные общие компетенции)	Основные показатели результатов подготовки	Формы и методы контроля
ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес	Демонстрация устойчивого интереса к будущей профессии	<p>Экспертное наблюдение и оценка деятельности обучающегося в процессе освоения образовательной программы на практических занятиях, при выполнении работ по учебной и производственной практике.</p> <p>Экспертное наблюдение и оценка активности учащегося при проведении учебно-воспитательных мероприятий профессиональной направленности.</p>
ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество	Мотивированное обоснование выбора и применения методов и способов решения профессиональных задач при осуществлении выполнения задания. Точность, правильность и полнота выполнения профессиональных задач	<p>Экспертное наблюдение и оценка деятельности обучающегося в процессе освоения образовательной программы на практических занятиях, при выполнении работ по учебной и производственной практике.</p>
ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность	Демонстрация способности принимать оперативные решения в стандартных и нестандартных ситуациях, нести за них ответственность при выполнении задания.	<p>Наблюдение и оценка результатов деятельности обучающегося в процессе освоения образовательной программы на практических занятиях, при выполнении индивидуальных домашних заданий.</p> <p>Наблюдение и оценка активности учащихся при проведении учебно-воспитательных мероприятий профессиональной направленности.</p>
ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития	Оперативность поиска и использования необходимой информации для качественного выполнения профессиональных заданий, профессионального и личностного развития	<p>Наблюдение и оценка деятельности обучающегося в процессе освоения образовательной программы на практических занятиях, при выполнении индивидуальных домашних заданий, работ по учебной и производственной практике</p>
ОК 5. Владеть информационной культурой, анализировать и оценивать информацию с использованием информационно-коммуникационных технологий	Демонстрация умения оперативно осуществлять операции, предлагаемые преподавателем, делать анализ и давать оценку полученной информации, в т.ч. и с использованием	<p>Экспертное наблюдение и оценка деятельности обучающегося в процессе освоения образовательной программы на практических занятиях, в ходе компьютерного тестирования, подготовки электронных презентаций, при выполнении индивидуальных домашних заданий.</p>

<b>Результаты (освоенные общие компетенции)</b>	<b>Основные показатели результатов подготовки</b>	<b>Формы и методы контроля</b>
	программного обеспечения	Наблюдение и оценка использования учащимися информационных технологий при подготовке и проведении учебно-воспитательных мероприятий различной тематики.
ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями	Коммуникабельность при взаимодействии с обучающимися и преподавателями в ходе обучения. Воспитание уважения к мнению сокурсников.	<p>Экспертное наблюдение и оценка коммуникативной деятельности обучающегося в процессе освоения образовательной программы на практических занятиях, при выполнении индивидуальных домашних заданий.</p> <p>Наблюдение и оценка использования учащимися коммуникативных методов и приемов при подготовке и проведении учебно-воспитательных мероприятий различной тематики.</p>
ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий	Ответственность за результат выполнения задания. Способность к самоанализу и коррекции результатов собственной работы.	<p>Экспертное наблюдение и оценка деятельности обучающегося в процессе освоения образовательной программы на практических занятиях при работе в малых группах.</p> <p>Экспертное наблюдение и оценка уровня ответственности учащегося при подготовке и проведении учебно-воспитательных мероприятий различной тематики.</p> <p>Экспертное наблюдение и оценка динамики достижений учащегося в учебной и общественной деятельности.</p>
ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации	Способность к организации и планированию самостоятельных занятий при изучении дисциплины. Демонстрация потребности в получении дополнительных знаний, возможностей самореализации.	<p>Экспертное наблюдение и оценка использования учащимися методов и приемов личной организации в процессе освоения образовательной программы на практических занятиях, при выполнении индивидуальных домашних заданий.</p> <p>Экспертное наблюдение и оценка использования учащимися методов и приемов личной организации при подготовке и проведении учебно-воспитательных мероприятий различной тематики.</p> <p>Экспертное наблюдение и оценка динамики достижений учащихся в учебной и общественной деятельности.</p>
ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности	Проявление интереса к инновациям в области профессиональной деятельности	Экспертное наблюдение и оценка деятельности учащегося в процессе освоения образовательной программы на практических занятиях, при выполнении индивидуальных домашних заданий.

### 5.3. Оценка индивидуальных образовательных достижений по результатам текущего контроля и промежуточной аттестации

Критерии оценки качества выполнения учебной практики	Качественная оценка индивидуальных практических достижений	
	балл (отметка)	вербальный аналог
91 - 100	5	отлично
76 - 90	4	хорошо
61 - 75	3	удовлетворительно
менее 61	2	неудовлетворительно

На этапе аттестации по итогам выполнения программы учебной практики комиссией определяется интегральная оценка освоенных обучающимися профессиональных и общих компетенций как результат приобретения практического опыта по одному из видов профессиональной деятельности.

## 6. ГЛОССАРИЙ ОСНОВНЫХ ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ, ВСТРЕЧАЮЩИХСЯ В ПЕРИОД ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ

<b>Алгоритм</b>	Формально описанная последовательность действий, которые необходимо выполнить для получения требуемого результата
<b>Выражение</b>	Все вычисления в программе записываются в виде выражений
<b>Диалог</b>	Регламентированный обмен информацией между человеком и компьютером
<b>Запись</b>	Структура данных, состоящая из фиксированного числа компонент
<b>Консоль</b>	В различных программах и играх консолью стали называть окно для вывода системных сообщений и приёма команд.
<b>Меню</b>	Выводится на экран при записи программы
<b>Модуль</b>	Автономно компилируемая коллекция программных ресурсов
<b>Библиотека Win32 API</b>	Библиотека программного интерфейса приложений
<b>Классы</b>	Экземпляры определённого типа свойств, образующие иерархию с наследованием свойств.
<b>Объекты</b>	Элементы конкретной предметной области.
<b>Объектно-ориентированное программирование</b>	Технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является классом.
<b>Отладка</b>	Процесс локализации и исправления ошибок, обнаруженных при тестировании программного продукта.
<b>Сообщение</b>	Порция информации, участвующая в диалоговом обмене.
<b>Среда программирования</b>	Программный комплекс, который включает специализированный текстовый редактор, встроенные компилятор, компоновщик, отладчик, справочную систему.
<b>Проектирование</b>	Детальная проработка последовательности действий будущей программы
<b>IP адрес (IP address)</b>	Уникальный адрес, который присваивается каждому компьютеру, получающему доступ в Интернет. IP адрес представляет собой последовательность, состоящую из четырех чисел.
<b>Антивирусное программное обеспечение (Anti-virus software)</b>	Программное обеспечение, которое установлено на вашем компьютере и предназначено для проверки всех входящих и исходящих файлов на наличие вирусов.
<b>Аппаратное обеспечение (Hardware)</b>	Физическое оборудование, составляющее вашу сеть.
<b>Архитектура (Architecture)</b>	Способ организации сети, при помощи которого информация передается между компьютерами, входящими в сеть.
<b>Архитектура Ethernet (Ethernet architecture)</b>	Наиболее популярная и наименее затратная сетевая архитектура, включающая в себя следующие топологии - звезда, последовательное соединение, кольцо и гибридные виды.
<b>Архитектура Token-ring (Token-ring architecture)</b>	Архитектура, зачастую используемая в крупных организациях, но в настоящий момент практически устаревшая.
<b>База данных поисковой системы</b>	Набор всей информации, к которой вы можете получить доступ. База данных скрыта от пользователя, но именно в ней компьютер ведет поиск, когда вы направляете свой поисковый запрос.
<b>Беспорядочный веб-сайт (Random Web site)</b>	Сайт, практически не имеющий организации, составляющий массивный набор информации, соединенной различными ссылками.
<b>Бот (Bot)</b>	Также известен как «паук» (spider). Смотрите определение «паука».
<b>Брандмауэр (Firewall)</b>	Программное и/или аппаратное обеспечение, предназначенное для защиты компьютера или сети от угроз извне.
<b>Британский морской</b>	Устройство, соединяющее коаксиальные кабели между собой и с

<b>коннектор (British Naval Connector, BNC)</b>	сетевым адаптером.
<b>Веб-браузер, браузер (Web browser)</b>	Программное обеспечение, которое отображает веб-страницы, включая текст, графику и другое мультимедиа содержимое, например, музыку.
<b>Веб-краулер (Webcrawler)</b>	Программа, которая «обходит» Всемирную Паутину в поисках адресов электронной почты для дальнейшего сохранения и использования для рассылки спама.
<b>Веб-сайт (Web site)</b>	Группировка похожей информации во Всемирной Паутине, состоящая из индивидуальных веб-страниц
<b>Веб-сайт с древовидной структурой (Tree web site)</b>	Сайт, который организован подобно генеалогическому древу с набором различных опций и настроек для просмотра информации на сайте.
<b>Веб-страница (Web page)</b>	Индивидуальная страница, содержащая определенную информацию. Набор сгруппированных веб-страниц образует веб-сайт.
<b>Вирус (Virus)</b>	Небольшая программа, которая копирует себя и нарушает работу компьютеров или сетей.
<b>Витая пара (Twisted-pair cable)</b>	Кабель, состоящий из одной или более пар перевитых между собой медных проводов.
<b>Всемирная Паутина (World Wide Web, WWW)</b>	Набор страниц и сайтов в Интернет, созданных для просмотра веб-браузером. Также известна под названием "Паутина" или веб.
<b>Гибридная сеть (Hybrid mesh network)</b>	Сеть, которая сочетает в себе как минимум две разные топологии построения.
<b>Гигабитный Ethernet (Gigabit Ethernet)</b>	Новая, более быстрая архитектура Ethernet, которая передает информацию более чем в десять раз быстрее по сравнению с архитектурой Fast Ethernet.
<b>Гиперссылка (Hyperlink)</b>	Ссылка на веб-сайте, которая позволяет вам перемещаться между веб-страницами. Гиперссылка обычно отображается подчеркнутым текстом, часто синего цвета.
<b>Гипертекст (Hypertext)</b>	Способ связывания информации с различных веб-страниц или даже с различных сайтов с помощью гиперссылок.
<b>Динамический IP адрес (Dynamic IP address)</b>	IP адрес, который назначается каждый раз, когда компьютер входит в сеть, на время онлайн сессии.
<b>Дискуссионная группа, список рассылки (Discussion list)</b>	Группа людей, общающихся при помощи электронной почты, имеющих общие интересы и общий адрес, почта на который идет всем находящимся в списке.
<b>Домашняя страница (Home page)</b>	Верхняя, первая страница веб-сайта
<b>Домен верхнего уровня (Top-level domain, TLD)</b>	Суффикс в доменном имени, который демонстрирует тип сервера, хранящего веб-сайт, например - .com (коммерческий) или .edu (образовательный).
<b>Доменное имя (Domain name)</b>	Имя сервера, который содержит всю информацию веб-сайта, например, microsoft.com.
<b>Защищенная (экранированная) витая пара (Shielded Twisted Pair, STP)</b>	Одна или несколько пар медных проводов, которые покрыты защитным металлом или фольгой под пластиковым кожухом, обеспечивающими защиту от помех и сохраняющими целостность данных.
<b>Индексирующая программа (Index program)</b>	Скрытая от пользователя часть поисковой системы, которая структурирует и организует информацию, сохраненную для веб-сайта, в форме, удобной для автоматического поиска.
<b>Инструменты для "подглядывания" (Snooping tools)</b>	Инструменты для наблюдения, используемые для изучения электронной почты, получаемой и отправляемой по Интернет.
<b>Интернет (Internet)</b>	Самая известная и большая в мире компьютерная сеть,



	соединяющая миллионы компьютеров в одну огромную сеть сетей
<b>Интерфейс (Interface)</b>	Окно, которое вы видите на веб-сайте
<b>Кабель-канал (Conduits)</b>	Полая трубка, используемая для защиты кабеля от механических повреждений.
<b>Клиент (Client)</b>	Компьютер в сети, подключающийся к серверу для получения информации.
<b>Коаксиальный кабель (Coaxial cable)</b>	Кабель, выполненный в соответствии с уже устаревшим промышленным стандартом. Похож на кабель для подключения телевизионных антенн. Состоит из медного сердечника в изолирующем слое пластмассы. Поверх этого слоя - экранирующее покрытие металлической оплетки или фольги и защитный слой
<b>Кольцевая сеть, закольцованная сеть (Ring network)</b>	Сеть, построенная на основе непрерывного кабеля, соединяющего компьютеры, которые объединены им в кольцо
<b>Коммутатор (Switch)</b>	Центральное соединительное устройство, похожее на концентратор. Получая информацию из сети, коммутатор отправляет ее в конкретное место назначения в этой сети.
<b>Кража идентификационных данных (Identity theft)</b>	Использование чужих идентификационных данных с целью выдать себя за другое лицо.
<b>Крекер (Cracker)</b>	Другое название хакера.
<b>Куки (Cookie)</b>	Небольшой текстовый файл, содержащий информацию о вашем предыдущем визите на веб-сайт
<b>Линейный веб-сайт (Linear Web site)</b>	Сайт, который организован на манер книги, в которой вы перемещаетесь с первой страницы на вторую и так далее.
<b>Маршрутизатор (Router)</b>	Сетевое оборудование, которое соединяет разные сети и направляет, или маршрутизирует информацию между компьютерами в сети
<b>Мгновенное сообщение (Instant Message, IM)</b>	Сообщение, отправляемое при помощи программы, которая позволяет людям обмениваться информацией по Интернет в режиме реального времени.
<b>Медленная почта, "улиточная" почта (Snail Mail)</b>	Стандартное жаргонное название для традиционной почты, отправляемой через отделения почтовой связи.
<b>Многопользовательская игра (Multiplayer gaming)</b>	Процесс одновременной игры с несколькими людьми по сети.
<b>Модем (Modem)</b>	Сетевое оборудование, которое подключает компьютер к Интернет посредством телефонной линии
<b>Не запрошенная корреспонденция, спам (Spam)</b>	см. Спам.
<b>Незащищенная витая пара (Unshielded Twisted Pair, UTP)</b>	Самый дешевый кабель в настоящее время, сделан из одной или более пар медных проводов без какой-либо защиты.
<b>Новости Usenet (Usenet News)</b>	Сеть компьютеров, которые обмениваются статьями на определенные темы, обеспечивают поддержку продуктам или предлагают ответы на различные категории вопросов.
<b>Новостная группа (Newsgroup)</b>	Сетевая конференция в USENET, организованная для ведения дискуссий и обмена новостями.
<b>Общественная собственность (Public domain)</b>	Бесплатная информация, которая доступна любому пользователю.
<b>Одноранговая сеть (Peer-to-Peer Network)</b>	Сеть, которая объединяет равноправные компьютеры.
<b>Окончание, терминатор</b>	устройство, размещаемое на каждом конце кабеля в сети

<b>(Terminator)</b>	последовательного подключения устройств
<b>Оптоволоконный кабель (Fiber optic cable)</b>	Кабель, который, в отличие от обычного кабеля, вместо электрических импульсов передает импульсы света. Самый дорогостоящий вид кабеля, который рассчитан на большие расстояния.
<b>"Паук" (Spider)</b>	Автоматизированная программа, которая путешествует по Интернет, обнаруживая веб-сайты или определенные веб-страницы, и добавляет их в базовый список. Также известна под названиями бот и веб-краулер.
<b>"Пешая" сеть (Sneaker net)</b>	Процесс переноса файлов, записанных на гибких дисках, с одного компьютера на другой.
<b>Плагиат (Plagiarism)</b>	Использование данных и информации, созданных другими людьми, одновременно с попыткой представить это как собственное творение.
<b>Подмена DNS (DNS spoofing)</b>	Изменение DNS записи таким образом, что она ведет на другой веб-сайт.
<b>Подмена внешнего вида веб-страницы (Web page defacement)</b>	Нелегальный доступ к веб-сайту с целью изменения его внешнего вида и информации.
<b>Подписка (Subscribe)</b>	Добавление вашего адреса электронной почты в список рассылки таким образом, что вы можете участвовать в процессе обсуждения.
<b>Поисковая система (Search engine)</b>	Веб-сайт, на котором вы можете найти интересующую вас информацию, используя набор ключевых слов.
<b>Поисковая система по метаданным (Meta Search Engine)</b>	Веб-сайт с системой, которая исследует огромное количество сайтов поисковых систем и комбинирует для вас полученные результаты.
<b>Порт (Port)</b>	Розетка на соединительном устройстве, в которую подключается кабель от компьютерного оборудования. Центральное соединительное устройство обычно содержит несколько портов.
<b>Потоковое аудио (Streaming audio)</b>	Аудио файлы, начинающие проигрываться в программе через браузер прежде, чем файл загружается полностью.
<b>Потоковое видео (Streaming video)</b>	Видео, которое можно просматривать по Интернет, не ожидая завершения загрузки клипа целиком на ваш компьютер.
<b>Провайдер услуг Интернет (Internet Service Provider, ISP)</b>	Компания, предоставляющая доступ в Интернет. Оплата услуг может осуществляться повременно или на основе учета трафика.
<b>Программа "Троянский конь" (Trojan horse program)</b>	Небольшая программа, которую люди случайно загружают в свою систему и тем самым обеспечивают доступ для хакеров в сеть.
<b>Программное обеспечение против спама (Anti-spam software)</b>	Программное обеспечение, которое фильтрует спам.
<b>Пропускная способность (Bandwidth)</b>	Количество информации, которое может быть передано через сетевое соединение за одну единицу времени
<b>Протокол (Protocol)</b>	Набор правил, которые помогают компьютерам "понимать" друг друга.
<b>Протокол безопасного соединения (Secure Sockets Layer, SSL)</b>	Набор правил, или протокол, используемый для безопасной передачи информации.
<b>Протокол передачи гипертекста (Hypertext transfer protocol, HTTP)</b>	набор правил, или протокол, используемый для отправки и получения информации по Всемирной Паутине
<b>Протокол передачи файлов (File Transfer Protocol, FTP)</b>	Набор правил, или протокол, который управляет перемещением или копированием файлов с одного компьютера на другой
<b>Протокол управления</b>	Протокол (Transmission Control Protocol/Internet Protocol, TCP/IP) -

<b>передачей данных/Межсетевой (Интернет)</b>	набор правил, или протокол, который обеспечивает отправку и получение информации по сети Интернет.
<b>Режим реального времени (Real Time)</b>	Режим, при котором не существует ощутимых промежутков времени между отправкой информации по Интернет одним человеком и ее получением другим через открытое соединение между ними.
<b>Рекламное программное обеспечение (Adware)</b>	Программное обеспечение, демонстрирующее рекламу, всплывающую на вашем экране, когда программа работает.
<b>Сервер (Server)</b>	Компьютер в сети клиент-сервер, который хранит всю информацию и ресурсы, а также обеспечивает доступ к ним с других компьютеров в сети.
<b>Сервер доменных имен (Domain Name Server, DNS)</b>	Сервер, который преобразует IP адрес в доменное имя и наоборот.
<b>Сервер печати (Print server)</b>	Сервер, который управляет процессом печати и хранит все задания для принтера, посланные со всех компьютеров в сети.
<b>Сетевая операционная система (Network Operating System, NOS)</b>	Программное обеспечение, которое контролирует, организует и управляет всей деятельностью, происходящей в сети.
<b>Сетевой этикет (Netiquette)</b>	Набор правил для написания электронных почтовых сообщений.
<b>Сетевой адаптер (Network Interface Card, NIC)</b>	Аппаратное обеспечение, установленное внутри компьютера, которое подсоединяет его к сети.
<b>Сеть (Network)</b>	Группа компьютеров, соединенных каким-либо способом так, что люди могут обмениваться информацией и совместно использовать оборудование
<b>Сеть клиент-сервер (Client/Server Network)</b>	Сеть, в которой выделенный компьютер содержит всю информацию и ресурсы, предоставляя доступ к ним другим компьютерам, находящимся в сети.
<b>Сеть с шинной организацией (Bus network)</b>	Сеть, в которой все компьютеры подсоединяются вдоль одного кабеля, также называемого опорным (backbone)
<b>Сеть с шиной типа "звезда" (Star bus network)</b>	Сеть, в которой каждый компьютер присоединяется к центральной точке сети. Одна из самых часто применяемых в настоящее время сетевых топологий.
<b>Совместное использование файлов в одноранговой сети (Peer-to-Peer file sharing)</b>	Использование файлов в сети Интернет совместно с другими пользователями непосредственно с вашего компьютера.
<b>Соединение класса T1 (T1 connection)</b>	Соединение, которое используется компаниями и зачастую небольшими провайдерами для подключения к сети Интернет на скорости приблизительно 1.544 мегабит в секунду.
<b>Соединение класса T3 (T3 connection)</b>	Соединение, в котором используется оптоволоконный кабель для передачи информации на скоростях до 44.73 мегабит в секунду.
<b>Социальное проектирование (Social engineering).</b>	Специальная игра, разработанная для получения информации, которую обычно никто не станет раскрывать незнакомцу.
<b>Спам (Spam)</b>	Электронный почтовый мусор с предложениями что-либо продать потребителю.
<b>Среда передачи данных (Transmission media)</b>	Другое название кабельной или беспроводной сети, используемой для передачи данных.
<b>Статический IP адрес (Static IP address)</b>	Фиксированный IP адрес, назначаемый определенному компьютеру. Статический IP адрес является необходимым для веб-серверов
<b>Стек протоколов TCP/IP (TCP/IP Protocol Suite)</b>	Набор протоколов или правил, которые управляют передачей информации по сети Интернет.
<b>Сцепление гирляндой (Daisy chaining)</b>	Вид соединения нескольких концентраторов.
<b>Топология (Topology)</b>	Структура или тип построения и разводки сети, часто зависит от

	сетевой архитектуры.
<b>Трассировщики клавиатуры (Keystroke loggers)</b>	Программы, которые записывают каждую нажатую на компьютере клавишу и отправляют данную информацию человеку, который отслеживает ее.
<b>Трассировщики пакетов (Packet sniffers)</b>	Программы, которые наблюдают за информацией в сети.
<b>Универсальная поисковая система (General Purpose Search Engine)</b>	Поисковая система, охватывающая широкий спектр информации, удобна для поиска неспециализированной информации.
<b>Универсальный локатор ресурса (Uniform resource locator, URL)</b>	Адрес веб-сайта во Всемирной Паутине, например, <a href="http://www.microsoft.com">http - //www.microsoft.com</a> .
<b>Устройство хранения информации, накопитель (Storage device)</b>	Устройство, на котором вы храните файлы, например, жесткий диск, компакт диск, ленточный и оптический приводы.
<b>Хаб, концентратор (Hub)</b>	Центральное соединительное устройство, к которому присоединяются все сетевые кабели.
<b>Хакер (Hacker)</b>	Человек, который получает несанкционированный доступ к вашей компьютерной системе с целью уничтожения или похищения данных.
<b>Хождение по сети (Surfing the net)</b>	Пролистывание страниц во Всемирной Паутине.
<b>Центральный процессор (Central Processing Unit, CPU)</b>	Микросхема, которая управляет вашим компьютером.
<b>Цифровая абонентская линия (Digital Subscriber Line, DSL)</b>	Вид подключения, который использует существующую телефонную линию для установки постоянного Интернет-соединения на высокой скорости от 1 до 9 мегабит.
<b>Цифровая сеть интегрированного обслуживания (Integrated Services Digital Network, ISDN)</b>	Вид подключения, который передает информацию со скоростью 128 килобит в секунду
<b>Чат (Chat)</b>	Программа, позволяющая группам людей общаться в реальном времени, используя Интернет.
<b>Червь (Worm)</b>	Небольшая программа, которая похожа на вирус. Червь постоянно себя копирует и не требует "хозяина" для заражения компьютера. Часто попадает в компьютер через электронную почту.
<b>Шлюз (Gateway)</b>	Устройство сопряжения, которое соединяет два разных типа сетей. Шлюз получает информацию, переводит ее, а затем пересылает перевод по месту назначения.
<b>Шифрование (Encryption)</b>	Процесс кодирования пересылаемой информации таким образом, чтобы ее мог прочитать только человек или компьютер, которому она предназначена.
<b>Шпионское программное обеспечение (Spyware)</b>	Программа, которая без вашего разрешения отправляет кому-либо информацию о том, что вы делаете в Интернет.
<b>Электронная почта (Electronic mail, email)</b>	Сообщение, пересылаемое по Интернету.
<b>Ява апплет (Java applet)</b>	Небольшая программа, которая запускается внутри браузера и выполняет файлы Ява.
<b>Язык гипертекстовой разметки (Hypertext markup language, HTML)</b>	Язык программирования, используемый для создания веб-страниц.
<b>Адаптер</b>	Устройство сопряжения центрального процессора и периферийных устройств компьютера; кроме того, иногда осуществляет функции управления периферийным устройством.

<b>Адаптер локальной сети</b>	Адаптер для подключения компьютера к локальной сети компьютеров
<b>Аккумулятор</b>	Устройство, вырабатывающее электричество путем преобразования химической энергии в электрическую.
<b>Базовая система ввода/вывода (иначе, BIOS)</b>	Устройство, предназначенное для выполнения следующих функций: тестирование основных устройств компьютера; распознавание типов устройств; вызов блока начальной загрузки операционной системы; обслуживание системных прерываний.
<b>Буфер</b>	Дополнительная память для временного хранения данных.
<b>Быстродействие процессора</b>	Скорость выполнения операций процессором.
<b>Векторная графика</b>	Способ представления изображения как совокупности графических элементов.
<b>Видеоадаптер</b>	Электронная плата, которая обрабатывает видеоданные и управляет работой дисплея
<b>Волоконно- оптический кабель</b>	Кабель, передающий данные с помощью света.
<b>Графический адаптер</b>	Устройство, управляющее дисплеем и обеспечивающее вывод графических изображений. Определяет разрешающую способность дисплея, количество цветов

## 7. ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ, ВНЕСЕННЫХ В РАБОЧУЮ ПРОГРАММУ

№ изменения, дата внесения изменения, № страницы с изменением:	
<b>БЫЛО:</b>	<b>СТАЛО:</b>
Основание:	
Подпись лица, внесшего изменения	
№ изменения, дата внесения изменения, № страницы с изменением:	

**БЫЛО:**

**СТАЛО:**

Основание:

Подпись лица, внесшего изменения

**ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ЭКОНОМИКИ И СЕРВИСА» В Г. АРТЕМЕ**

Техническая экспертиза программы учебной практики по модулям ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем, ПМ.02 Разработка и администрирование баз данных, ПМ.03 Участие в интеграции программных модулей, ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин», специальности 09.02.03 Программирование в компьютерных системах, представленной преподавателем кафедры экономики, управления и информационных технологий филиала Волошиным Е.В.

**ЭКСПЕРТНОЕ ЗАКЛЮЧЕНИЕ**

№	Наименование экспертного показателя	Экспертная оценка	
		да	нет
<b>Экспертиза оформления титульного листа и содержания</b>			
1.	Наименование рабочей программы модуля на титульном листе совпадает с наименованием модуля в тексте ФГОС и учебном плане колледжа	да	
2.	Название филиала соответствует названию по Уставу	да	
3.	На титульном листе указан код и наименование специальности	да	
4.	Оборотная сторона титульного листа содержит все необходимые данные (выходные данные ФГОС, ФИО экспертов, сведения о согласовании программы)	да	
5.	Нумерация страниц в «Содержании» верна	да	
<b>Экспертиза раздела 1 «Паспорт рабочей программы профессионального модуля»</b>			
6.	Раздел 1 «Паспорт рабочей программы профессионального модуля» имеется	да	
7.	Подстрочные надписи удалены	да	
8.	Наименование программы профессионального модуля совпадает с наименованием на титульном листе	да	
9.	Пункт 1.1. «Область применения программы» заполнен	да	
10.	Паспорт программы профессионального модуля содержит базовую часть	да	
11.	Наименование основного вида профессиональной деятельности (ВПД) совпадает с наименованием профессионального модуля	да	
12.	Перечень профессиональных компетенций (ПК) содержит все компетенции, перечисленные в тексте ФГОС	да	
13.	Пункт 1.2. «Цели и задачи модуля – требования к результатам освоения модуля» заполнен	да	



14.	Требования к практическому опыту, умениям и знаниям соответствуют перечисленным в тексте ФГОС	да	
15.	Пункт 1.3. «Рекомендуемое количество часов на освоение программы профессионального модуля» заполнен	да	
16.	Все строки и графы пункта 1.3. заполнены	да	
17.	Перечислены виды самостоятельной работы	да	
18.	Указанное количество часов в графе «Итого» соответствует учебному плану	да	
<b>Экспертиза раздела 2 «Результаты освоения профессионального модуля»</b>			
19.	Раздел 2 «Результаты освоения профессионального модуля» имеется	да	
20.	Перечень профессиональных компетенций совпадает с указанными в п. 1.1	да	
21.	Перечень общих компетенций соответствует перечисленным в тексте ФГОС	да	
<b>Экспертиза раздела 3 «Структура и содержание профессионального модуля»</b>			
22.	Раздел 3 «Структура и содержание профессионального модуля» имеется	да	
23.	Форма таблицы 3.1. «Тематический план профессионального модуля» соответствует макету программы	да	
24.	Таблица 3.1. «Тематический план профессионального модуля» заполнена	да	
25.	Форма таблицы 3.2. «Содержание обучения по профессиональному модулю (ПМ)» соответствует макету программы	да	
26.	Таблица 3.2. «Содержание обучения по профессиональному модулю (ПМ)» заполнена	да	
27.	Количество и наименования междисциплинарных курсов совпадают с указанными в тексте ФГОС	да	
28.	Перечислены виды самостоятельной работы студентов, сформулированные через деятельность	да	
29.	Наименования разделов модуля в табл. 3.1 и 3.2 совпадают	да	
<b>Экспертиза раздела 4 «Условия реализации программы профессионального модуля»</b>			
30.	Раздел 4 «Условия реализации программы профессионального модуля» имеется	да	
31.	Пункт 4.1. «Требования к минимальному материально-техническому обеспечению» заполнен	да	
32.	Пункт 4.2. «Информационное обеспечение обучения» заполнен в соответствии с требованиями ГОСТ по оформлению литературы	да	
33.	В списке основной литературы отсутствуют издания, выпущенные более 5 лет назад	да	
34.	Пункт 4.3. «Общие требования к организации образовательного процесса» заполнен	да	
35.	Пункт 4.4. «Кадровое обеспечение образовательного процесса» заполнен	да	
<b>Экспертиза раздела 5 «Контроль и оценка результатов освоения профессионального модуля (вида профессиональной деятельности)»</b>			
36.	Раздел 5. «Контроль и оценка результатов освоения профессионального модуля (вида	да	

	профессиональной деятельности)» имеется		
37.	Наименования профессиональных и общих компетенций совпадают с указанными в п. 1.1	да	
<b>Экспертиза показателей объемов времени, отведенных на освоение ПМ, указанных в п. 1.3 раздела 1 «Паспорт рабочей программы профессионального модуля» и в табл. 3.1 и 3.2 раздела 3 «Структура и содержание профессионального модуля»</b>			
38.	Общий объем времени, отведенного на освоение модуля (всего часов), в паспорте программы, таблицах 3.1 и 3.2 совпадает	да	
39.	Объем обязательной аудиторной нагрузки в паспорте программы, таблицах 3.1 и 3.2 совпадает	да	
40.	Объем времени, отведенного на выполнение лабораторных и практических занятий, в таблицах 3.1 и 3.2 совпадает	да	
41.	Объем времени, отведенного на практику, в паспорте программы, таблицах 3.1 и 3.2 совпадает	да	
42.	Объем времени, отведенного на самостоятельную работу студентов, в паспорте программы, таблицах 3.1 и 3.2 совпадает	да	

<b>ИТОГОВОЕ ЗАКЛЮЧЕНИЕ</b>		да	нет
Программа профессионального модуля может быть направлена на содержательную экспертизу		да	

Разработчик программы: \_\_\_\_\_ Е.В. Волошин

СОГЛАСОВАНО

Заведующий кафедрой \_\_\_\_\_ А.А.Власенко

«22» апреля 2020 г.

Заведующий отделением \_\_\_\_\_ М.С.Словилова

Методист УМЧ \_\_\_\_\_ Т.И. Теплякова

«25 » апреля 2020 г.

**ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ЭКОНОМИКИ И СЕРВИСА» В Г. АРТЕМЕ**

Содержательная экспертиза рабочей программы учебной практики по модулям ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем, ПМ.02 Разработка и администрирование баз данных, ПМ.03 Участие в интеграции программных модулей, ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин», специальности 09.02.03 Программирование в компьютерных системах, представленной преподавателем кафедры экономики, управления и информационных технологий филиала Волошиным Е.В.

**ЭКСПЕРТНОЕ ЗАКЛЮЧЕНИЕ**

№	Наименование экспертного показателя	Экспертная оценка			Примечание
		да	нет	заключение отсутствует	
<b>Экспертиза раздела 1 «Паспорт программы профессионального модуля»</b>					
1.	Требования к результатам освоения модуля соответствуют перечисленным в ФГОС СПО (в т. ч. конкретизируют и/или расширяют требования ФГОС)	да			
2.	Возможности использования профессионального модуля описаны полно и точно.	да			
3.	Вариативная часть содержит требования к результатам освоения модуля (при наличии)	не предусмотрена			
<b>Экспертиза раздела 2 «Результаты освоения профессионального модуля»</b>					
4.	Перечень общих и профессиональных компетенций, представленных в разделе модуля, соответствует перечисленным компетенциям, указанным к данному модулю в ФГОС СПО.	да			
<b>Экспертиза раздела 3 «Структура и содержание профессионального модуля»</b>					
5.	Структура программы профессионального модуля соответствует принципу единства теоретического и практического обучения.	да			
6.	Разделы программы модуля выделены дидактически целесообразно.	да			
7.	Соотношение учебной и производственной практики соответствует	да			

	дидактическим задачам модуля.				
8.	Выбор варианта проведения практики (концентрированно, рассредоточено, комбинированно) дидактически целесообразен.	да			
9.	Содержание практики (виды работ) соответствует требованиям к практическому опыту и умениям	да			
10.	Содержание учебного материала соответствует требованиям к знаниям и умениям.	да			
11.	Объем времени достаточен для освоения указанного в содержании учебного материала.	да			
12.	Объем и содержание лабораторных и практических работ определены дидактически целесообразно и соответствуют требованиям к умениям и знаниям и ориентированы на подготовку к овладению ПК профессионального модуля.	да			
13.	Уровни освоения соответствуют видам учебной деятельности в разделе.	да			
14.	Тематика домашних заданий определена дидактически целесообразно.	да			
15.	Содержание самостоятельной работы студентов, в т.ч. внеаудиторной, направлено на выполнение требований к результатам освоения ПМ («иметь практический опыт», «уметь», «знать»).	да			
16.	Формулировки самостоятельной работы понимаются однозначно.	да			
17.	Примерная тематика курсовых работ соответствует целям и задачам освоения профессионального модуля <i>(пункт заполняется, если в программе дисциплины предусмотрена курсовая работа)если в программе дисциплины предусмотрена курсовая работа)</i>	не предусмотрена			
18.	Содержание программы модуля предусматривает формирование перечисленных общих и профессиональных компетенций.	да			
<b>Экспертиза раздела 4 «Условия реализации программы профессионального модуля»</b>					
19.	Перечень учебных кабинетов (мастерских, лабораторий и др.) обеспечивает изучение междисциплинарного курса, проведение всех видов лабораторных и практических работ и тем учебной практики, предусмотренных программой профессионального модуля.	да			

20.	Перечисленное оборудование обеспечивает изучение междисциплинарного курса, проведение всех видов лабораторных и практических работ и тем учебной практики, предусмотренных программой профессионального модуля.	да			
21.	Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники.	да			
22.	Перечисленные Интернет-ресурсы актуальны и достоверны (пункт заполняется, если нормативно-правовые акты указаны в качестве источников).	да			
23.	Перечисленные источники соответствуют структуре и содержанию программы модуля.	да			
24.	Информационные источники указаны с учетом содержания модуля.	да			
25.	Общие требования к организации образовательного процесса соответствуют модульно - компетентностному подходу.	да			
26.	Общие требования к организации образовательного процесса описаны подробно (перечислены условия проведения занятий, организация учебной практики, консультационной помощи обучающимся).	да			
27.	Дисциплины, изучение которых должно предшествовать освоению данного модуля, определены с учетом принципа систематичности и последовательности обучения.	да			
28.	Требования к кадровому обеспечению (в т.ч. к уровню квалификации преподавателей МДК и руководителя практики) позволяют обеспечить должный уровень подготовки современного рабочего (специалиста).	да			
<b>Экспертиза раздела 5 «Контроль и оценка результатов освоения профессионального модуля (вида профессиональной деятельности)»</b>					
29.	Основные показатели оценки результатов обучения позволяют однозначно диагностировать сформированность соответствующих профессиональных компетенций (ПК).	да			
	Наименование форм и методов контроля и оценки освоения ПК точно и однозначно описывает процедуру аттестации.	да			

30.	Формы и методы контроля и оценки освоения ПК позволяют оценить сформированность ПК.	да			
31.	Основные показатели оценки результата позволяют однозначно диагностировать сформированность соответствующих общих компетенций (ОК).	да			
32.	Наименование форм и методов контроля и оценки освоения ОК точно и однозначно описывает процедуру аттестации.	да			
33.	Формы и методы контроля и оценки освоения ОК позволяют оценить сформированность ОК.	да			

Итоговое заключение (из трех альтернативных позиций следует выбрать одну)	да	нет
Программа профессионального модуля может быть рекомендована к утверждению	да	
Программу профессионального модуля следует рекомендовать к доработке		
Программу профессионального модуля следует рекомендовать к отклонению		

**Замечания и рекомендации эксперта по доработке**

Разработчик программы: \_\_\_\_\_

Е.В. Волошин

СОГЛАСОВАНО

И.о.Заведующий кафедрой \_\_\_\_\_

А.А.Власенко

«22» апреля 2020 г.

Заведующий отделением \_\_\_\_\_

М.С.Словилова

Методист УМЧ \_\_\_\_\_

Т.И. Теплякова

«25 » апреля 2020 г.

**ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ЭКОНОМИКИ И СЕРВИСА» В Г. АРТЕМЕ**

Содержательная экспертиза рабочей программы учебной практики по модулям ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем, ПМ.02 Разработка и администрирование баз данных, ПМ.03 Участие в интеграции программных модулей, ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин» специальности 09.02.03 Программирование в компьютерных системах, представленной преподавателем кафедры экономики, управления и информационных технологий филиала Волошиным Е.В..

**ЭКСПЕРТНОЕ ЗАКЛЮЧЕНИЕ**

№	Наименование экспертного показателя	Экспертная оценка			Примечание
		да	нет	заключение отсутствует	
<b>Экспертиза раздела 1 «Паспорт программы профессионального модуля»</b>					
1.	Требования к результатам освоения модуля соответствуют перечисленным в ФГОС СПО (в т. ч. конкретизируют и/или расширяют требования ФГОС)	да			
2.	Возможности использования профессионального модуля описаны полно и точно.	да			
3.	Вариативная часть содержит требования к результатам освоения дисциплины (при наличии)	не предусмотрена			
<b>Экспертиза раздела 2 «Результаты освоения профессионального модуля»</b>					
4.	Перечень общих и профессиональных компетенций, представленных в разделе модуля, соответствует перечисленным компетенциям, указанным к данному модулю в ФГОС СПО.	да			
<b>Экспертиза раздела 3 «Структура и содержание профессионального модуля»</b>					
5.	Структура программы профессионального модуля соответствует принципу единства теоретического и практического обучения.	да			
6.	Разделы программы модуля выделены дидактически целесообразно.	да			
7.	Соотношение учебной и производственной практики соответствует	да			

	дидактическим задачам модуля.				
8.	Выбор варианта проведения практики (концентрированно, рассредоточено, комбинированно) дидактически целесообразен.	да			
9.	Содержание практики (виды работ) соответствует требованиям к практическому опыту и умениям	да			
10.	Содержание учебного материала соответствует требованиям к знаниям и умениям.	да			
11.	Объем времени достаточен для освоения указанного в содержании учебного материала.	да			
12.	Объем и содержание лабораторных и практических работ определены дидактически целесообразно и соответствуют требованиям к умениям и знаниям и ориентированы на подготовку к овладению ПК профессионального модуля.	да			
13.	Уровни освоения соответствуют видам учебной деятельности в разделе.	да			
14.	Тематика домашних заданий определена дидактически целесообразно.	да			
15.	Содержание самостоятельной работы студентов, в т.ч. внеаудиторной, направлено на выполнение требований к результатам освоения ПМ («иметь практический опыт», «уметь», «знать»).	да			
16.	Формулировки самостоятельной работы понимаются однозначно.	да			
17.	Примерная тематика курсовых работ соответствует целям и задачам освоения профессионального модуля <i>(пункт заполняется, если в программе дисциплины предусмотрена курсовая работа)если в программе дисциплины предусмотрена курсовая работа)</i>	не предусмотрена			
18.	Содержание программы модуля предусматривает формирование перечисленных общих и профессиональных компетенций.	да			
<b>Экспертиза раздела 4 «Условия реализации программы профессионального модуля»</b>					
19.	Перечень учебных кабинетов (мастерских, лабораторий и др.) обеспечивает изучение междисциплинарных курсов, проведение всех видов лабораторных и практических работ и тем учебной практики, предусмотренных программой профессионального	да			



	модуля.				
20.	Перечисленное оборудование обеспечивает изучение междисциплинарных курсов, проведение всех видов лабораторных и практических работ и тем учебной практики, предусмотренных программой профессионального модуля.	да			
21.	Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники.	да			
22.	Перечисленные Интернет-ресурсы актуальны и достоверны (пункт заполняется, если нормативно-правовые акты указаны в качестве источников).	да			
23.	Перечисленные источники соответствуют структуре и содержанию программы модуля.	да			
24.	Информационные источники указаны с учетом содержания модуля.	да			
25.	Общие требования к организации образовательного процесса соответствуют модульно - компетентностному подходу.	да			
26.	Общие требования к организации образовательного процесса описаны подробно (перечислены условия проведения занятий, организация учебной практики , консультационной помощи обучающимся).	да			
27.	Дисциплины, изучение которых должно предшествовать освоению данного модуля, определены с учетом принципа систематичности и последовательности обучения.	да			
28.	Требования к кадровому обеспечению (в т.ч. к уровню квалификации преподавателей МДК и руководителя практики) позволяют обеспечить должный уровень подготовки современного рабочего (специалиста).	да			
<b>Экспертиза раздела 5 «Контроль и оценка результатов освоения профессионального модуля (вида профессиональной деятельности)»</b>					
29.	Основные показатели оценки результатов обучения позволяют однозначно диагностировать сформированность соответствующих профессиональных компетенций (ПК).	да			
	Наименование форм и методов контроля и оценки освоения ПК	да			

	точно и однозначно описывает процедуру аттестации.				
30.	Формы и методы контроля и оценки освоения ПК позволяют оценить сформированность ПК.	да			
31.	Основные показатели оценки результата позволяют однозначно диагностировать сформированность соответствующих общих компетенций (ОК).	да			
32.	Наименование форм и методов контроля и оценки освоения ОК точно и однозначно описывает процедуру аттестации.	да			
33.	Формы и методы контроля и оценки освоения ОК позволяют оценить сформированность ОК.	да			

<b>Итоговое заключение</b> (из трех альтернативных позиций следует выбрать одну)	<b>да</b>	<b>нет</b>
Программа профессионального модуля может быть рекомендована к утверждению	да	
Программу профессионального модуля следует рекомендовать к доработке		
Программу профессионального модуля следует рекомендовать к отклонению		

**Замечания и рекомендации эксперта по доработке** \_\_\_\_\_

**Разработчик программы:** \_\_\_\_\_ **Е.В. Волошин**  
«22 апреля» 2020 г.

**Эксперты:** \_\_\_\_\_ **В.В. Неслюзов**  
\_\_\_\_\_ **О.В. Бажин**

« 25 » апреля 2020 г.

МИНОБРНАУКИ РОССИИ

ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО  
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ЭКОНОМИКИ И СЕРВИСА» В Г. АРТЁМЕ



**КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА  
УЧЕБНОЙ ПРАКТИКИ  
К ПРОФЕССИОНАЛЬНЫМ МОДУЛЯМ:**

**ПМ.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ;**

**ПМ.02 РАЗРАБОТКА И АДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ**

**ПМ.03 УЧАСТИЕ В ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ**

**ПМ.04 ВЫПОЛНЕНИЕ РАБОТ ПО ПРОФЕССИИ РАБОЧЕГО «ОПЕРАТОР  
ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНЫХ И ВЫЧИСЛИТЕЛЬНЫХ МАШИН»**

К программе подготовки специалистов среднего звена  
по специальности

09.02.03 Программирование в компьютерных системах

Год набора на ООП

2020

Артем 2020

**СОГЛАСОВАНО**

Протокол заседания научно-методического совета

от 18 мая 2020 года № 7

Председатель  О.И. Иванюга


**РАСМОТРЕНО И ОДОБРЕНО**

на заседании кафедры ЭУИТ


Протокол № 14 ЭУИТ от 06 мая 2020

И.о. зав. кафедрой  А.А. Власенко

**Разработчик:**

Место работы	Занимаемая должность, ученая степень и ученое (почетное) звание, квалификационная категория	Инициалы, фамилия	Подпись
Филиал ФГБОУ ВО «ВГУЭС» в г. Артёме	Преподаватель кафедры экономики, управления и информационных технологий	Е.В.Волошин	

**Эксперты:**

Место работы	Занимаемая должность, ученая степень и ученое (почетное) звание, квалификационная категория	Инициалы, фамилия	Подпись
ООО «СКС – Сервис», г. Артем	Директор	О.В. Бажин	
Филиал ФГБОУ ВО «ВГУЭС» в г. Артеме	Руководитель информационно-технического центра	В.В. Неслюзов	

## СОДЕРЖАНИЕ

<b>1. ОБЩИЕ ПОЛОЖЕНИЯ .....</b>	<b>50</b>
<b>2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ .....</b>	<b>50</b>
<b>3. СТРУКТУРА И СОДЕРЖАНИЕ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ .....</b>	<b>58</b>
<b>3.1 ПМ.04 ВЫПОЛНЕНИЕ РАБОТ ПО ПРОФЕССИИ РАБОЧЕГО «ОПЕРАТОР ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНЫХ И ВЫЧИСЛИТЕЛЬНЫХ МАШИН».....</b>	<b>59</b>
<b>3.2 ПМ.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ.....</b>	<b>275</b>
<b>3.3 ПМ.02 РАЗРАБОТКА И АДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ .....</b>	<b>287</b>
<b>3.4 ПМ.03 УЧАСТИЕ В ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ .....</b>	<b>436</b>
<b>4. ШКАЛА ОЦЕНКИ ОБРАЗОВАТЕЛЬНЫХ ДОСТИЖЕНИЙ .....</b>	<b>493</b>
<b>5. ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ И ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....</b>	<b>493</b>

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной практики к профессиональным модулям:

**ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем;**

**ПМ.02 Разработка и администрирование баз данных**

**ПМ.03 Участие в интеграции программных модулей**

**ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»**

КОС разработаны в соответствии с:

- программой подготовки специалистов среднего звена по специальности **09.03.02**

**Программирование в компьютерных системах;**

- программой учебной практики

**Формы контроля и оценивания элементов профессионального модуля**

Таблица 1

Элемент модуля	Форма контроля и оценивания	
	Промежуточная аттестация	Текущий контроль
УП.04.01.Учебная практика	Дифференцированный зачёт	- выполнение практических работ по индивидуальным заданиям
УП.01.01.Учебная практика	Дифференцированный зачёт	- выполнение практических работ по индивидуальным заданиям
УП.02.01 Учебная практика	Дифференцированный зачет	- выполнение практических работ по индивидуальным заданиям
УП.03.01.Учебная практика	Дифференцированный зачёт	- выполнение практических работ по индивидуальным заданиям

## 2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

### 2.1. Профессиональные и общие компетенции:

В результате аттестации по учебным практикам осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Таблица 2

Профессиональные и общие компетенции	Показатели оценки результата
ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.	– участие в работе научно-студенческих обществ, – выступления на научно-практических конференциях, – участие во внеурочной деятельности связанной с будущей профессией/специальностью (конкурсы профессионального мастерства, выставки и т.п.) – высокие показатели производственной деятельности
ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.	– выбор и применение методов и способов решения профессиональных задач, – оценка их эффективности и качества

Профессиональные и общие компетенции	Показатели оценки результата
ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	<ul style="list-style-type: none"> <li>– анализ профессиональных ситуаций;</li> <li>– решение стандартных и нестандартных профессиональных задач</li> </ul>
ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития	<ul style="list-style-type: none"> <li>– эффективный поиск необходимой информации;</li> <li>– использование различных источников, включая электронные при изучении теоретического материала и прохождении различных этапов производственной практики</li> </ul>
ОК 5.Использовать информационно-коммуникационные технологии в профессиональной деятельности.	<ul style="list-style-type: none"> <li>– использование в учебной и профессиональной деятельности различных видов программного обеспечения, в том числе специального, при оформлении и презентации всех видов работ</li> </ul>
ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями	<p>взаимодействие:</p> <ul style="list-style-type: none"> <li>– с обучающимися при проведении деловых игр, выполнении коллективных заданий (проектов),</li> <li>– с преподавателями, мастерами в ходе обучения,</li> <li>– с потребителями и коллегами в ходе производственной практики</li> </ul>
ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.	<ul style="list-style-type: none"> <li>– самоанализ и коррекция результатов собственной деятельности при выполнении коллективных заданий (проектов),</li> <li>– ответственность за результат выполнения заданий.</li> </ul>
ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации	<ul style="list-style-type: none"> <li>– планирование и качественное выполнение заданий для самостоятельной работы при изучении теоретического материала и прохождении различных этапов производственной практики ;</li> <li>– определение этапов и содержания работы по реализации самообразования</li> </ul>
ОК 9. Осуществлять профессиональную деятельность в условиях обновления её целей, содержания, смены технологий	<ul style="list-style-type: none"> <li>– адаптация к изменяющимся условиям профессиональной деятельности;</li> <li>– проявление профессиональной маневренности при прохождении различных этапов производственной практики</li> </ul>
ПК 1.1. Выполнять разработку спецификаций отдельных компонент	<ul style="list-style-type: none"> <li>– создание отдельных компонент</li> <li>– выполнение спецификаций компонент</li> </ul>
ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля	<ul style="list-style-type: none"> <li>– выполнение создания кода программного продукта на уровне модуля в соответствии с требованиями к готовому программному продукту.</li> <li>– разработка пользовательского интерфейса.</li> </ul>
ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств	<ul style="list-style-type: none"> <li>– использование специализированных программных средств отладки программных модулей.</li> <li>– разработка компонент программных модулей с использованием современных инструментальных средств и технологий</li> </ul>
ПК 1.4. Выполнять тестирование	<ul style="list-style-type: none"> <li>– выполнение тестирования качества разработки</li> </ul>

Профессиональные и общие компетенции	Показатели оценки результата
программных модулей	программных модулей с помощью разработанных тестовых наборов и сценариев. – определение ошибок в программном коде с использованием тестовых наборов.
ПК 1.5. Осуществлять оптимизацию программного кода модуля	– выявление избыточности кода программного продукта и его оптимизация. – анализ оптимизации программного кода модуля
ПК 1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций	– использование инструментальных средств и графических языков спецификаций для создания компонент проектной и технической документации. – оформление проектной и технической документации в соответствии со стандартами.
ПК 2.1. Разрабатывать объекты базы данных.	– определение и нормализация отношений между объектами баз данных; – изложение правил установки отношений между объектами баз данных; – демонстрация нормализации и установки отношений между объектами баз данных; – выбор методов описания и построения схем баз данных; – демонстрация построения схем баз данных; – демонстрация методов манипулирования данными; – выбор типа запроса к СУБД; – демонстрация построения запроса к СУБД
ПК 2.2. Реализовывать базу данных в конкретной СУБД.	– выбор архитектуры и типового клиента доступа в соответствии с технологией разработки базы данных; – выбор технологии разработки базы данных исходя из ее назначения; – изложение основных принципов проектирования баз данных; – демонстрация построения концептуальной, логической и физической моделей данных с помощью утилиты автоматизированного проектирования базы данных; – выбор и использование утилит автоматизированного проектирования баз данных; – демонстрация навыков разработки серверной части базы данных в инструментальной оболочке; – демонстрация навыков модификации серверной части базы данных в инструментальной оболочке; – демонстрация навыков разработки клиентской части базы данных в инструментальной оболочке; – демонстрация навыков построения запросов SQL к базе данных; – демонстрация навыков изменения базы данных (в соответствии с ситуацией)
ПК 2.3. Решать вопросы администрирования базы данных.	– определение вида и архитектуры сети, в которой находится база данных;



Профессиональные и общие компетенции	Показатели оценки результата
	<ul style="list-style-type: none"> <li>– определение модели информационной системы;</li> <li>– выбор сетевой технологии и, исходя из нее, методов доступа к базе данных;</li> <li>– выбор и настройка протоколов разных уровней для передачи данных по сети;</li> <li>– демонстрация устранения ошибок межсетевого взаимодействия в сетях;</li> <li>– выбор технологии разработки базы данных, исходя из требований к ее администрированию;</li> <li>– демонстрация навыков разработки и модификации серверной части базы данных в инструментальной оболочке с возможностью ее администрирования;</li> <li>– демонстрация навыков разработки и модификации клиентской части базы данных в инструментальной оболочке с возможностью ее администрирования;</li> <li>– демонстрация навыков построения запросов SQL к базе данных с учетом распределения прав доступа;</li> <li>– демонстрация навыков изменения прав доступа в базе данных (в соответствии с ситуацией);</li> <li>– определение ресурсов администрирования базы данных;</li> <li>– демонстрация навыков правильного использования программных средств защиты</li> </ul>
<p>ПК 2.4. Реализовывать методы и технологии защиты информации в базах данных.</p>	<ul style="list-style-type: none"> <li>– выбор сетевой технологии и, исходя из неё, методов доступа к базе данных;</li> <li>– выбор и настройка протоколов разных уровней для передачи данных по сети;</li> <li>– демонстрация устранения ошибок межсетевого взаимодействия в сетях;</li> <li>– демонстрация использования сетевых устройств для защиты данных базы данных при передаче по сети;</li> <li>– демонстрация обеспечения непротиворечивости и целостности данных в базе данных;</li> <li>– демонстрация навыков внесения изменения в базу данных для защиты информации;</li> <li>– демонстрация навыков правильного использования аппаратных средств защиты;</li> <li>– демонстрация навыков правильного использования программных средств защиты</li> </ul>
<p>ПК 3.1 – Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения</p>	<ul style="list-style-type: none"> <li>– определение характера взаимодействия компонентов программного обеспечения</li> <li>– участие в проектировании программного обеспечения с использованием языка объектно-ориентированного моделирования UML</li> <li>– точность и грамотность оформления технологической документации</li> </ul>
<p>ПК 3.2. Выполнять интеграцию модулей в программную систему</p>	<ul style="list-style-type: none"> <li>– владение основными подходами интеграции программных модулей</li> </ul>

Профессиональные и общие компетенции	Показатели оценки результата
	<ul style="list-style-type: none"> <li>– определение этапов разработки программного обеспечения</li> <li>– демонстрация построения концептуальной, логической и физической моделей баз данных</li> <li>– самостоятельный обоснованный выбор технологии разработки исходного модуля исходя из его назначения</li> <li>– самостоятельный обоснованный выбор методов разработки программных модулей</li> <li>– демонстрация навыков модификации программных модулей</li> </ul>
ПК 3.3. Выполнять отладку программного продукта с использованием специализированных программных средств	<ul style="list-style-type: none"> <li>– овладение принципами построения, структурами и приемами работы с инструментальными средствами, поддерживающими отладку программного обеспечения</li> <li>– определение возможности увеличения быстродействия программного продукта</li> <li>– определение способов и принципов оптимизации</li> <li>– выбор методов отладки программных модулей и программного обеспечения</li> <li>– выбор специализированных средств для отладки программного обеспечения</li> <li>– демонстрация устранения ошибок в программных модулях</li> </ul>
ПК 3.4. Осуществлять разработку тестовых наборов и тестовых сценариев	<ul style="list-style-type: none"> <li>– тестовый набор для тестирования программного обеспечения на уровне пакета, модуля, класса, компонента</li> <li>– тестовый сценарий для тестирования графического интерфейса</li> <li>– результаты нагрузочного тестирования программного обеспечения</li> <li>– демонстрация навыков правильного использования инструментальных средств тестирования программных модулей</li> </ul>
ПК 3.5. Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования	<ul style="list-style-type: none"> <li>– оформление программного кода программного модуля в соответствии с требованиями стилевого оформления</li> <li>– оформление программного кода программного обеспечения в соответствии с требованиями стандартов кодирования</li> <li>– заключение о соответствии программного модуля стандартам качества программного обеспечения</li> <li>– способен производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования</li> </ul>
ПК 3.6. Разрабатывать технологическую документацию	<ul style="list-style-type: none"> <li>– выполненная документация для пользователя программного обеспечения</li> <li>– документация к программному коду</li> <li>– контрольные варианты использования программного обеспечения</li> </ul>

Профессиональные и общие компетенции	Показатели оценки результата
	<ul style="list-style-type: none"> <li>– правильность выбора методов средств разработки программной документации</li> <li>– точность и грамотность оформления технологической документации</li> </ul>

**2.2** В целях овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения **программы учебной практики** должен :

**по ПМ.01 Разработка программных модулей программного обеспечения компьютерных систем:**

**иметь практический опыт:**

ПО 1. Разработка проекта программного продукта согласно постановке задачи пользователя.

ПО 2. Разработки кода программного продукта на основе готовой спецификации на уровне модуля.

ПО 3. Использования инструментальных средств на этапе отладки программного продукта.

ПО 4. Проведения тестирования программного модуля по определенному сценарию.

**уметь:**

У 1. Осуществлять разработку кода программного модуля на современных языках программирования.

У 2. Создавать программу по разработанному алгоритму как отдельный модуль.

У 3. Выполнять отладку и тестирование программы на уровне модуля.

У 4. Оформлять документацию на программные средства.

У 5. Использовать инструментальные средства для автоматизации оформления документации.

**знать:**

З 1. Основные этапы разработки программного обеспечения.

З 2. Основные принципы технологии структурного и объектно-ориентированного программирования.

З 3. Основные принципы отладки и тестирования программных продуктов.

З 4. Методы и средства разработки технической документации

**по ПМ.02 Разработка и администрирование баз данных:**

**иметь практический опыт:**

– работы с объектами базы данных в конкретной СУБД;

– использования средств заполнения базы данных;

– использования стандартных методов защиты объектов базы данных;

**уметь:**

– создавать объекты баз данных в современных СУБД и управлять доступом к этим объектам;

– работать с современными Case-средствами проектирования баз данных;

– формировать и настраивать схему базы данных;

– разрабатывать прикладные программы с использованием языка SQL;

– создавать хранимые процедуры и триггеры на базах данных;

– применять стандартные методы для защиты объектов базы данных;

**знать:**

– основные положения теории баз данных, хранилищ данных, баз знаний.

- основные принципы построения концептуальной, логической и физической модели данных.
- современные инструментальные средства разработки схемы базы данных.
- методы описания схем баз данных в современных СУБД.
- структуры данных СУБД, общий подход к организации представлений, таблиц, индексов и кластеров.
- методы организации целостности данных.
- способы контроля доступа к данным и управления привилегиями.
- основные методы и средства защиты данных в базах данных.
- модели и структуры информационных систем.
- основные типы сетевых топологий, приемы работы в компьютерных сетях.
- информационные ресурсы компьютерных сетей.
- технологии передачи и обмена данными в компьютерных сетях.
- основы разработки приложений баз данных.

### **по ПМ.03 Участие в интеграции программных модулей:**

#### **иметь практический опыт:**

- участия в выработке требований к программному обеспечению;
- участия в проектировании программного обеспечения с использованием специализированных программных пакетов;

#### **уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;
- использовать методы для получения кода с заданной функциональностью и степенью качества;

#### **знать:**

- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;
- основные методы и средства эффективной разработки;
- основы верификации и аттестации программного обеспечения;
- концепции и реализации программных процессов;
- принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения;
- методы организации работы в коллективах разработчиков программного обеспечения;
- основные положения метрологии программных продуктов, принципы построения, проектирования и использования средств для измерений характеристик и параметров программ, программных систем и комплексов;
- стандарты качества программного обеспечения;
- методы и средства разработки программной документации.

### **по ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»:**

#### **иметь практический опыт:**

- работы в операционной системе WINDOWS;
- работы в основных приложениях OFFICE;

#### **знать:**

- основы работы в операционных системах и сервисных оболочках;
- устройство персонального компьютера, работу с его основными и периферийными устройствами;
- структуру основных папок операционной системы WINDOWS;

- основные антивирусные программы;
- основные приёмы работы с папками и файлами;
- стандартные программы операционной системы WINDOWS;
- основные программы - архиваторы;
- основные приёмы работы в локальной и глобальной сети;

**уметь:**

- запускать программы, установленные в операционной системе;
- выполнять основные операции над папками и файлами;
- выполнять поиск информации в компьютере, флеш-картах, картах памяти, оптических носителях, локальной и глобальной сети;
- удалять и устанавливать программное обеспечение;
- обновлять антивирусную программу, проверять диски на вирусы;
- сканировать и форматировать информацию;
- создавать и форматировать презентации;
- устранять сбои и ошибки, возникающие в работе программного обеспечения;
- создавать тесты в электронных оболочках;
- создавать, сохранять, модифицировать, выводить на печать документы, созданные в приложениях WINDOWS;
- изменять настройки окон приложений WINDOWS;
- осуществлять настройку операционной системы WINDOWS

### 3. СТРУКТУРА И СОДЕРЖАНИЕ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ

Тематический план программы производственной практики (по профилю специальности)

Коды профессиональных компетенций	Наименование профессиональных модулей	Всего часов	Распределение часов по семестрам
1	2	3	4
ПК 1.1, ПК 1.4, ПК 3.3	ПМ.04 Выполнение работ по профессии рабочего «Оператор электронно-вычислительных и вычислительных машин»	108	4
ПК 1.1. – ПК 1.6.	ПМ.01. Разработка программных модулей программного обеспечения компьютерных систем	108	5
ПК 2.1. – ПК 2.4.	ПМ.02 Разработка и администрирование баз данных	108	5
ПК 3.1 – ПК 3.6	ПМ.03 Участие в интеграции программных модулей	36	7
	<b>Всего:</b>	<b>360</b>	

### 3.1 ПМ.04 ВЫПОЛНЕНИЕ РАБОТ ПО ПРОФЕССИИ РАБОЧЕГО «ОПЕРАТОР ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНЫХ И ВЫЧИСЛИТЕЛЬНЫХ МАШИН»

#### УП.04.01. УЧЕБНАЯ ПРАКТИКА К МДК.04.01 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ

#### 4 СЕМЕСТР

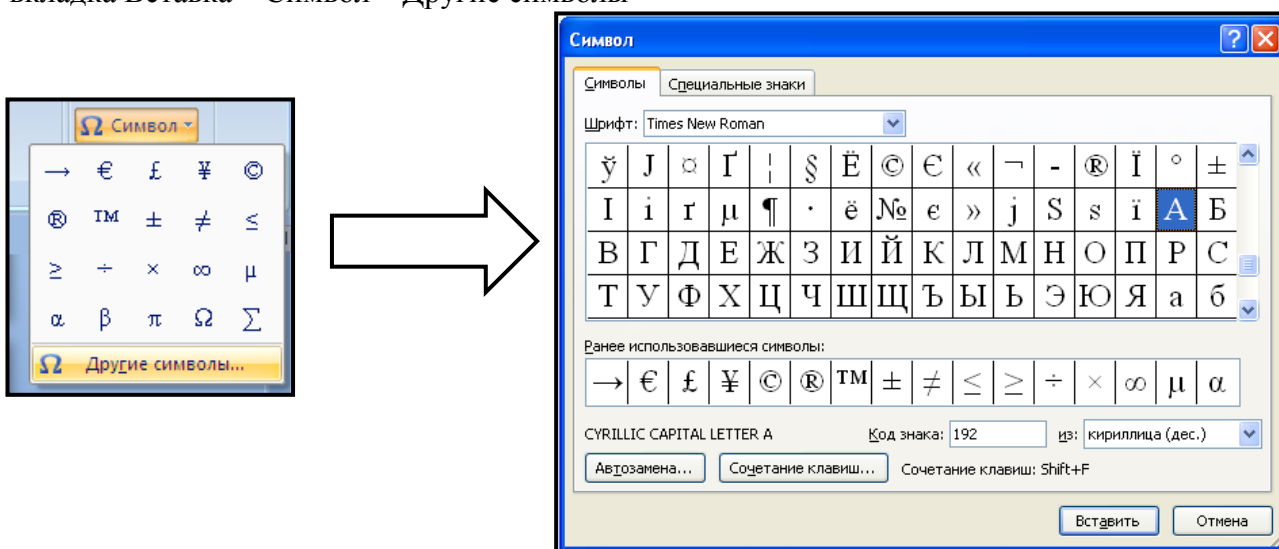
### Практическая работа №1 (ПР1)

**Тема:** Дискретное (цифровое) представление текстовой, графической, звуковой информации и видеоинформации

**Цель:** изучить способы представления текстовой, графической, звуковой информации и видеоинформации, научиться записывать числа в различных системах счисления.

#### Выполнение работы:

**Задание №1.** Используя таблицу символов, записать последовательность десятичных числовых кодов в кодировке Windows для своих **ФИО, названия улицы, по которой проживаете**. Таблица символов отображается в редакторе MS Word с помощью команды: вкладка Вставка→Символ→Другие символы



В поле **Шрифт** выбираете Times New Roman, в поле из выбираете кириллица. Например, для буквы «А» (русской заглавной) код знака– 192.

**Пример:**

И	В	А	Н	О	В
200	194	192	205	206	194

А	Р	Т	Е	М
192	208	210	197	204

П	Е	Т	Р	О	В	И	Ч
207	197	210	208	206	194	200	215

**Задание №2.** Используя стандартную программу **БЛОКНОТ**, определить, какая фраза в кодировке Windows задана последовательностью числовых кодов и продолжить код.

1. Запустить **БЛОКНОТ**.
2. С помощью дополнительной цифровой клавиатуры при нажатой клавише **ALT** ввести код, отпустить клавишу **ALT**. В документе появиться соответствующий символ.

## Выполнение задания №2

<b>Я</b>		<b>У</b>												
<b>0255</b>		<b>0243</b>	<b>0247</b>	<b>0243</b>	<b>0241</b>	<b>0252</b>		<b>0226</b>		<b>0225</b>	<b>0232</b>	<b>0234</b>	<b>0239</b>	<b>0238</b>

<b>0241</b>	<b>0239</b>	<b>0229</b>	<b>0246</b>	<b>0232</b>	<b>0224</b>	<b>0235</b>	<b>0252</b>	<b>0237</b>	<b>0238</b>	<b>0241</b>	<b>0242</b>	<b>0232</b>	

<b>0207</b>	<b>0208</b>	<b>0206</b>	<b>0195</b>	<b>0208</b>	<b>0192</b>	<b>0204</b>	<b>0204</b>	<b>0200</b>	<b>0209</b>	<b>0210</b>			

**Задание №3.** Заполнить пропуски числами:

5 Кбайт	? байт	? бит
---------	--------	-------

6 Мбайт	? байт	? бит
---------	--------	-------

3 Гбайт	? байт	? бит
---------	--------	-------

**Задание №4.** Перевести десятичное число в двоичную систему счисления и сделать проверку:

1.  $438_{10}$

2.  $396_{10}$

**Пример:**

$581_{10}$  перевести в двоичную систему и затем обратно в десятичную

Целая часть от деления	Остаток от деления
$581 / 2 = 290$	$581 \% 2 = 1$
$290 / 2 = 145$	$290 \% 2 = 0$
$145 / 2 = 72$	$145 \% 2 = 1$
$72 / 2 = 36$	$72 \% 2 = 0$
$36 / 2 = 18$	$36 \% 2 = 0$
$18 / 2 = 9$	$18 \% 2 = 0$
$9 / 2 = 4$	$9 \% 2 = 1$
$4 / 2 = 2$	$4 \% 2 = 0$
$2 / 2 = 1$	$2 \% 2 = 0$
$1 / 2 = 0$	$1 \% 2 = 1$

Остаток от деления записываем **в обратном порядке**. Получаем число в **2-ой системе**

**счисления:** 1001000101

Чтобы перевести число  $1001000101_2$  обратно в число 10-й системе:

$$1 * 2^9 + 0 * 2^8 + 0 * 2^7 + 1 * 2^6 + 0 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 512 + 64 + 4 + 1 = 581$$

**Задание №5.** Записать в развернутой форме восьмеричное число и, произведя вычисления, выразить в десятичной системе счисления:



1. 52624<sub>8</sub>
2. 773602<sub>8</sub>

**Пример:**

$$77643_8 = 7 \cdot 8^4 + 7 \cdot 8^3 + 6 \cdot 8^2 + 4 \cdot 8^1 + 3 \cdot 8^0 = 32675$$

**Задание №6.** Ответить на вопросы:

1. Что такое информация?	Информация — это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний
2. Какие виды информации Вы знаете?	
3. Приведите примеры аналогового представления графической информации.	
4. Что такое пиксель?	
5. Что такое система счисления?	
6. Напишите правило перевода десятичных чисел в двоичный код.	
7. Перечислите единицы измерения информации.	

**Задание №7.** Сделать вывод о проделанной лабораторной работе. Что вы изучили?

---



---



---



---

**Практическая работа №2 (ПР2)**

**Тема:** Создание архива данных. Извлечение данных из архива. Атрибуты файла и его объем

**Цель:** изучение принципов архивации файлов, функций и режимов работы наиболее распространенных архиваторов, приобретение практических навыков работы по созданию архивных файлов и извлечению файлов из архивов.

**Выполнение работы:**

**Задание №1.**

1. В своей папке создайте папку с именем **Архивы**. В ней создайте папки **Изображения** и **Документы**.
2. Найдите и скопируйте в папку **Изображения** рисунки с расширениями **\*.jpg**, **\*.bmp** и **\*.gif**.
3. Сравните размеры файлов **\*.bmp**, **\*.gif** и **\*.jpg** и запишите данные в таблицу\_1.
4. В папку **Документы** поместите файлы **\*.doc** (не менее 3) и запишите их исходные размеры в таблицу №1.

**Задание №2.**

Архивация файлов в формате **7z**

1. Выберите файл **Пейзаж1.jpg** нажмите по нему **правой кнопкой мыши** и выберите «**7-Zip/Добавить архив...**»

2. В появившемся окне
3. Введите имя архива в поле **Архив – Пейзаж1.7z** и убедитесь, что в поле **Формат архива** установлен тип **7z**.
4. Установите в поле **Метод обновления**: *Добавить и заменить*
5. В раскрывающемся списке **Уровень сжатия**: выберите пункт **Нормальный**. Запустите процесс архивации кнопкой **ОК**.
6. Сравните размер исходного файла с размером архивного файла. Данные запишите в таблицу №1.
7. Для извлечения файлов из архива создайте папку **Извлеченные** внутри папки **Изображения**. Выделите архив **Пейзаж1.7z** выполните команду «**7-Zip/Распаковать**». В появившемся диалоговом окне выберите папку – **Архивы/Изображения/Извлеченные**.
8. Создайте самораспаковывающийся **7z**-архив. Для этого установите курсор на имя файла **Пейзаж1.jpg**, выполните команду «**7-Zip/Добавить архив...**»»
9. Установите флажок **Создать SFX-архив**.
10. В раскрывающемся списке **Уровень сжатия**: выберите пункт **Нормальный**. Запустите процесс архивации кнопкой **ОК**.
11. Аналогичным образом создайте архивы для файлов **Пейзаж2.gif**, **Пейзаж3.bmp**, **Документ1.doc**, **Документ2.doc**, **Документ3.doc**.
12. Сравнительные характеристики исходных файлов и их архивов занести в таблицу №1.

13.

### Задание №3. Архивация файлов в формате Zip

1. Прделайте архивирование тех же файлов, только в формате архивирования **Zip**
2. Сравнительные характеристики исходных файлов и их архивов занести в таблицу\_1.
3. Создайте самораспаковывающийся **Zip** – архивы, включающие в себя текстовые и графические файлы.
4. Определите процент сжатия файлов и заполните таблицу №1. Процент сжатия определяется по формуле  $P = \frac{S}{S_0} * 100\%$ , где S– размер архивных файлов, S<sub>0</sub>– размер исходных файлов.

**Таблица №1**

	Архиваторы				Размер исходных файлов
	7z	Процент сжатия 7z	Zip	Процент сжатия Zip	
<b>Текстовые файлы:</b>					
1. Документ1.doc					
2. Документ2.doc					
3. Документ3.doc					
<b>Графические файлы:</b>					
1. Пейзаж1.jpg					
2. Пейзаж2.gif					
3. Пейзаж3.bmp					

## Практическая работа №3(ПР3)

**Тема:** Операционная система. Графический интерфейс

**Цель:** закрепить навыки работы с операционной системой Windows, отработать навыки работы с файлами и папками в ОС Windows; научиться выполнять навигацию с помощью левой панели программы ПРОВОДНИК и изучить приемы копирования и перемещения объектов методом перетаскивания между панелями

### Выполнение работы:

#### Задание №1.

Заполнить таблицу:

Выполняемое действие	
1. Открыть Главное меню (Пуск).	
2. Перечислить пункты обязательного раздела Главного меню.	
3. Перечислить пункты произвольного раздела Главного меню.	

#### Задание №2.

Заполнить таблицу:

Выполняемое действие	
1. Открыть Контекстное меню.	
2. Перечислить пункты Контекстного меню, не выделяя объекты.	
3. Перечислить пункты Контекстного меню, выделив какой-либо из объектов. Указать, какой объект выделили.	

#### Задание №3.

Выполняемое действие
1. Создать на рабочем столе папку с именем – номер группы.
2. В созданной папке создать папку с именем – своя фамилия.
3. В папке с именем – своя фамилия создать текстовый документ. Сохранить его под любым именем.
4. Создать на рабочем столе еще одну папку с именем БИК.
5. Скопировать папку – своя фамилия в папку БИК.
6. Переименовать папку – своя фамилия и дать название – свое имя.
7. Создать в папке БИК ярлык на приложение Word.

Изучить структуру **созданных папок** с помощью программы ПРОВОДНИК, схематически отобразить её и подписать все элементы окна.

#### Задание №4.

С помощью программы Проводник создайте следующую структуру каталогов (Рис.1.):

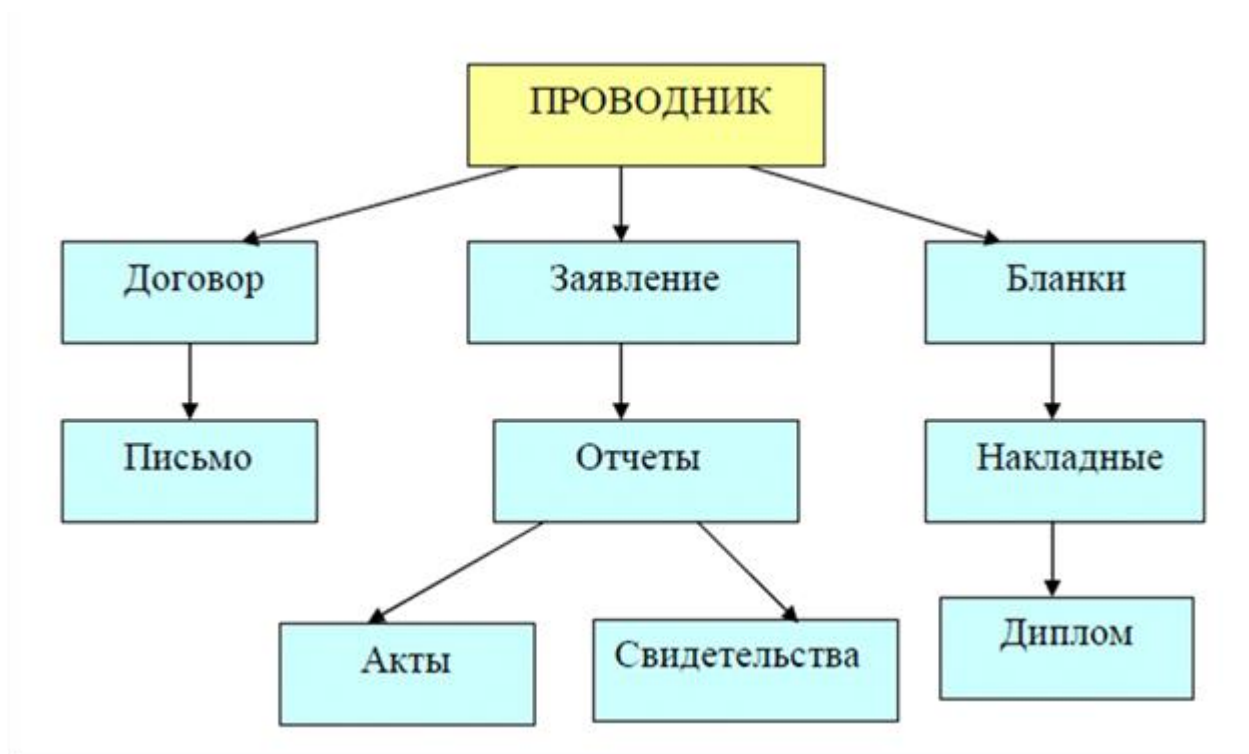


Рис. 1. Структура каталогов

#### Задание №5.

Ответить на вопросы:

1. Что такое файловая структура компьютера?	
2. Для чего предназначен ПРОВОДНИК?	
3. Что отображается на левой панели ПРОВОДНИКА?	
4. Что отображается на правой панели ПРОВОДНИКА?	
5. Для чего предназначено Главное меню (Пуск)?	
6. Как открывается контекстное меню?	
7. Что является средствами управления ОС Windows?	
8. Перечислите основные элементы управления ОС Windows?	
9. Для чего предназначена Корзина?	
10. Перечислите методы сортировки объектов.	

#### Задание №6.

Сделать вывод о проделанной лабораторной работе. Что вы изучили?

---

---

---

---

## Практическая работа №4 (ПР4)

**Тема:** Виды сетей. Разграничение прав доступа в сети, общее дисковое пространство в локальной сети

**Цель:** освоение приемов обмена файлами между пользователями локальной компьютерной сети.

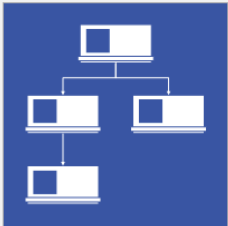
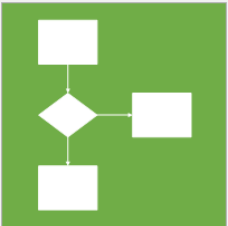
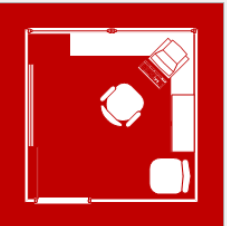

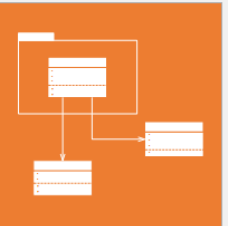
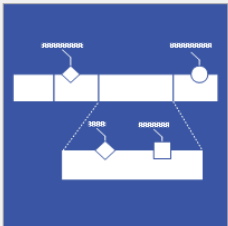

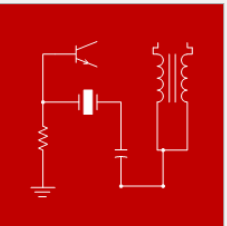
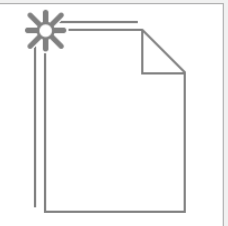
### Выполнение работы:

#### Задание №1.

Лекция ПО ЭВМ №22

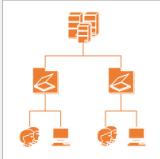
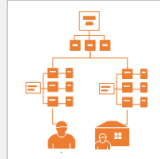
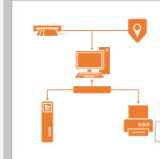
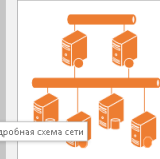
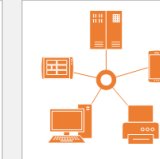
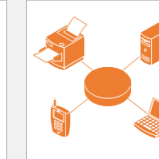
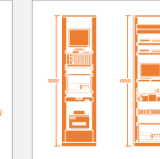
Поиск шаблонов в сети

РЕКОМЕНДУЕМЫЕ КАТЕГОРИИ

				
Бизнес	Блок-схема	Карты и планы этажей	Общие	Программы и базы данных
				
Расписания	Сеть	Техника	Из существующего документа	

Создать

Домой > Сеть

						
Active Directory	Каталог LDAP	Подробная схема сети	Подробная схема сети — объём...	Принципиальная схема сети	Принципиальная схема сети —...	Схема стоек

#### Составьте 3 схемы топологий сетей в Visio:

- Шинная
- Звезда
- Кольцевая

**Задание №2.**

Ответить на вопросы:

1. Укажите основное назначение компьютерной сети.	
2. Что такое локальная сеть, глобальная сеть?	
3. Что понимается под топологией локальной сети?	
4. Какие существуют виды топологии локальной сети?	
5. Охарактеризуйте кратко топологию «шина»	
6. Охарактеризуйте кратко топологию «звезда»	
7. Охарактеризуйте кратко топологию «кольцо».	
8. Перечислите технические средства для создания сетей.	
9. Что такое протокол обмена?	
10. <i>Решите задачу.</i> Максимальная скорость передачи данных в локальной сети 100 Мбит/с. Сколько страниц текста можно передать за 1 сек, если 1 страница текста содержит 50 строк и на каждой строке - 70 символов 1 символ = 1 байт	

**Задание №3.**

<b>Топология локальных компьютерных сетей</b>					
<b>«Шина»</b>		<b>«Звезда»</b>		<b>«Кольцо»</b>	
<b>Достоинства</b>	<b>Недостатки</b>	<b>Достоинства</b>	<b>Недостатки</b>	<b>Достоинства</b>	<b>Недостатки</b>

**Задание №4.**

Сделать вывод о проделанной лабораторной работе:

---



---



---

## Практическая работа №5 (ПР5)

**Тема:** «Знакомство с основными инструментами программы MS Word»

**Цель:** изучить принципы ввода и редактирования текста, операции с фрагментами текста, инструменты форматирования текста.

### Выполнение работы:

#### Задание 1.

Создание и сохранений документа

Выполнение работы:

1. Загрузите процессор Word.
2. Установите режим Разметки документа.
3. Задайте основные параметры (тип и размер шрифта: Times New Roman, размер - 12, выравнивание «По левому краю», левое поле страницы 2,5 см. правое -1,5 см.)
4. Наберите следующий текст:

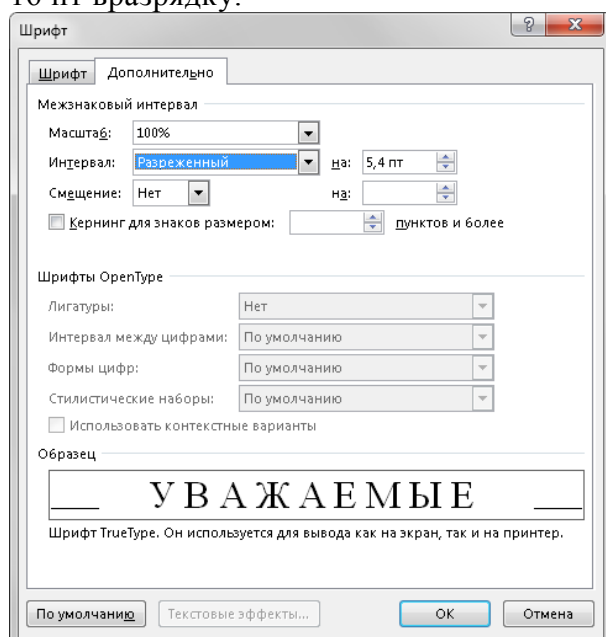
Уважаемые господа! Приглашаем вас на юбилейную презентацию компьютерной фирмы «МАКУЛШИРС». Мы работаем на российском рынке много лет. Информационные продукты нашей фирмы знают и любят многие пользователи страны. Наши компьютеры работают без рекламаций! Будем рады видеть вас. Запомните адрес и время нашей презентации - Компьютерная улица, д. 5, 18-00. Справки по телефону 123-45-67.

5. Замените слово **вас** на слово **Вас**. (через «Правка», «Найти и заменить»),
6. Сохраните набранный текст в файле с именем Текст1. (Для этого: в меню «Файл» введите команду «Сохранить как...», создайте в своей папке папку «Тексты», перейдите в эту папку, задайте в диалоговом окне имя файла «Текст 1» и нажмите кнопку «Сохранить»).

#### Задание 2.

##### Форматирование текста.

1. Оформите в тексте обращение «Уважаемые господа!» полужирным шрифтом размером 16 пт вразрядку.



2. Отцентрируйте абзац - фрагмент текста «Уважаемые господа!».

3. Измените строчные буквы на прописные фрагмент текста «уважаемые господа!» (Shift+F3)
4. Оформите фрагмент текста с указанием адреса фирмы полужирным курсивом размером 12 пт и разбейте этот фрагмент на абзацы.
5. Произведите выравнивание и оформление шрифтами всего текста в соответствии с рисунком.
6. Заключите весь текст в рамку и сделайте фон.
7. Сохраните документ в файле с именем Текст2

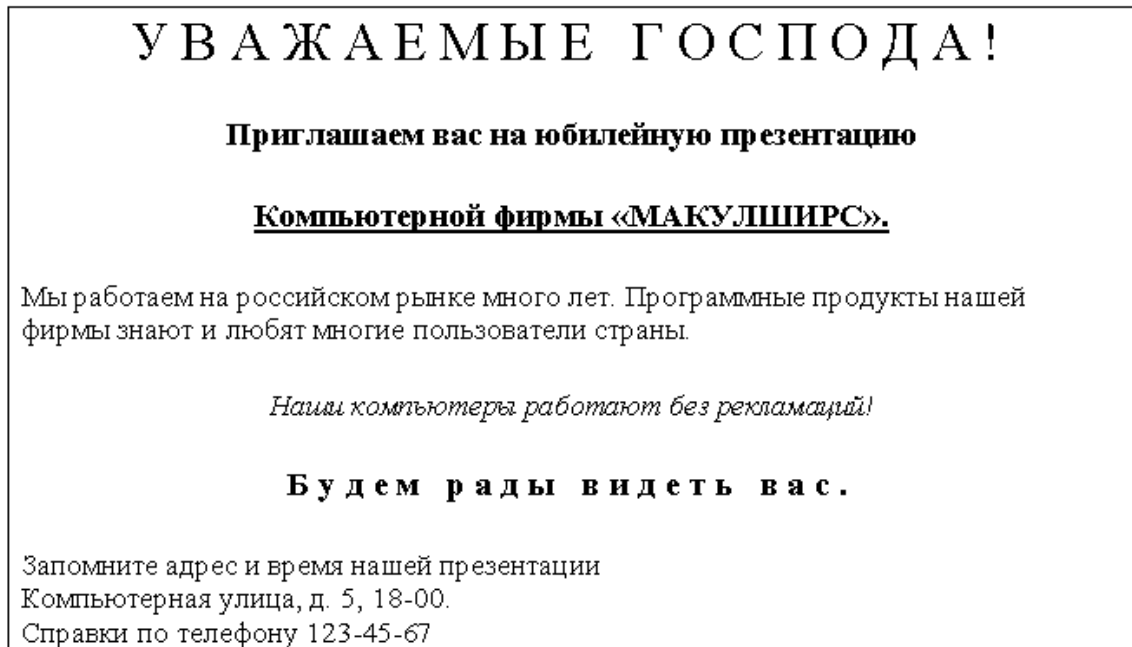


Рис. 1. Отформатированный текст

### Задание 3.

Создайте и отформатируйте документы по варианту:





ЗАКРЫТОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО  
"ТЕРМИКА"

**ПРОТОКОЛ**

24.08.2004

№ 14

Москва

заседания Совета директоров

Председатель – Цицин А.Г.

Секретарь – Прохорова Т.А.

Присутствовали: Воробьев К.В., Сенина Е.О., Ярославцева Е.Н., Ярославцева Т.П.

**ПОВЕСТКА ДНЯ:**

1. Об организации производственной деятельности Управления производства и внедрения информационных систем.

Доклад технического директора Воробьева К.В.

2. О введении системы скидок для вузов на информационные системы по документационному обеспечению управления и архивному делу.

Доклад коммерческого директора Ярославцевой Е.Н.

**1. СЛУШАЛИ:**

Воробьева К.В. – в связи с увеличением объема производственной деятельности и штатной численности Управления производства и внедрения информационных систем на 3 единицы представил подготовленное предложение по улучшению технического обеспечения управления (прилагается).

**ВЫСТУПИЛИ:**

Цицин А.Г. – поддержал предложение Воробьева К.В.

**ПОСТАНОВИЛИ:**

1.1. Произвести закупку 3-х компьютеров необходимой конфигурации для Управления производства и внедрения информационных систем.

1.2. Обеспечить рациональное распределение новых компьютеров между сотрудниками отдела в соответствии с подготовленным предложением.

2. СЛУШАЛИ:

Ярославцеву Е.Н. – текст доклада прилагается.

ВЫСТУПИЛИ:

Воробьев К.В. – поддержал предложение Ярославцевой Е.Н.

Ярославцева Т.П. – привела конкретные данные о применении системы скидок для вузов и учебных заведений.

ПОСТАНОВИЛИ:

Разработать Управлению сбыта информационных систем систему скидок для вузов и учебных заведений на информационные системы по документационному обеспечению управления и архивному делу.

Председатель

Секретарь



А.Г. Цицин

Т.А. Прохорова

## **Вариант 2**

2.12.2013

Письмо просьба

Большая Медведица.  
Ул. Фрунзе 18

Генеральному директору ООО «Защита» Синициной Л.В  
Ул. Генеральская 10

**Уважаемая Людмила Васильевна!**

Прошу Вас принять участие в благотворительной деятельности и рассмотреть вопросо выделении 2000-4000 руб. на поддержание Заповедника Большая Медведица.

Надеемся на сотрудничество.

Главный директор Зайцева Л.В  
Большая Медведица.

Унифицированная форма № Т-8  
 Утверждена Постановлением Госкомстата России  
 от 05.01.04 № 1

**Филиал ФГБОУ ВГУЭС**  
 (наименование организации)

Форма по ОКУД  
 по ОКПО

Код
0301006
12487542

**ПРИКАЗ**  
 (распоряжение)

Номер документа	Дата составления
12-П	28.11.13

**о прекращении (расторжении) трудового договора с работником (увольнении)**

Прекратить действие трудового договора от " 12 " февраля 20 10 г. № 92 ,  
 уволить " 29 " ноября 20 13 г.

(письменное замечание)

☒

**Иванов Иван Иванович**

(фамилия, имя, отчество)

**отдел систематизации**

(структурное подразделение)

**помощник системного администратора**

(должность (специальность, профессия), разряд, класс (категория) квалификации)

Табельный номер
00026

**ст. 77 п.3 ТК РФ (по собственному желанию)**

(основание прекращения (расторжения) трудового договора (увольнения))

Основание (документ,  
 номер, дата):

**личное заявление от 14.12.13**

(заявление работника, служебная записка, медицинское заключение и т.д.)

Руководитель организации

**директор**

(должность)

*Власенко*

(личная подпись)

**А.А. Власенко**

(расшифровка подписи)

С приказом (распоряжением) работник ознакомлен

*Васильева*

(личная подпись)

" 28 " ноября 20 13 г.

Мотивированное мнение выборного  
 профсоюзного органа в письменной форме  
 (от " \_\_\_ " \_\_\_\_\_ 20 \_\_\_ г. № \_\_\_\_\_ ) рассмотрено

## Практическая работа №6 (ПР6)

**Тема:** Лицензионные и свободно распространяемые программные продукты. Организация обновления программного обеспечения.

**Цель:** изучить лицензионные и свободно распространяемые программные продукты; научиться осуществлять организацию обновления программного обеспечения.

### Выполнение работы:

**Задание №1.** Используя лекции ПО ЭВМ или Интернет ответить на следующие вопросы:

1. информация	
2. информационные технологии	
3. информационно-телекоммуникационная сеть	
4. доступ к информации	
5. конфиденциальность информации	
6. электронное сообщение	
7. документированная информация	

**Задание 2.** Изучив лекцию «Программное обеспечение ЭВМ», заполните таблицу:

	Понятие	Значение понятия
1.	Программное обеспечение (ПО) – это	
2.	Утилитарные программы предназначены для	
3.	Программные продукты (ПП) предназначены для	
4.	Системное программное обеспечение включает в себя	
5.	Операционная система предназначена для	
6.	Функции ОС:	
7.	Пакеты прикладных программ (ППП) –это	
8.	К пакетам прикладных программ относят (пример):	

**Задание 3.** Изучив программное обеспечение компьютера, за которым Вы работаете, заполните список:

Перечень программ Microsoft Office

1.

2.

3.

4.

Перечень стандартных программ

1.

2.

3.

4.

**Задание 4.** Ответьте на вопросы:

1. Что такое программное обеспечение компьютера?	
2. Какие программы являются условно бесплатными?	
3. Какие программные средства относят к свободно распространяемым программам?	
4. В чем преимущества лицензионного программного обеспечения?	
5. Какие проблемы могут возникнуть при использовании нелегального программного продукта?	

**Задание 5.** Сделайте вывод о проделанной лабораторной работе:

---

---

---

## Практическая работа №7 (ПР7)

**Тема:** «Представление информации в табличной форме»

**Цель:** научиться создавать, редактировать и форматировать таблицы.

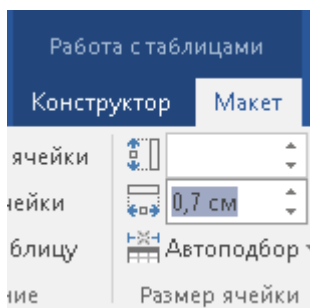
### Выполнение работы

#### Задание 1.

Создание таблицы по размерам столбцов.

При выполнении этого задания рекомендуется придерживаться следующей последовательности действий;

1. Создайте новый документ.
2. Создайте таблицу из 9 столбцов и 14 строк (Рис.1.).
3. Выполните объединение ячеек первой строки.
4. Установите ширину столбца 1 (для строк 2-14) - 0,7 см.



5. Установите ширину столбца 2 (для строк 2-14) - 2,75 см.
6. Установите ширину столбца 3 (для строк 2-14) - 1,5 см.
7. Установите ширину столбца 4 (для строк 2-14) - 1,75 см.
8. Введите текст в ячейки таблицы.
9. Затемните итоговые строки и столбцы для подсчета количества оценок.
10. Сохраните документ в файле Table.doc

Сведения об успеваемости студентов								
	Учебная дисциплина	Группа	Всего сдавало	Отлично	Хорошо	Удовл.	Неудовл.	Неявки
	Информатика							
1.		133	32	12	10	6	3	1
2.		134	27	7	9	6	3	2
3.		135	28	9	8	3	5	<i>n</i>
4.		136	29	8	8	8	3	2
	ИТОГО		116	36	35	23	14	8
	Высшая математика							
1.		133	32	8	12	10	1	1
2.		134	32	12	9	6	3	2
3.		135	31	12	8	3	5	3
4.		136	28	7	8	8	3	2
	ИТОГО		123	39	37	27	12	8

## Задание 2.

Форматирование таблицы, объединение ячеек.

Создайте таблицу для записи расписания занятий на неделю следующего вида:

День		Предмет	День		Предмет
понедельник	1		четверг	1	
	2			2	
	3			3	
	4			4	
	5			5	
	6			6	
вторник	1		пятница	1	
	2			2	
	3			3	
	4			4	
	5			5	
среда	1		суббота	1	
	2			2	
	3			3	
	4			4	
	5			5	

1. Создайте таблицу с количеством столбцов (6) и строк (17).
2. Объединить ячейки для записи дня недели. Сперва выделите необходимые ячейки, а затем выполните команду **Макет** → **Объединить ячейки**
3. Задать вертикальную ориентацию текста. Сперва выделите необходимые ячейки, а затем выполните команду **Макет** → **Выравнивание** → **Направление текста** **необходимое кол-во раз**.
4. Перетащив мышью границы столбцов, задать размеры столбцов.
5. Оформить таблицу, задав границы рамок, как указано в примере таблицы.
6. Используя команду **Главная** → **Границы и заливки**, залить ячейки с записью дней недели избранным вами цветом;

## Задание 3.

Создайте таблицу по варианту:

### Вариант 1

**Таблица 1** Рекомендуемое суточное потребление энергии, белков, жиров и углеводов для взрослого трудоспособного населения различных групп интенсивности труда

Группы труда	Возрастные группы (годы)	Мужчины				Женщины			
		энергия, ккал	белки, г	жиры, г	углеводы, г	энергия, ккал	белки, г	жиры, г	углеводы, г
1.	18-19	2450	72	81	358	2000	61	67	289
	30-39	2300	68	77	335	1900	59	63	274
	40-59	2100	65	70	303	1800	58	60	257
2.	18-19	2800	80	93	411	2200	66	73	318
	30-39	2650	77	88	387	2150	65	72	311
	40-59	2500	72	83	366	2100	63	70	305
3.	18-19	3300	94	110	484	2600	76	87	378
	30-39	3150	89	105	462	2550	74	85	372
	40-59	2950	84	98	432	2500	72	83	366
4.	18-19	3850	108	128	566	3050	87	102	452
	30-39	3600	102	120	528	2950	84	98	432
	40-59	3400	96	113	499	2850	82	95	417
5.	18-19	4200	117	154	586	-	-	-	-
	30-39	3950	11	144	550	-	-	-	-
	40-59	3750	104	137	524	-	-	-	-



## Вариант 2

### Оценки по информатике и математике учеников 7 класса

Ученик	Предмет					
	Информатика			Математика		
	I чет- верть	II чет- верть	1-е полу- годие	I чет- верть	II чет- верть	1-е полу- годие
Баутин Дима	4	5	5	4	4	4
Голубев Миша	4	4	4	3	4	4
Куликов Иван	5	5	5	5	5	5
Радугина Алла	4	5	5	5	5	5

## Вариант 3

### Расписание поезда:

№ поезда	Направление	Отправление		Прибытие			
		дни	время	станция	время	стоимость	
						купе	плацкарт
№49 «Урал»	Н.Тагил- Москва	еж.	6.38	Киров	14.28	116.50	72.20
				Н.Новгород	21.48	196.60	123.50
				Владимир	1.31	230.30	144.70
				Москва	5.10	281.30	177.20
№21 «Кама»	Пермь- Москва	вт, пт, вс	12.10	Киров	15.28	127.10	78.80
				Н.Новгород	22.38	214.40	134.80
				Владимир	2.00	251.30	157.80
				Москва	5.35	306.80	193.50

**Вариант 4****Оценки по информатике и математике  
учеников 7 класса**

Ученик		Предмет	
		Математика	Информатика
Баутин Дима	I четверть	4	4
	II четверть	4	5
	1-е полугодие	4	5
Голубев Миша	I четверть	3	4
	II четверть	4	4
	1-е полугодие	4	4
Куликов Иван	I четверть	5	5
	II четверть	5	5
	1-е полугодие	5	5
Радугина Алла	I четверть	5	4
	II четверть	5	5
	1-е полугодие	5	5

**Вариант 5**

Территория	Месяц года			
	июль		январь	
	Осадки (мм)	Температура (°C)	Осадки (мм)	Температура (°C)
Тульская обл.	85	23	26	-15
Курская обл.	63	27	32	-12
Псковская обл.	78	19	35	-21
Пермская обл.	54	16	45	-27

### Вариант 6

Имя	Рост	Вес
Вася	190	90
Петя	175	80
Галя	168	70
Катя	150	50

№ п/п	Название товара	Характеристики		
		размер	цвет	цена
1	Сапоги	45	чёрный	2000
2	Туфли	42	белый	750
3	Кеды	43	красный	250

3

Итоги продаж за два года, тыс. руб., по годам и кварталам									
Название товара		2000				2001			
		1	2	3	4	1	2	3	4
Мебель	Шкафы	100					250		
	Диваны	120					120		
	Стол		500					600	
Посуда	Тарелки			440		256			
	Чайники		150				325		
	Стаканы		14					12	
	Блюда			12				11	10

### Практическая работа №8 (ПР8)

**Тема:** Использование графических изображений в документах Word.

**Цель работы:** научиться вставлять простейшие графические изображения (фигуры, картинки, рисунки).

**Выполнение работы:**

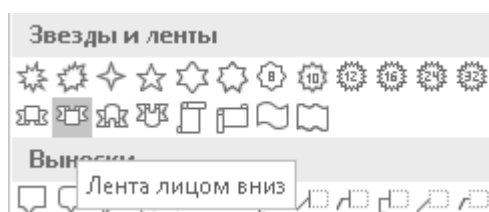
#### Задание 1.

Рисование схемы, используя стандартные графические объекты панели

Освойте технологию в процессе создания схемы, изображенной на рисунке.



1. Создайте новый документ
2. Выберите вкладку Вставка.
3. В группе Иллюстрации. Изобразите название схемы с помощью графического объекта (Фигуры/Звезды и ленты/Лента лицом вниз)



В контекстное меню выберите «Добавить текст», написать название схемы - по центру).

4. Из группы Иллюстрации вставьте рисунок **SmartArt** – Иерархия – Горизонтальная иерархия. Заполните рисунок согласно примеру.
5. Сохраните файл под именем Графика 1

## Задание 2.

Вставка графических объектов

1. Наберите следующий текст:


## Тигр

**Ареал:** В XIX веке тигр обитал в Малой Азии, Месопотамии, Иране, Афганистане, Индии, Непале, Бирме, Индокитае, Таиланде, Индонезии, Китае и Корее. К настоящему времени он истреблен на большей части своего ареала, а там, где еще сохранился, численность его катастрофически сокращается. В Малой, Средней Азии и Казахстане тигр уничтожен полностью. Почти истреблен в Китае, кроме северо-востока и крайнего юга; на большей части Индии, на острове Бали и т.п. Если рассматривать мировой ареал в целом, то тигр, очевидно, истреблен на большей его части.

Тигр - один из самых больших наземных хищников, крупнее их только белые медведи. По общему складу - вполне типичная кошка. Он имеет гибкое тело, на невысоких ногах и с довольно длинным хвостом. Передняя часть корпуса развита сильнее задней, в плечах зверь выше, чем в крестце: создается впечатление некоторой грузности, но вместе с тем и огромной мощи, которая подчеркивается сильными ногами с большими и широкими передними лапами, тяжелой, всегда несколько опущенной головой, высокой холкой и могучей мускулатурой.

2. Вставьте рисунок тигра:
3. Вставить в первый абзац рисунок. Используя маркеры на контурной рамке рисунка, уменьшить его размеры.



4. Изучите назначение кнопок панели **Формат** (Двойной щелчок мыши по рисунку)
5. Используя инструменты панели **Формат**, изменить контрастность и яркость изображения.
6. Щелкнув кнопку **Обтекание текстом** , задать обтекание текстом по контуру рисунка.
7. Щелкнув в контекстном меню на **Формат рисунка**, открыть диалоговое окно **Формат рисунка**, выбрать в нем вкладку **Цвета и**

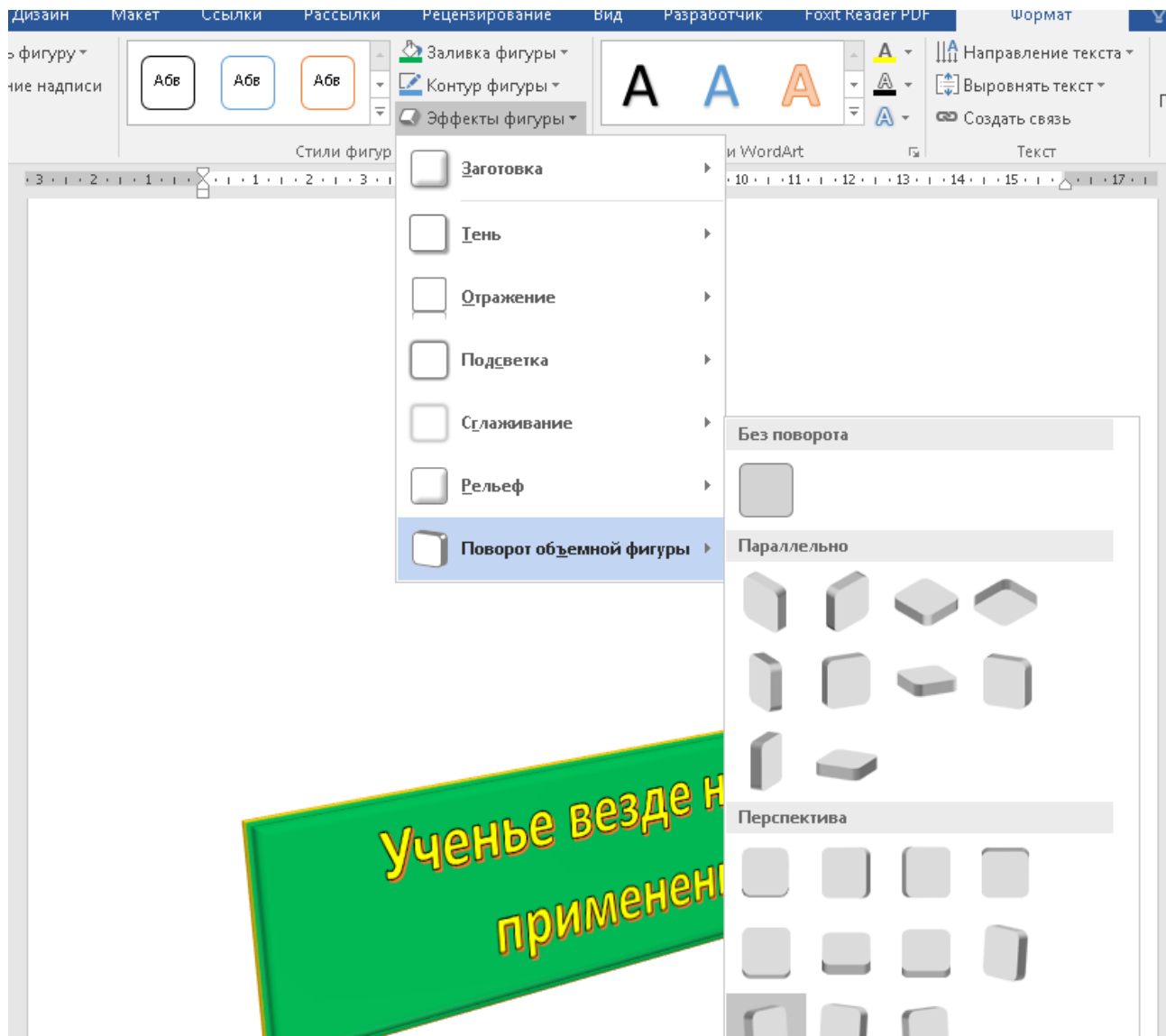
- линии** и задать цвет заливки, включив полупрозрачный режим заливки.
8. Перейти в режим предварительного просмотра документа перед печатью (кнопка Офис, Печать). Просмотреть, как выглядит документ. Закрыть окно просмотра перед печатью.
  9. Уменьшить размеры рисунка.
  10. Чтобы вставить рисунок в текст нужно воспользоваться контекстным меню **Обтекание текстом** и выбрать пункт **Вокруг рамки**.
  11. Выделить рисунок мышью и, прижав левую кнопку мыши, перетащить на середину строки, наблюдая, как текст обтекает рисунок.

### Задание 3.

Вставьте в конец документа объект **WordArt**, задав текст пословицы. Задать нужный эффект.

1. Вставить в конец документа объект **WordArt**, для чего во вкладке **Вставка** выбрать команду – **WordArt** из группы **Текст**.
2. Выбрать стиль надписи, задать текст: *«Ученые везде найдут применение»*; определить шрифт – **Times New Roman**, размер символов – 20 и щелкнуть <ОК>.
3. Используя инструменты во вкладке **Формат**, задать нужный эффект

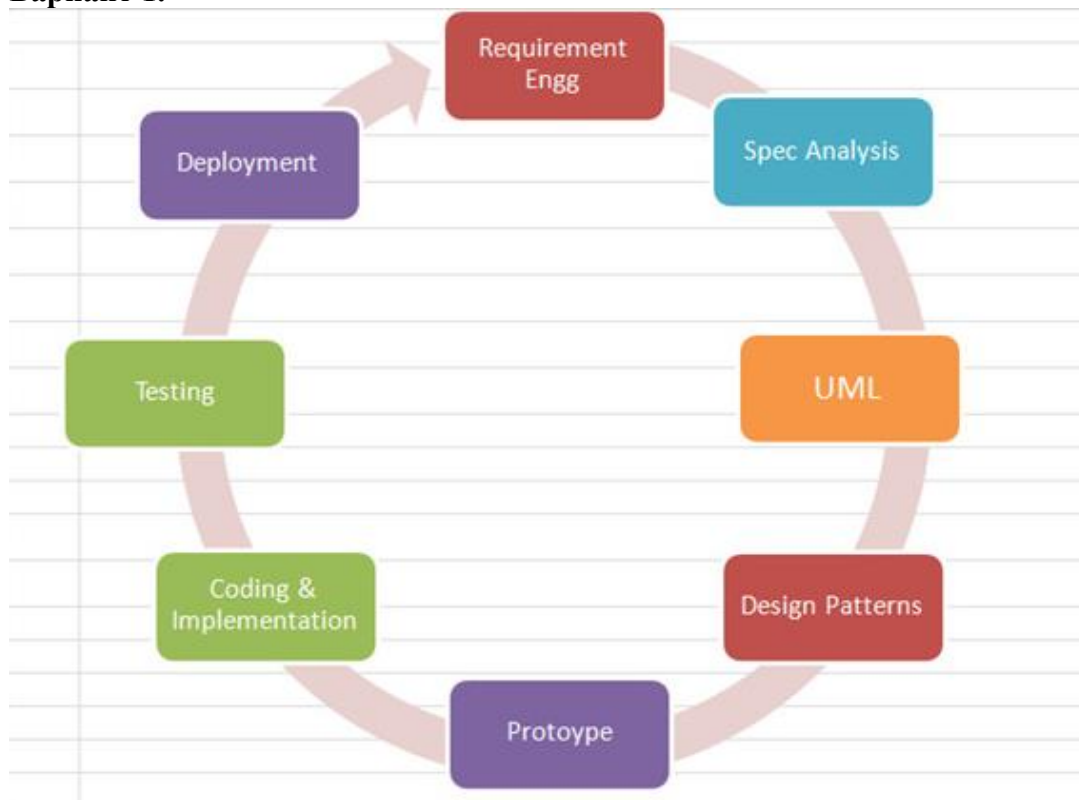
Например:



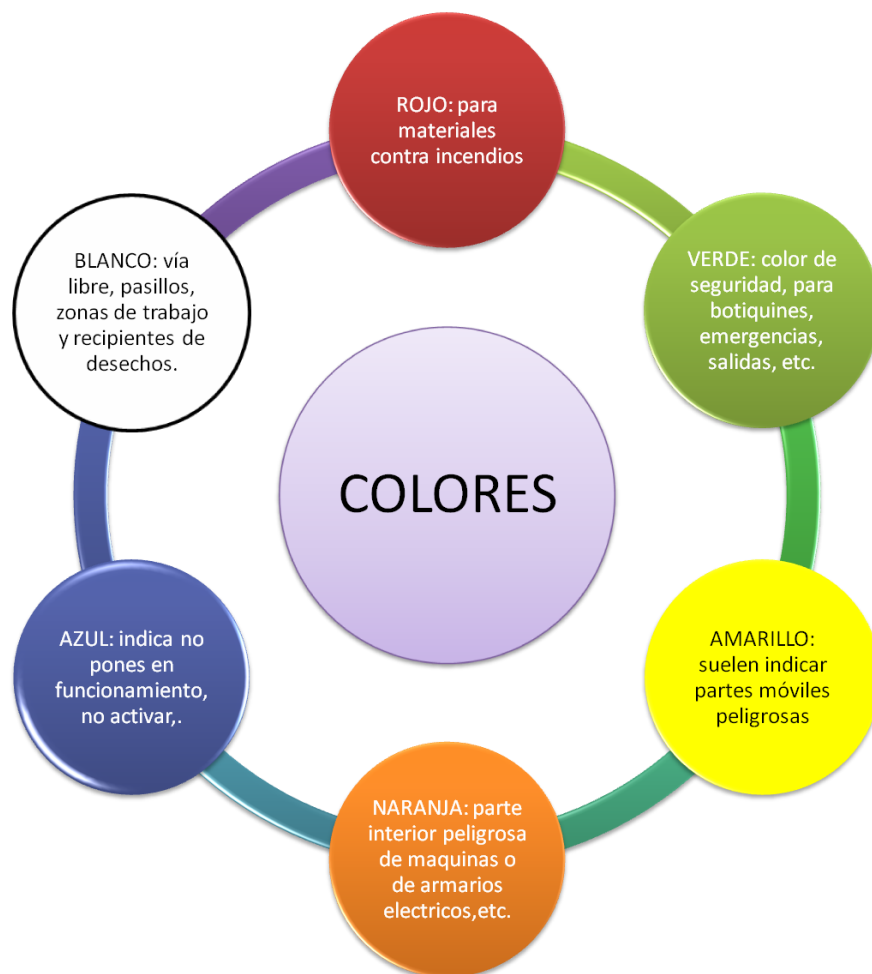
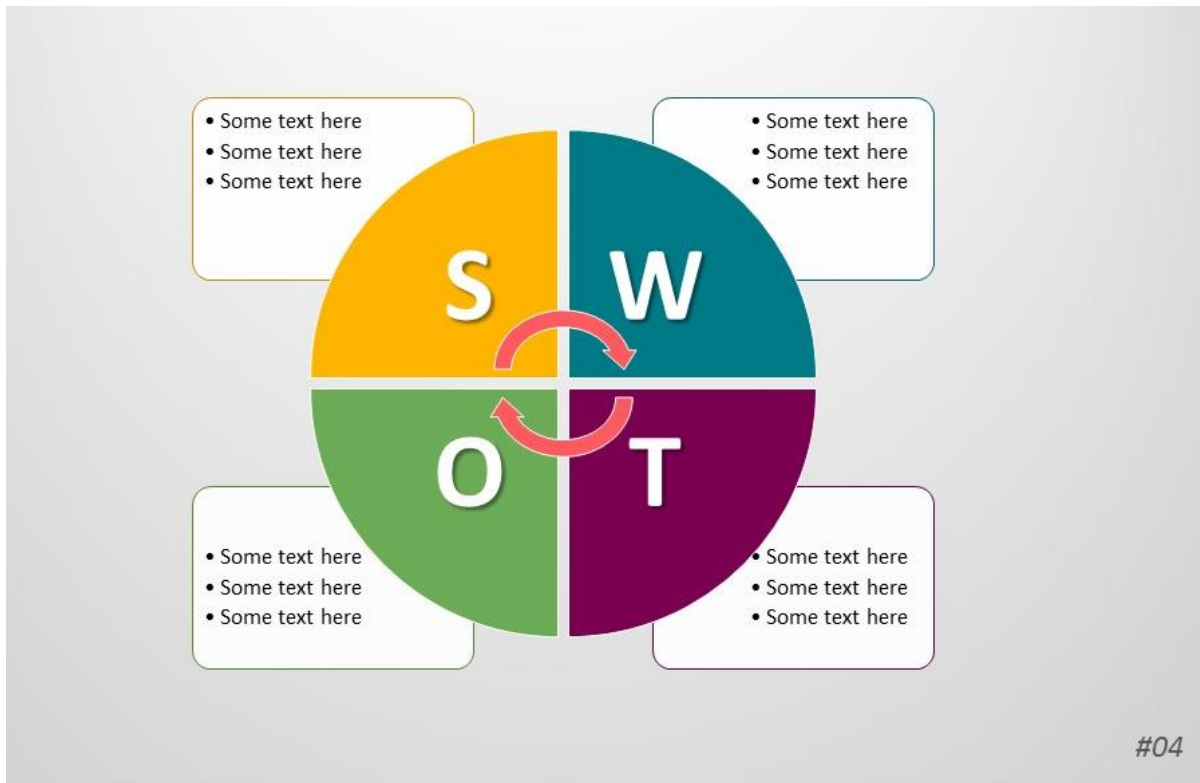
**Задание 4.**

Создайте объекты **SmartArt** по варианту:

**Вариант 1.**

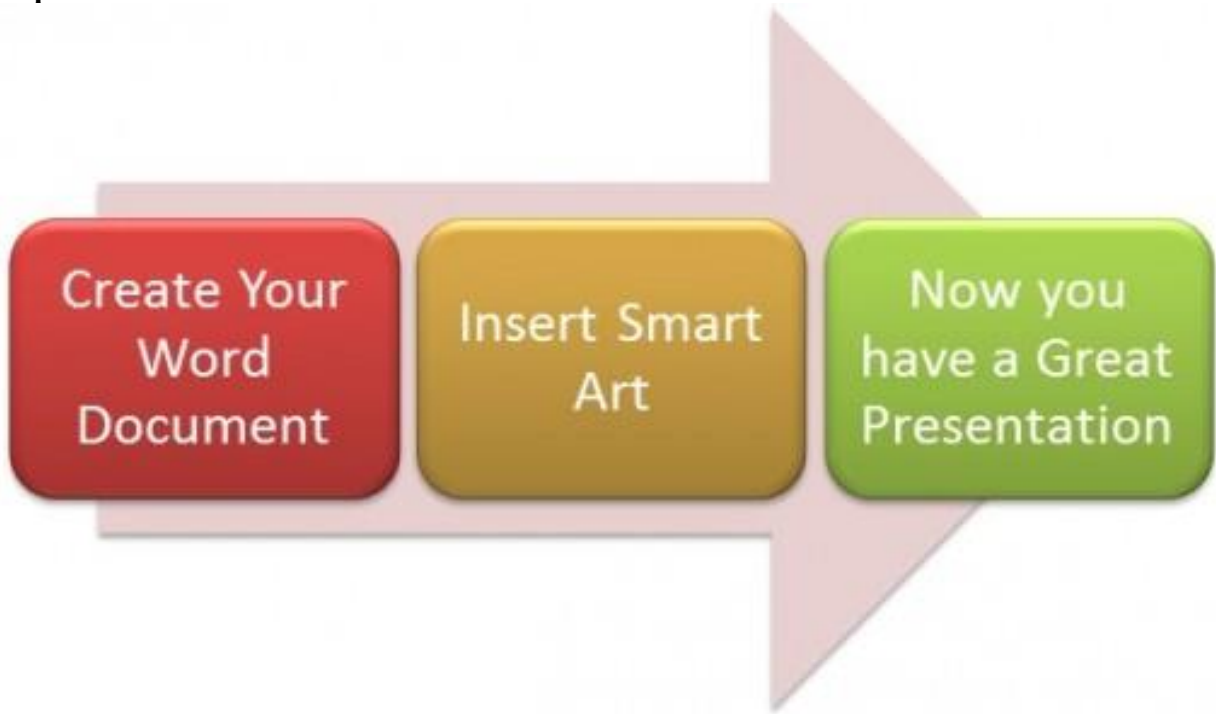


## Вариант 2.





Вариант 3.



## Guiding factors





## Практическая работа №9 (ПР9)

**Тема:** Работа со списками

**Цель работы:** создание маркированных и нумерованных списков.

### Выполнение работы:

#### Задание №1

Создание нумерованного списка

1. Установите режим постраничного отображения документа на экране.
2. Создайте новый документ.
3. Наберите текст, приведенный на рисунке 1.

**Запомните!**

Прежде чем начать форматирование фрагмента текста, надо его выделить.

Форматировать можно с помощью вкладки Главная группы Шрифт, Абзац и Стили.

Форматировать текст - это значит уметь выполнять следующие операции:

устанавливать шрифт, т.е. задавать параметры:

тип,

начертание,

размер,

подчеркивание,

цвет;

определять эффекты в шрифтах:

верхний индекс,

нижний индекс,

зачеркнутый,

утопленный,

приподнятый

и пр.,

Поставлять межсимвольные интервалы;

устанавливать межстрочные интервалы;

делая первую строку абзаца красной или висячей и еще множество других операций, с которыми можно познакомиться в меню **Формат**.

Рисунок 1. – исходный вид документа

4. Выделите шрифтом заголовков (строка 1).
5. Оформите строки 2-3 как нумерованный список.
6. Установите полужирный шрифт в строке 4.
7. Оформите строку 5 текста как маркированный список 1-го уровня.
8. Оформите строки 11 и 18-21 как маркированный список 1-го уровня.
9. Оформите группу строк 6-10 и 12-17 маркированным списком 2-го уровня.
10. Измените вид маркированного списка 2-го уровня для группы строк 6-10 и 12-17
11. Сохраните его как нумерованный список (Текст4)

**Запомните!**

1. Прежде чем начать форматирование фрагмента текста, надо его выделить.
2. Форматировать можно с помощью вкладки Главная группы Шрифт, Абзац и Стили.

**Форматировать текст - это значит уметь выполнять следующие операции:**

- устанавливать шрифт, т. е. задавать параметры:
  - тип,
  - начертание,
  - размер,
  - подчеркивание,
  - цвет;
- определять эффекты в шрифтах:
  - верхний индекс,
  - нижний индекс,
  - зачеркнутый,
  - утопленный,
  - приподнятый
  - и пр.,
- Постановлять межсимвольные интервалы;
- устанавливать межстрочные интервалы;
- делать первую строку абзаца красной или висячей и еще множество других
- операций, с которыми можно познакомиться в меню Формат.

Рисунок 2. – итоговый вид документа

**Задание 2.**

1. Создание маркированного списка
2. Откройте предыдущий список в и отформатируйте по следующему примеру:

**Запомните!**

1. Прежде чем начать форматирование фрагмента текста, надо его выделить.
2. Форматировать можно вкладки Главная группы Шрифт, Абзац и Стили.

**Форматировать текст — это значит уметь выполнять следующие операции:**

- устанавливать шрифт, т. е. задавать параметры:
  - тип,
  - начертание,
  - размер,
  - подчеркивание,
  - цвет;
- определять эффекты в шрифтах:
  - верхний индекс,
  - нижний индекс,
  - зачеркнутый,
  - утопленный,
  - приподнятый
  - и пр.,
- устанавливать межсимвольные интервалы;
- устанавливать межстрочные интервалы;
- делать первую строку абзаца красной или висячей
- и еще множество других операций, с которыми можно познакомиться во вкладке Главная.

3. Сохраните файл под именем Текст 5.

### Задание 3.

Создание нумерованного списка

В данном подразделении имеется 5 персональных компьютеров следующих характеристик:

1) Состав системного блока:

- Системные платы: Intel LGA 775.Socket;
- Звуковые карты: встроенные Realtek;
- Сетевые адаптеры: встроенные Intel;
- Процессор: Intel Core 2 Duo E7500 с тактовой частотой 2.93 ГГц;
- Оперативная память: Объем 2 Гб;
- Видеокарта: Intel G45/G43 Express chipset.

– Привод для дисков: DVD-ROM

– Жесткий диск: SATA-II на 500 Гб

2) Периферийные устройства:

- Монитор: Flatron L170S LG.
- Мышь и клавиатура: Стандартные
- Принтер: Canon LBP 9000;

### Задание 4.

Создание многоуровневого нумерованного списка.

Устройства современного компьютера:

1. Процессор
2. Память
  - 2.1. Оперативная память
  - 2.2. Долговременная память
    - 2.2.1. Жесткий магнитный диск
    - 2.2.2. Дискета
    - 2.2.3. Флэш-память
    - 2.2.4. Оптические диски
      - 2.2.4.1. CD
      - 2.2.4.2. DVD
3. Устройства ввода
  - 3.1. Клавиатура
  - 3.2. Мышь
  - 3.3. Сканер
  - 3.4. Графический планшет
  - 3.5. Цифровая камера
  - 3.6. Микрофон
  - 3.7. Джойстик
4. Устройства вывода
  - 4.1. Монитор
    - 4.1.1. Жидкокристаллический монитор
    - 4.1.2. Монитор на электронно-лучевой трубке
  - 4.2. Принтер
    - 4.2.1. Матричный принтер
    - 4.2.2. Струйный принтер
    - 4.2.3. Лазерный принтер

## Практическая работа №10 (ПР10)

**Тема:** Создание сносок и колонок

**Цель:** Научиться оформлять сложные документы

### Выполнение работы:

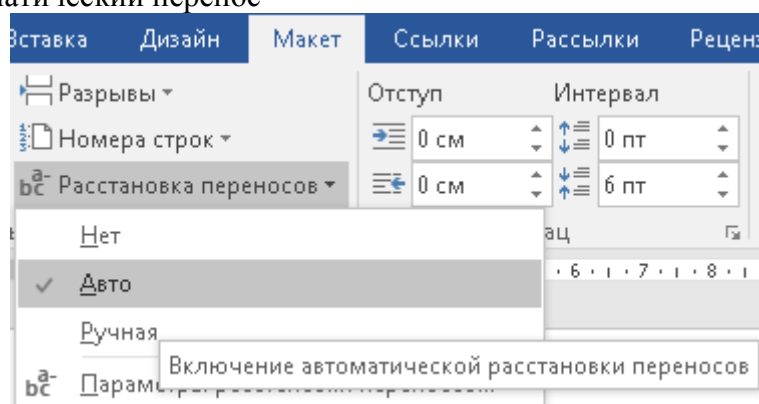
#### Задание 1.

Создание сносок.

4. Установите следующие параметры абзаца:

- межстрочное расстояние – 1,5 строки
- отступ первой строки — 1,25
- отступ: после абзаца – 6 пт.

5. Включите Автоматический перенос



6. Наберите предлагаемый текст шрифтом Times New Roman размером 12 пт.

Из книги В. Язвицкого «Иван III – государь всея Руси»

В трапезной великого князя, перед тем как сесть за стол, поднес владыка Феофил и дар государю панагию, обложенную золотом и жемчугом, кубок, чарку сердоликовую, окованную серебром, и хрустальный бочонок да мису, серебрянную весом в двенадцать гривенок и двести золотых корабленников.

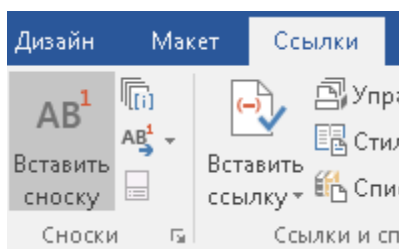
Прислал король и подарки невесте: дойное ожерелье из багряных и синих яхонтов и серьги золотые с такими же камнями драгоценными.

Решено было совершить обручение в базилике св. Петра при участии все прелатов и с большим торжеством. Избрать для поездки в Москву и царевной папским легатом епископом Антонио Бонумбре ... Поставлено было так же выдать епископу на дорожные расходы шестьсот дукатов, а царевне на те же цели папа хотел назначить свыше четырёх тысяч дукатов.

7. В первой строке текста «Из книги...» установите выравнивание по Правому краю

8. Используя панель «Навигация» (Ctrl+F) найдите слово «панагия». Выделите его и измените шрифт на «Полужирный».

9. Не снимая выделения вставьте сноску (Ссылки/Вставить сноску)



10. В конце документа должна появиться цифра 1. Введите после нее определение слова «панагия».

<sup>1</sup> ПАНАГИЯ - круглая икона Божьей матери, носимая на груди как знак архиерейского достоинства

11. Аналогично найдите остальные определения и создайте под ними сноски

---

<sup>1</sup> ПАНАГИЯ - круглая икона Божьей матери, носимая на груди как знак архиерейского достоинства.

<sup>2</sup> СЕРДОЛИК - (карнеол), минерал, розовая или красная разновидность халцедона.

<sup>3</sup> ГРИВЕНКА - гривна, денежная и весовая едагоша в Древней Руси, слиток серебра в % фунта (XIV в.)

<sup>4</sup> КОРАБЛЕННИК - английские золотые монеты - «нобили» - с изображением корабля.

<sup>5</sup> ЯХОНТ - яхонт красный - рубин, яхонт синий - сапфир.

<sup>6</sup> БАЗИЛИКА - древняя католическая церковь. В данном случае церковь св. Петра, построенная в IV в.

<sup>7</sup> ПРЕЛАТ - представитель высшего католического духовенства.

<sup>8</sup> ЛЕГАТ - посол от папы, кардинал или епископ,

<sup>9</sup> ДУКАТ - старинная золотая монета ценой около 3 рублей золотом по курсу XIX в.

12. Установите для текста сносок:

- шрифт – Times New Roman 10 пт
- одинарный отступ
- перед абзацем 0 пт
- после абзаца – 6 пт

13. Подпишите текст в конце области сносок, указав фамилию, группу, дату выполнения

14. Сохраните документ под именем **Сноски.doc**

## Задание 2.

Создайте 2 страницы документа со сносками

---

Проблема отмывания грязных денег давно перестала быть проблемой отдельного региона или отдельно взятой страны.<sup>1</sup> Террористические акции в США, Великобритании, России убедительно доказали это. При этом, по мнению ученых, занимающихся проблемами отмывания грязных денег, сам процесс не может происходить без участия в нем банковской системы, которая может принимать как пассивную, так и активную роль. При пассивном участии банк выступает средством для отмывания денег в руках преступником, ключевым звеном.<sup>2</sup> При активном участии отмывание происходит при содействии банковского учреждения либо определенных работников этого банка.

---

<sup>1</sup> Богатин Ю.В., Швандар В.А. Инвестиционный анализ: Учебное пособие для вузов по экономическим специальностям. – М. : ЮНИТИ, 2007. – 286 с.

<sup>2</sup> Бизнес-план инвестиционного проекта: Отечественный и зарубежный опыт: Учебное пособие для экономических вузов / Под ред. В.М. Попова. – М.: Финансы и статистика, 2007. – 431 с.

---

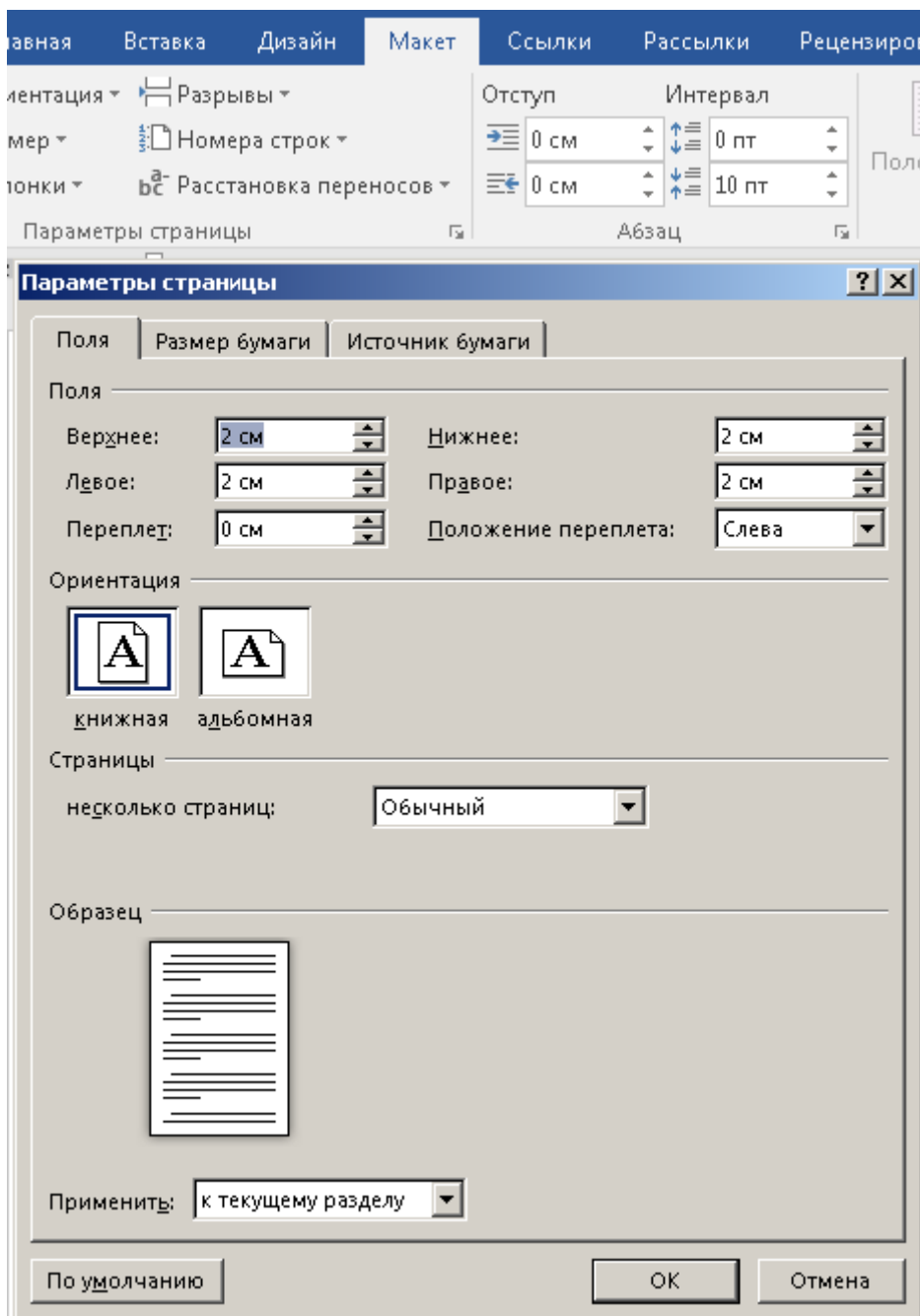
Разработка эффективных средств борьбы в банковской системе можно условно разделить на два уровня, к первому уровню отнести разработку нормативно правовой базы, которая поможет эффективно противодействовать преступникам, внедрение унифицированных правил бухгалтерского учета в банках и унификация международных правил, которая позволит оградить банки с более жесткими правилами от оттока клиентов.<sup>1</sup>

---

<sup>1</sup> Бизнес-план инвестиционного проекта : Учебник / Под ред. Попова В. М. – Москва.: Финансы и статистика, 2008. – 670 с

### Задание 3

#### Создание колонок



1. Установите в параметрах страницы все поля = 2 см.
2. Установите межстрочное расстояние - полупетух. Размер шрифта 12.
3. Введите заголовок документа.
4. Установите разрыв «Без разрыва»
5. Установите режим ввода в две колонки (**Макет/Колонки/Две**)
6. Согласно рисунку, представленному ниже введите текст из первой колонки
7. Установите разрыв колонки (**Макет/Разрывы/Колонка**), чтобы начать печатать текст во второй колонке.
8. После того как напечатаете текст 2-ой колонки, установите разрыв «Без разрыва» (**Макет/Разрывы/Без разрыва**). После этого на следующей строке вы вернетесь в первую колонку.
9. Установите режим ввода в три колонки и аналогичным образом введите оставшийся текст.

## Солнце

Солнце – центральное тело солнечной системы. Это звезда, имеющая средний размер, среднюю температуру и среднюю яркость. Источником энергии Солнца

является непрерывная последовательность термоядерных реакций. Температура поверхности Солнца – порядка 6000 градусов

Светящийся участок Солнца называют фотосферой. Над фотосферой расположена хромосфера, наблюдать которую невооруженным глазом можно только во время полного солнечного

затмения. Над хромосферой расположена «корона», наблюдать которую невооруженным глазом можно тоже лишь в период полного солнечного затмения.

Солнечные пятна – наиболее темные участки солнечной поверхности, имеющие нерегулярную форму. Обычный срок жизни группы солнечных пятен составляет много лет.

Солнце – одна из звезд нашей Галактики (Млечный Путь) и единственная звезда Солнечной системы.

### Задание 4

Создайте колонки с текстом по варианту:

#### Вариант 1.

##### Число 6

*Пифагор считал 6 удивительным числом, так как оно обладает замечательным свойством: получается в результате сложения или перемножения всех чисел, на которые делится. Шестерка делится на 1, 2, 3. И если сложить или перемножить эти числа, то вновь получится 6:  $1+2+3=1*2*3=6$ . Таким свойством не обладает ни одно другое число.*

#### Вариант 2.

##### Число 8

*Это число древние считали воплощением надежности, доведенной до совершенства. Разделенное пополам, оно имеет равные части (4 и 4). Если его еще разделить, то части тоже будут равными (2,2,2,2).*



### Вариант 3.

<i>Число 7</i>		
<i>Особенно большим почетом в древности была окружена семерка. Отголоски почитания этого числа дошли и до наших дней, когда мы употребляем в речи пословицы и поговорки типа «Семь бед — один ответ», .</i>	<i>Еще в Древнем Вавилоне были известны семь планет, к которым причисляли тогда и Солнце, и Луну. Все непонятные явления природы приписывались богам, и постепенно представление о богах соединилось и с семью планетами. По ним стали считать и время. Так родилась семидневная неделя.</i>	<i>Названия дней связаны с именами богов. Во многих языках эти названия остались до сих пор: вторник у французов — марди (день Марса); среда — меркроди (день Луны); воскресенье у немцев — зонтаг (день Солнца).</i>

### Практическая работа №11 (ПР11)

**Тема:** Табличный редактор Excel. Ввод и редактирование данных.

**Цель:** Изучение основных приемов работы с электронными таблицами (ввод текста, чисел и формул, обрамление таблицы, выравнивание текста по центру выделения)

#### Выполнение работы:

##### Задание 1.

Заполнить таблицу и выполнить необходимые вычисления

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	
1	<b>Продажа мороженого по округам города (млн. руб.)</b>							
2	<b>Лето 2000 года</b>							
3								
4								
5			<b>Июнь</b>	<b>Июль</b>	<b>Август</b>			
6	<b>Центральный</b>		140	160	120			
7	<b>Западный</b>		85	80	100			
8	<b>Северный</b>		120	135	140			
9	<b>Южный</b>		110	115	105			
10								
11								

На основании этих исходных данных необходимо найти *производные* величины:

1. Сумму выручки по городу за каждый месяц, (ячейки C10, D10, E10).
2. Сумму выручки по каждому округу за все лето. (F6, F7, F8, F9).
3. Общую сумму выручки. (F10).
4. Процент выручки по каждому округу относительно общей суммы (G6, G7, G8, G9).

**Порядок расчетов:**

- 1) В C10 ввести формулу: =C6+C7+C8+C9 или воспользоваться формулой суммы: =СУММ(C6:C9). Подобным образом работать с ячейками D10, E10.
- 2) В F6 ввести формулу =СУММ(C6:E6). Подобным образом работать с ячейками F7, F8, F9.
- 3) В F10 ввести формулу либо =СУММ(F6:F9) либо =СУММ(C10:E10).
- 4) Вместо процента вначале подсчитать долю каждого округа: В ячейку G6 ввести формулу — =F6/F10, в ячейку G7 формулу =F7/F10 и т.д. (ячейки G8 и G9).

**Задание 2**

Создать и отформатировать таблицу по образцу:

	<b>A</b>	<b>D</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>F</b>
1	<b>ВЕДОМОСТЬ</b>						
2	<hr/>						
3	<i>№№</i>	<i>Фамилия</i>	<i>Зарплата</i>	<i>Премия</i>	<i>ДВ 30%</i>	<i>Налог</i>	<i>Доход</i>
4							
5	1	Иванов НИ.	1381	100			
6	2	Сухов А.С.	1325				
7	3	Михайлов ММ.	1131				
8	4	Абрамов А. А.	1410				
9	5	Петров Р.Р.	1242				
10	6	Веселое В.Б.	1239				
11	<i>Суммарное значение</i>						
12							
13	<i>Среднее значение</i>						
14	<i>Наиб. значение</i>						
15	<i>Наим. значение</i>						

12. Заполнить столбец **Премия**, используя **автозаполнение** арифметической прогрессией с шагом 15
13. Для расчета **ДВ** (районный коэффициент) используйте формулу:  
**(Зарплата + Премия) \* процентная ставка ДВ.**
14. Налог рассчитайте по формуле: **(Зарплата + Премия + ДВ) \*13%**
15. Рассчитайте **Доход: (Зарплата + Премия + ДВ) - Налог**
16. Найдите Среднее значения для каждого столбца
17. Найдите Наибольшее и Наименьшее значения, используя необходимые функции для каждого столбца
18. Для числовых значений столбцов Зарплата, Премия, Дальневосточный коэффициент, Налог, Доход установите денежный формат
19. Ниже таблицы укажите группу, Фамилию и Дату выполнения задания.

## Практическая работа №12 (ПР12)

**Тема:** Табличный редактор Excel. Решение задач.

**Цель:** Изучение встроенных функций Excel

### Выполнение работы:

#### Задание 1.

1. Составить таблицу, содержащую следующие поля:

№	Месяц	ФИО	Полный долг	Тип льготы	Льгота	Внесено	Оставшийся долг

2. Заполнить произвольно таблицу в соответствии с условиями:

- Записей в таблице — 12.
- Различных месяцев — 3.
- Различных фамилий — 4, фамилии повторяются в каждом месяце.
- Значения столбца «Полный долг» задаются произвольно.

3. Столбец «Тип льготы» заполняется цифрами (0,1,2) в зависимости от типа льготы

4. Чтобы вычислить столбец «Льгота»:

- для 0 (нет льгот) будет 0,
- для 1 — скидка в оплате 25%, т.е «Льгота» = «Полный долг» \* 25/100
- 2 — скидка в оплате — 50%

**Все это проще выполнить с помощью функции ЕСЛИ:**

$$\text{ЕСЛИ}(E2=0;0;\text{ЕСЛИ}(E2=1;D2*25/100;D2*50/100)) \quad (1)$$

Данная функция представляет из себя следующий синтаксис:

= **ЕСЛИ (Условие; Значение если условие верно; Значение если условие неверно)**

Как видно из формулы (1) функцию ЕСЛИ можно комбинировать вместе необходимое количество раз.

5. Значения столбца «Внесено» задайте произвольно. Внесено должно быть меньше чем 25% от «Полного долга»

6. «**Оставшийся долг**» = «**Полный долг**»-«**Льгота**»-«**Внесено**»

7. В свободную ячейку вычислить сумму всех «Внесенных оплат» и в другую свободную ячейку сумму всех «Оставшихся долгов».

8. Вычислить среднее значение по полю «Оставшийся Долг».

9. Используя стандартные функции (МАКС, МИН, СРЗНАЧ) найти максимальное, минимальное и среднее значение «Оставшегося долга» в отдельные ячейки.

10. Вычислить сумму «Оставшегося долга» по каждой фамилии и оформить в виде таблицы со столбцами «ФИО», «Оставшийся Долг» в ячейки **J2:K5**

Например, чтобы просуммировать все долги по фамилии «Иванов» можно использовать следующую формулу:

=СУММЕСЛИ(\$C\$2:\$C\$13;"Иванов";\$H\$2:\$H\$13)

Просуммируйте остальные фамилии и оформите по этим данным таблицу

11. Используя функцию СЧЁТ, определить количество абонентов с долгом.

**СЧЁТ(K2:K5)**

12. Используя функцию СЧЁТЕСЛИ, определить количество абонентов с долгом больше 2000

**СЧЁТЕСЛИ(K2:K5;">2000")**

13. Аналогично вычислить сумму «Внесено» и «Оставшийся долг» по месяцам и оформить в виде таблицы в соответствии с образцом:

Месяц	Внесено	Оставшееся оплаты

14. Построить круговые диаграммы для «Внесено» и «Оставшийся долг» по месяцам. Не забудьте их подписать.

## Задание 2.

Составить таблицу по варианту:

### Вариант 1.

Составить таблицу проката транспорта, содержащую следующие поля:

№ п/п	Наименование транспорта	Стоимость проката в сутки	Дата выдачи	Дата сдачи

**Заполнить таблицу в соответствии с условиями:**

- Записей в таблице — 10.
- Различных наименований проката — 4.
- Значения в столбце «Дата сдачи» вводятся меньше чем текущая дата. (не более 15 дней)

### Выполнить:

1. Вставить в таблицу столбец «Дней в прокате». Заполнить данный столбец данными не более чем: **Текущая дата – Дата выдачи – 2.**

2. Вставить в таблицу столбец «Просрочено дней» и вычислить:

**«Просрочено дней» = «Дата сдачи» - «Дата выдачи» - «Дней в прокате».**

3. Вставить столбец «Пеня». Вычислить сумму пени в соответствии со следующим условием, используя функцию ЕСЛИ

- если количество просроченных дней меньше 5, то **Пеня = 0,5** за просроченный каждый день,
- от 5 до 10 дней – **1** за каждый просроченный день

- более 10 дней – 3 за каждый просроченный день

4. Дополнить таблицу столбцом «К оплате» и вычислить с учетом пени:

$$\text{К оплате} = \text{Стоимость в 1д.} * \text{Дней в прокате} * (100 + \text{Пеня}) / 100$$

5. Используя функцию СЧЕТЕСЛИ, определить количество просроченных заказов.
6. Используя стандартные функции, найти минимальное и максимальное значения дней проката.
7. Вычислить сумму к оплате по наименованиям транспорта и оформить в виде таблицы в соответствии с образцом:

Наименование	К оплате

8. Построить диаграмму по наименованиям.

## Вариант 2.

Составить таблицу, содержащую следующие поля:

№ п/п	ФИО	Дата Рождения	Месяц выплаты пособия

**Заполнить таблицу в соответствии с условиями:**

- Записей в таблице — 10.
- Различных месяцев — 3.
- Различных фамилий — 4, фамилии повторяются в каждом месяце.
- Значения в столбце «Дата рождения» вводятся относительно Текущей даты таким образом, чтобы были дети в возрасте:
  - меньше 3 лет,
  - до 8 лет,
  - от 8 до 17 лет
  - старше 17 лет.
- Минимальный заработок задается произвольно в свободной ячейке вне таблицы.

**Выполнить:**

1. Вставить в таблицу столбец «Возраст» и вычислить возраст ребенка.

$$\text{Возраст} = \text{Текущая дата} - \text{Дата рождения}$$

2. Дополнить таблицу столбцом «Процент мин. заработка». С помощью функции ЕСЛИ и условий ниже вычислить значения данного столбца:
- если возраст меньше 3 лет, то 100% от мин. заработка,

- если возраст меньше 8 лет, то 80%,
- от 8 до 17 лет — 60%,
- старше 17 лет — пособия нет.

3. Дополнить таблицу столбцом «Сумма пособия». Значения в столбце вычислить следующим образом:

$$\text{«Сумма пособия»} = \text{«Процент мин. заработка»} * \text{«Мин.Зарботок»}$$

4. Вычислить сумму пособий всех детей.
5. Вычислить средний возраст детей.
6. Используя функцию **СЧЕТЕСЛИ**, определить количество детей старше 8 лет.
7. Вычислить сумму выплаты пособий всех детей по месяцам и оформить в виде таблицы в соответствии с образцом.

Месяц	Сумма пособия

8. Построить диаграмму выплат по месяцам.

### Вариант 3.

Составить таблицу, содержащую следующие поля:

№ п/п	Место работы	Дата поступления	Дата увольнения (перевода)	Должность	Оклад

**Заполнить таблицу в соответствии с условиями:**

- Записей в таблице (мест работы) — 6.
- Заполните столбцы «Дата поступления» и «Дата увольнения». Значения в
- Различных должностей — 3.
- Оклады произвольные.

**Выполнить:**

1. Вставить в таблицу столбец «Стаж» и вычислить стаж работы в каждой должности.
2. Дополнить таблицу столбцами «Надбавка» и «Оплата труда».
3. Значения в столбце «Надбавка» вычислить в соответствии с условием с помощью **ЕСЛИ**
  - Если стаж работы меньше 5 лет, то надбавка 10,
  - от 5 до 10 лет — 15.

– если стаж больше 10 лет - 20.

4. Вычислить оплату труда для каждого места работы.

$$\text{«Оплата труда»} = \text{«Оклад»} * ((\text{«Надбавка»} + 100) / 100)$$

5. Вычислить среднюю оплату труда всех сотрудников с помощью функции: **СРЗНАЧ**

6. Используя функцию **СЧЕТЕСЛИ**, определить количество сотрудников с оплатой меньше средней.

7. Используя стандартные функции, найти минимальное и максимальное значения оплаты труда.

8. Вычислить среднюю оплату труда по должностям и оформить в виде таблицы в соответствии с образцом (**СРЗНАЧЕСЛИ**)

Должность	Средняя оплата

9. Построить диаграмму по таблице

#### Вариант 4.

Составить таблицу, содержащую следующие поля:

ФИО	Дата рождения	Дата поступления	Должность	Оклад

**Заполнить таблицу в соответствии с условиями:**

- Записей в таблице — 10.
- Значения в столбце «Дата поступления» соотносятся со столбцом «Дата рождения». Разница между ними должна быть минимум 14 лет
- Различных должностей — 3.
- Оклады произвольные.

**Выполнить:**

1. Вставить в таблицу столбец «Возраст» и вычислить возраст каждого сотрудника.
2. Вставить в таблицу столбец «Стаж» и вычислить стаж.
3. Дополнить таблицу столбцами «Надбавка» и «Оплата труда».
4. Значения в столбце «Надбавка» вычислить в соответствии с условием с помощью функции **ЕСЛИ**:
  - Если стаж работы меньше 5 лет, то надбавка 15
  - от 5 до 10 лет — 20.
  - если стаж больше 10 лет - 25.
5. Вычислить оплату труда с помощью формулы:

$$\text{«Оплата труда»} = \text{«Оклад»} * (\text{«Надбавка»} + 100/100)$$

6. Вычислить среднюю оплату труда всех сотрудников.
7. Используя функцию **СЧЕТЕСЛИ**, определить количество сотрудников с оплатой меньше средней.
8. Используя стандартные функции, найти минимальное и максимальное значения оплаты труда.
9. Вычислить сумму оплат труда по должностям и оформить в виде таблицы в соответствии с образцом.

Должность	Средняя оплата

10. Построить диаграмму средней оплаты по должностям.

### Практическая работа №13 (ПР13)

**Тема:** Построение экспериментального графика

**Цель:** Научиться строить графики на основе данных, содержащихся на рабочем листе. Научиться настраивать формат диаграммы, задавать отображаемые данные и оформлять получающуюся диаграмму. Научиться изменять формат готовой диаграммы.

#### Выполнение работы:

##### Задание №1

Построить график функции  $y=x^2$ , следуя указанием инструкции:

1. Запустите программу Excel
2. Переименуйте лист как «**Обработка эксперимента**».
3. В столбец А, начиная с ячейки А1 до А15, введите произвольный набор значений (в пределах 100)
4. В столбец В, начиная с ячейки В1, вычислите результат функции  $y=x^2$  от значений ячеек А1:А15
5. Выделите все заполненные ячейки столбцов А и В.
6. На вкладке Вставка в группе Диаграммы в списке выберите пункт «**Вставить точечную (X, Y) или пузырьковую диаграмму/Точечная с гладкими кривыми и маркерами**» (для отображения графика, заданного x,y значениями).
7. Вызовите контекстное меню по диаграмме, выберите «**Выбрать данные**». В открывшемся окне «Выбор источника данных» выберите «**Ряд1**» и нажмите кнопку «**Изменить**», в поле «Имя Ряда» укажите: «Результаты измерений». Щелкните на кнопке **ОК**.
8. На вкладке **Конструктор** выберите пункт **Переместить диаграмму**. Установите переключатель «**На отдельном листе**». По желанию, задайте произвольное имя добавляемого рабочего листа. Щелкните на кнопке **ОК**.




9. Убедитесь, что диаграмма построена и внедрена в новый рабочий лист. Рассмотрите ее и щелкните на построенной кривой, чтобы выделить ряд данных.
10. Убедитесь, что вы находитесь во вкладке «Линия». В пункте «Цвет» выберите красный цвет. В пункте «Тип штриха» выберите штрих-пунктир.
11. Перейдите во вкладку «Маркер». Раскройте «Параметры маркера». Выберите в списке «Тип» - «Треугольный маркер». В пункте «Заливка/Цвет» выберите зеленый цвет.
12. Сохраните рабочую книгу book.xls

### Задание 2.

Построить самостоятельно два графика следующих функций варианту. Задать значения  $x$  последовательно.

Справка:

	Синтаксис в Excel
$\frac{x+2}{x+3}$	(x+2)/(x+3)
$\sqrt{x}$	x^(1/2)
$x^n$	x^n
$\sqrt[3]{x}$	x^(1/3)

№	$f(x)$	№	$f(x)$	№	$f(x)$
1	$\sqrt[3]{(1+x)(x^2+2x-2)}$	11	$\sqrt[3]{(1-x)(x^2-2x-2)}$	21	$\sqrt[3]{x(x^2+2)^2}$
2	$\sqrt[3]{(x^2-4x+3)^2}$	12	$\sqrt[3]{x^2(x^2+2)^2}$	22	$\sqrt[3]{(3+x)(2x^2+x-1)}$
3	$\sqrt[3]{(3+x)x^2}$	13	$\sqrt[3]{(x+2)^2(x-1)}$	23	$\sqrt[3]{(x-1)^2} - \sqrt[3]{(x-2)^2}$
4	$\sqrt[3]{(2+x)^2(x^2-4)}$	14	$\sqrt[3]{(x-2)^2(x+1)}$	24	$\sqrt[3]{(4+x)^2} - \sqrt[3]{(x+3)^2}$
5	$\sqrt[3]{(1+x)^2} - \sqrt[3]{(x+2)^2}$	15	$\sqrt[3]{(2+x)^2} - \sqrt[3]{(x+3)^2}$	25	 $\sqrt[3]{(6+x)x^2}$

№	$f(x)$	№	$f(x)$	№	$f(x)$
1	$\frac{4x^2+5}{4x+8}$	11	$\frac{2-x^2}{\sqrt{9x^2-4}}$	21	$\frac{4x^3+3x^2-2x-2}{x^2-1}$
2	$\frac{17-x^2}{4x-5}$	12	$\frac{x^3+3x^2-2x-2}{2-3x^2}$	22	$\frac{1-x^2}{\sqrt{16x^2-9}}$
3	$\frac{x^2-3}{\sqrt{4x^2-3}}$	13	$\frac{3x^2-7}{2x+1}$	23	$\frac{2x^2-3x+1}{1-2x}$
4	$\frac{x^3-4x}{3x^2-4}$	14	$\frac{x^2-5}{\sqrt{9x^2-8}}$	24	$\frac{4x^3+x^2-2x-1}{2x^2-1}$
5	$\frac{4x^3+3x^2-8x-2}{2-3x^2}$	15	$\frac{x^2-6x+4}{2-2x}$	25	$\frac{5x^2-3}{\sqrt{3x^2-1}}$

## Практическая работа №14 (ПР14)

**Тема:** Построение диаграмм

**Цель:** Научиться представлять цифровые данные в графическом виде

### Выполнение работы:

#### Задание 1.

Заполнить ячейки недостающими формулами:

- доля городского населения от населения государства в целом (%):  
= **Городское население / Общее население**
- плотность населения данного государства (**количество человек на 1 км<sup>2</sup>**):  
= **Население / Территория**
- суммарные данные по каждому столбцу (строка «Всего»).

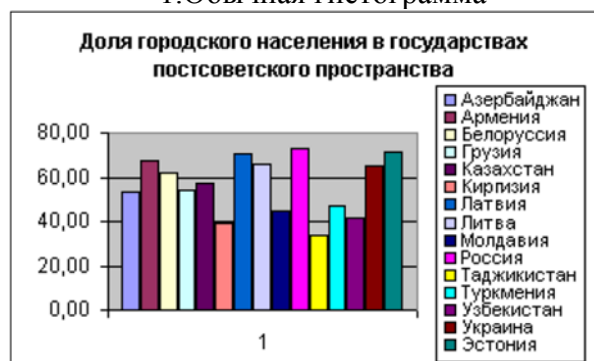
**Примечание.** Все вычисления производить с точностью до десятых.

№	Государство	Территория, тыс. км <sup>2</sup>	Население, тыс. чел.	В том числе городское	Доля городского населения, %	Плотность населения, чел. на 1 тыс км <sup>2</sup>
1	Азербайджан	86,6	6614	3553		
2	Армения	29,8	3317	2242		
3	Белоруссия	207,6	9942	6167		
4	Грузия	69,7	5201	2798		
5	Казахстан	2717,3	15842	9045		
6	Киргизия	198,5	3967	1569		
7	Латвия	63,7	2604	1836		
8	Литва	65,2	3570	2345		
9	Молдавия	33,7	4111	1838		
10	Россия	17075,4	143090	103860		
11	Таджикистан	143,1	4499	1512		
12	Туркмения	488,1	3189	1513		
13	Узбекистан	447,4	17974	7530		
14	Украина	603,7	50840	33244		
15	Эстония	45,1	1530	1093		
	Всего					

#### Задание 2.

Построить все диаграммы по значениям в таблице:

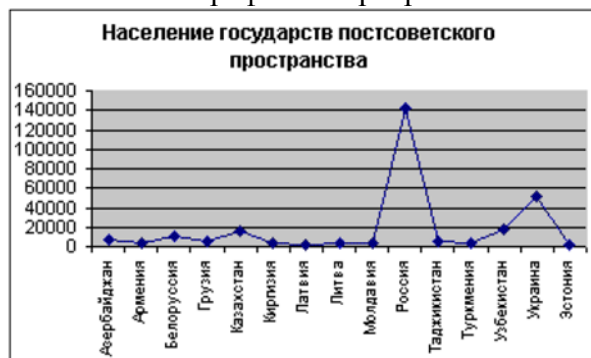
1. Обычная гистограмма



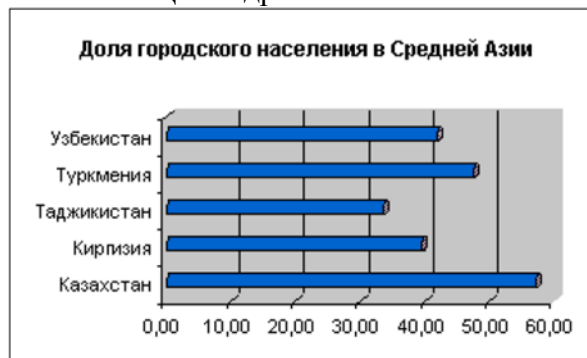
2. Круговая диаграмма



### 3.График с маркерами



### 4.Цилиндрическая линейная



### 5.Кольцевая



### 6.Круговая объемная

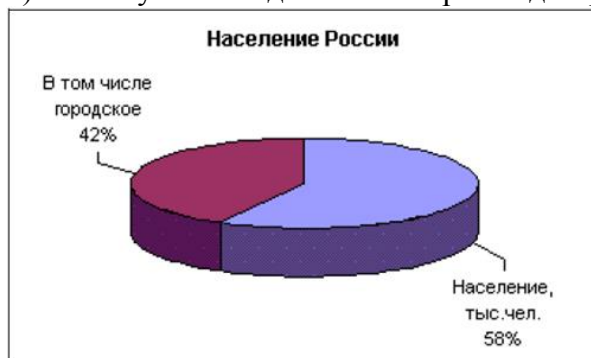
1) Вычислить «Население тыс. чел.» по формуле в процентах:

$$\text{Население тыс. чел} = \frac{\text{Городское население России}}{\text{Городское население Всего}}$$

2) Вычислить «В том числе городское» по формуле в процентах:

$$\text{В том числе городское} = \text{Население тыс. чел.} * \text{Долю Городского Населения России}$$

3) По полученным данным построить диаграмму



## Практическая работа №15 (ПР15)

**Тема:** Создание и форматирование прайс-листа

**Цель:** Научиться форматировать документ Excel (изменение ширины столбцов, объединение ячеек, управление выравниванием текста, создание рамок ячеек).

## Выполнение работы:

### Задание 1.

Создать прайс-лист

1. Запустите программу EXCEL.
2. Дважды щелкните на ярлычке листа и переименуйте его как «Прейскурант».
3. В ячейку A1 введите текст «Прейскурант» и нажмите клавишу ENTER.
4. В ячейку A2 введите текст «текущий курс пересчета»: и нажмите клавишу ENTER.
5. В ячейку B2 введите текст «1 у.е.=» и нажмите клавишу ENTER.
6. В ячейку C2 введите «65,36» и нажмите клавишу ENTER.
7. В ячейку A3 введите текст «Наименование товара» и нажмите клавишу ENTER.
8. В ячейку B3 введите текст «Цена (у.е.)» и нажмите клавишу ENTER.
9. В ячейку C3 введите текст «Цена (руб.)» и нажмите клавишу ENTER.
10. В последующие ячейки столбца A введите названия товаров, включенных в прейскурант.
11. В соответствующие ячейки столбца A введите наименование товаров.
12. В соответствующие ячейки столбца B введите цены товаров в условных единиц.

Перечень товаров для составления прайс-листа.			
№п-п	Наименование товара	Марка товара	Стоимость в у. е.
1	Материнская плата	Gigabyte	50
2	Материнская плата	Asus P4G8X	209
3	Процессор	Celeron 1300	45
4	Процессор	P4XEON	260
5	Оперативная память	SDRAM 128mb	20
6	Оперативная память	RIMM 512	265
7	Жесткий диск	40 Gb	60
8	Жесткий диск	120 Gb	124
9	Видеокарта	RIVA-TNT 32 mb	20
10	Видеокарта	ATI 128 mb	339
11	Звуковая карта	SB Player 5.1	30
12	Звуковая карта	SB Platinum	190
13	Дисковод	FDD 1.44mb	8
14	Дисковод	ZIP 1.44	69
15	CD-ROM	52xAcer	20
16	CD-ROM	52xAsus	25
17	Корпус	Midi Tower ATX	19
18	Корпус	Midi Tower ATX 3RAIR	94
19	Монитор (15")	Samsung	115
20	Монитор (17")	Sony	337

13. В ячейку C4 введите формулу: =B4\*\$C\$2, которая используется для пересчета цены из условных единиц в рубли.
14. Методом автозаполнения скопируйте формулы во все ячейки столбца C, которым соответствуют заполненные ячейки столбцов A и B.
15. Выделите методом протягивания диапазон A1 :C 1 и дайте команду **Вкладка Главная > группа Число > Формат ячеек**. На вкладке «Выравнивание» задайте выравнивание по горизонтали «По центру» и установите флажок «Объединение ячеек».
16. На вкладке «Шрифт» задайте размер шрифта в 14 пунктов и в списке «Начертание» выберите вариант «Полужирный». Щелкните на кнопке ОК.
17. Щелкните правой кнопкой мыши на ячейке B2 и выберите в контекстном меню команду «Формат ячеек». Задайте выравнивание по горизонтали «По правому краю» и щелкните на кнопке ОК.

18. Щелкните правой кнопкой мыши на ячейке C2 и выберите в контекстном меню команду «Формат ячеек». Задайте выравнивание по горизонтали «По левому краю» и щелкните на кнопке ОК.
19. Установите для диапазона A1:C23 Границы/Все границы
20. Установите для диапазона A3:C3 Границы/Толстые внешние границы
21. Дважды щелкните на границе между заголовками столбцов А и В, В и С, С и D. Обратите внимание, как при этом изменяется ширина столбцов А, В и С.
22. Посмотрите, устраивает ли вас полученный формат таблицы. Щелкните **Файл/Печать**, чтобы увидеть, как документ будет выглядеть при печати.
23. Измените область печати (предпросмотра). Для этого сперва выделите все ячейки, которые вы хотите вывести на предпросмотр (печать). Нажмите Разметки страницы/Область печати/Задать.
24. Проверьте еще раз **предварительный просмотр**
25. Сохраните рабочую книгу

## Практическая работа №16 (ПР16)

**Тема:** Форматирование текстового документа с использованием вставки объекта в виде математических формул.

**Цель:** Научиться вставлять, создавать, редактировать текстовую информацию. С использованием математических формул.

### Выполнение работы:

#### Задание 1.

Создание формул по образцу (*Вставка/Символы/Уравнение/Вставить новое уравнение*):

1. В первом уравнении используйте *Дробь* и *Индекс*.

$$\frac{1}{x-a} \text{ И } \frac{Mx+N}{x^2+px+q} \left( \frac{p^2}{4} - q < 0 \right)$$

2. В остальных уравнениях также используйте *Дробь* и *Индекс*.

$$\underline{A} \arctg(a_1x + a_2) + \underline{B} \ln(b_1x + b_2) + C.$$

$$\frac{A}{(x-a)}, a=2,3$$

$$\frac{Mx+N}{(x^2+px+q)^\beta}, \beta = 2,3,$$

$$\underline{A} \arctg(a_1x + a_2) + C$$

$$\frac{A}{x-a} \text{ И } \frac{Mx+N}{x^2+px+q}; \frac{p^2}{4} - q < 0$$

$$Q(x) = (x - a_1)^{\alpha_1} \dots (x - a_r)^{\alpha_r} (x^2 + p_1x + q_1)^{\beta_1} \dots (x^2 + p_s + q_s)^{\beta_s}.$$

3. Далее, чтобы ввести следующее уравнение используйте: *Интеграл*, *Скобка*, *Дробь* и *Индекс*.

$$\int \frac{P(x)}{q(x)} dx = \frac{P_1(x)}{Q_1(x)} + \int \left[ \sum_{i=1}^r \frac{A_i}{x-a_i} + \sum_{j=1}^s \frac{M_j x + N_j}{x^2 + p_j x + q_j} \right] dx,$$

4. Далее, чтобы ввести оставшийся текст с уравнениями, используйте *Интеграл*, *Дробь*, *Индекс*.

Откуда произведения под знаком интеграла сложенные дробей, имеет  $\int \frac{P(x)}{Q(x)} dx = \frac{P_1(x)}{Q_1(x)} + \int \frac{P_2(x)}{Q_2(x)} dx$ , где  $Q_2(x) = (x - a_1) \dots (x - a_r)(x^2 + p_1 x + q_1) \dots (x^2 + p_s x + q_s)$ ;

$Q_1(x) = (x - a_1)^{\alpha_1 - 1} \dots (x - a_r)^{\alpha_r - 1} (x^2 + p_1 x + q_1)^{\beta_1 - 1} \dots (x^2 + p_s x + q_s)^{\beta_s - 1}$ ,  
следовательно: многочлен  $Q_1(x)$  – наибольший общий делитель многочлена  $Q(x)$  и его производитель.

5. Сохраните файл Место для уравнения.

### Задание 2.

Создание формул по образцу:

1. Использовать в уравнении *Предел* и *логарифм*.

$$\text{A) } \lim_{x \rightarrow 0} \frac{\ln \cos x}{\ln \cos 3x}, \text{ b) } \lim_{x \rightarrow 0} \frac{x^2 - 2x + 1}{x^2 - 1}$$

2. Далее введите формулу с использованием *Радикала* и *Логарифма*

$$\text{A) } \lim_{x \rightarrow 0} \frac{n^2 + 3n - 2}{1 + 2 + \dots + n}, \text{ b) } \lim_{x \rightarrow 0} \frac{1 - \cos x}{x(\sqrt{1+x} - 1)}$$

3. Введите *интегралы*.

$$\text{A) } \int \arctg x dx, \text{ b) } \int \frac{x dx}{\sqrt{3-x^4}}$$

4. Введите *Матрицу*.

$$\begin{vmatrix} 1 & 2 & 3 & 4 \\ -1 & 3 & -1 & 7 \\ 4 & -2 & 2 & 6 \\ 5 & 5 & 1 & 3 \end{vmatrix}$$

5. Сохраните файл

### Задание 3.

Создание формул по образцу:

1. В первом примере используйте знаки  $\langle, \rangle$  и *индексы*.

$$a > 0, b > 0, x_1 \text{ и } x_2$$

2. Далее введите оставшийся пример, используя *Индекс, Радикал, Дробь* и т.д.

$$a^{x_1} a^{x_2} = a^{x_1+x_2}$$

$$\frac{a^{x_1}}{a^{x_2}} = a^{x_1-x_2}$$

$$(a^{x_1})^{x_2} = a^{x_1 \cdot x_2}$$

$$(ab)^x = a^x b^x$$

$$\left(\frac{a}{b}\right)^x = \frac{a^x}{b^x}$$

$$a^x > 0, a^x > 1, \text{ если } a > 1, x > 0$$

3. Сохраните файл

## Практическая работа №17 (ПР17)

**Тема:** PowerPoint. Создание слайдов, вставка фигур, текста и надписей.

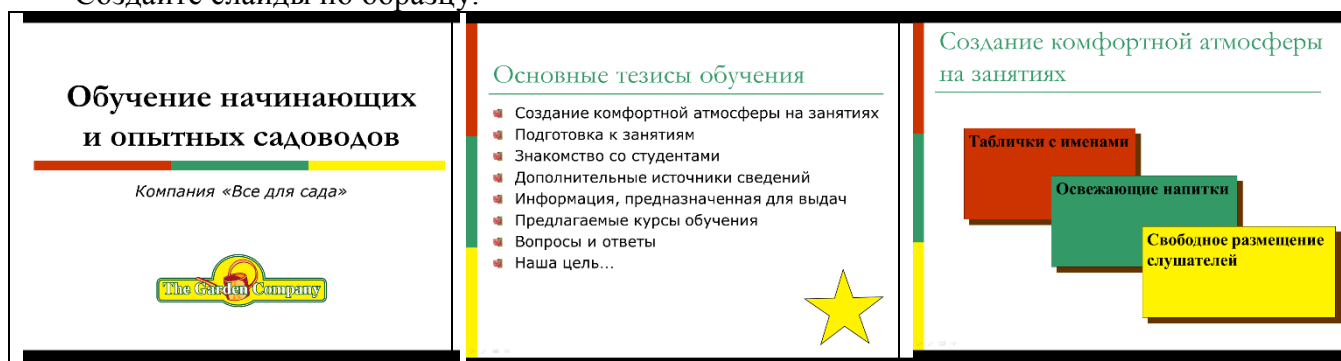
**Цель:** Ознакомиться с инструментами программы, научиться создавать простые слайды с использованием текста и рисунков.

### Выполнение работы:

#### Задание 1.

Создание слайдов по образцу.

- Для создания презентации без мастера автосодержания установите переключатель Пустую презентацию при запуске PowerPoint или в диалоговом окне Создать презентацию выберите значок Новая презентация. Откроется диалоговое окно Создание слайда.
- В диалоговом окне макет для нового слайда. Выберите для первого слайда презентации разметку Титульный слайд.
- Введите заголовок вашей презентации. Для ввода текста используйте текстовые поля на слайде. На фигурах текст вставлять с помощью инструментов: Вставка – Текст – Надпись.
- Закончив работу с первым слайдом, выберите команду Создать слайд на главной панели. При этом вновь открывается диалоговое окно Создать слайд и можно выбрать макет разметки.
- Создайте слайды по образцу:



### Подготовка к занятиям

### Знакомство со студентами

### Дополнительные источники сведений

Сотрудники компании «Все для сада» могут создать для вас презентацию PowerPoint до начала занятий.

### Предлагаемые курсы обучения

### Вопросы и ответы

- Запись на курсы, продолжительность и стоимость обучения
- Реклама занятий
- Скидки для преподавателей и студентов

**Компания «Все для сада» ищет опытных преподавателей для садоводческих курсов.**

### Наша цель...

- Рисование фигур с помощью пунктов меню: Вставка – Фигуры. Редактировать внешний вид фигур можно с помощью контекстного меню: Формат рисунка.

## Задание 2.

Вставка объектов на слайды.

- Добавление картинки в слайд. В режиме слайдов выберите команду Вставка - Рисунки - выберите вкладку Графика или дважды щелкните на месте для картинки на слайде с картинкой. Выберите картинку для вставки и щелкните на кнопке Вставить.
- Вставка объекта Microsoft Chart В PowerPoint выберите Вставка > Диаграмма. Введите собственные данные. Выделите диаграмму и задайте ее размеры.
- Вставка таблицы из Word Выберите эскиз слайда с таблицей (Вставка > Таблица). Укажите нужное число строк и столбцов и щелкните на кнопке ОК. Заполните таблицу, используя возможности Word для форматирования и улучшения оформления. Щелкните вне таблицы для закрытия таблицы.
- Вставьте объекты по образцу:

### Знакомство со студентами

- Определите уровень знаний студентов
- Учитывайте при обучении уровень подготовки студентов

### Пример бюджета проекта

\$100 Бюджет цветника на сезон Осень/Зима

- Цель-Семена/Рассада
- Садовые инструменты
- Почва/Удобрения
- Канцелярские принадлежности
- Полыноса/Средства ухода

### Штат компании «Все для сада»

### Цель обучения

Познакомиться с клиентами

Увеличить уровень продаж      Способствовать просвещению

### Расписание занятий и аудиторий

Курс обучения	Место проведения занятий
Деревья и кустарник Уход за газоном	Внутренний дворик
Цветы	Комната 1
Комнатные растения	Оранжерея

### Вопросы и ответы

- Запись на курсы, стоимость и продолжительность обучения.
- Реклама курсов.
- Скидки для преподавателей или студентов.



- Для вставки диаграммы нужно ее предварительно подготовить в программе Excel:

Ориентировочный бюджет цветника		
Поставки	Весна/Лето	Осень/Зима
Цветы--Семена/Рассада	\$250,00	\$300,00
Садовые инструменты	\$50,00	\$50,00
Почва/Удобрения	\$100,00	\$75,00
Камни/Элементы декора	\$75,00	\$50,00
Поливка/Средства ухода	\$25,00	\$25,00
Итого:	\$500,00	\$500,00

### Задание 3.

Создание эффектов анимации и эффектов переходов.

Эффекты анимации:

1. Выделите объект щелчком левой кнопки мыши (Заголовок слайда).
2. Выберите команду **Анимация>Добавить анимацию** и выберите эффект из группы **Вход**.
3. Установите необходимые параметры показа. В группе **Время показа слайда** в поле **Начало** установите **После предыдущего**
4. Переходите к другому объекту и повторите указанные действия.
5. Нажмите кнопку Просмотр для просмотра результата анимационного эффекта.
6. Создайте анимацию на каждом слайде презентации.

Эффекты перехода:

1. Перейти на вкладку **Переходы**.
2. На панели **Переход к этому слайду** раскройте выпадающее меню визуальных эффектов перехода.
3. Выберите одну из предложенных схем.
4. Для изменения параметров перехода нажмите кнопку **Параметры эффектов** и выберите вариант (справа, слева, снизу или сверху).
5. Для непрерывного показа слайдов нужно в пункте меню **Смена слайдов** установить время показа одного слайда и нажать кнопку **Применить ко всем**.
6. Установите эффекты перехода ко всей презентации.
7. Просмотрите результат показа презентации и при необходимости внесите изменения.

## Практическая работа №18 (ПР18)

**Тема:** Создание базы данных «Колледж»

**Цель:** Научиться проектировать и создавать реляционную базу

### Выполнение работы:

#### Задание №1

Создать проект и таблицы базы данных «Колледж» в соответствии с образцами:

### Конструктор таблицы «Студенты».

Имя поля	Тип данных	Размер поля
Код студента	Числовой	Целое
Фамилия	Короткий текст	15
Имя	Короткий текст	12
Отчество	Короткий текст	15
Номер группы	Числовой	Целое
Телефон	Короткий текст	9
Стипендия	Логический	Да/нет

### Конструктор таблицы «Дисциплины»

Имя поля	Тип данных	Размер поля
Код дисциплины	Числовой	Целое
Название дисциплины	Короткий текст	30

### Конструктор таблицы «Оценки»

Имя поля	Тип данных	Размер поля
Код студента	Числовой	Целое
Код дисциплины	Числовой	Целое
Оценки	числовой	байт

### Конструктор таблицы «Преподаватели»

Имя поля	Тип данных	Размер поля
Код преподавателя	Счетчик	
Код дисциплины	Числовой	
Фамилия	Короткий текст	15
Имя	Короткий текст	15
Отчество	Короткий текст	15
Дата рождения	Дата и время	Краткий
Телефон	Короткий текст	9
Зарплата	Денежный	

#### Создание таблиц:

- Нажмите **Создание/Таблица**.
- В левой панели **Все объекты Access** выберите первоначальную таблицу с именем **Таблица1**, нажмите **правую кнопку мыши** и выберите пункт **Конструктор**
- в появившемся окне **Сохранение** введите **Имя первой таблицы** «Студенты» и щелкните по кнопке **<ОК>**. В результате проделанных операций открывается окно таблицы в режиме конструктора, в котором следует определить поля таблицы.
- В окне конструктора таблицы создайте структуру таблицы «Студенты»:
- В столбце «Имя поля» задавайте имена полей согласно «Конструктора таблицы Студенты»
- В столбце «Тип поля» задавайте типы, определенные в проекте.
- Размер каждого поля указывается в свойствах поля на вкладке «Общие» (щелкнуть мышкой по данной ячейке и выбрать вариант из предложенного списка).

- Задайте ключевое поле на имени «Код студента»
- Закройте конструктор таблиц (крестик на вкладке таблицы) и сохраните таблицу (нельзя закрывать всю базу данных по верхнему крестику).
- Для создания новой таблицы откройте на ленте пункт «Создание», «Конструктор таблиц».
- Заполните столбцы конструктора и при сохранении задайте имя следующей таблицы.
- В таблице «Оценки» ключевое поле задавать не нужно, т.к. данные во всех полях могут повторяться
- Нужно создать все 4 таблицы.

## Задание 2.

### Создание форм

Формы необходимы для заполнения и редактирования записей в таблицах. Для каждой таблицы создается своя форма со всеми полями таблицы. Для создания формы выполните следующие действия:

- Открыть ленту «Создание» и выбрать пункт «Мастер форм»;
- Из списка «Таблицы и запросы» выбрать таблицу, для которой создается форма;
- С помощью кнопки со стрелками выбрать все поля таблицы и нажать «Далее»;
- Оставить внешний вид без изменения и нажать «Далее»;
- Для завершения создания формы нажать «Готово»
- Таким образом создать формы для всех таблиц.
- Для изменения внешнего вида формы можно перейти в режим Макет, вкладка «Конструктор»

## Задание 3.

### Заполнение таблиц

Открыть форму для конкретной таблицы (двойной щелчок по названию формы);

- Заполнить первую запись и перейти на следующую по счетчику вниз формы.
- Коды дисциплин должны быть одинаковыми во всех таблицах.

Код студента	фамилия	имя	отчество	Номер группы	телефон	стипендия
1	Арбузов	Николай	Николаевич	151	260-15-63	Да
2	Киршин	Петр	Валерьевич	151	110-67-82	Да
3	Кривин	Сергей	Николаевич	151	172-97-21	Нет
4	Крылова	Елена	Петровна	151	130-31-87	Да
5	Кулаков	Григорий	Викторович	151	269-53-75	Да
6	Патрикеев	Олег	Борисович	152	234-11-63	Нет
7	Перлов	Кирилл	Николаевич	152	312-21-33	Нет
8	Соколова	Наталия	Петровна	152	166-87-24	Нет
9	Степанова	Ольга	Витальевна	152	293-43-77	Да
10	Тимофеев	Сергей	Трофимович	152	260-11-57	Да

Таблица 1. Данные о студентах

Код дисциплины	Название дисциплины
1	Информатика
2	Математика
3	Физика
4	Экономика

Таблица 2. Данные о дисциплинах

Код студента	Код дисциплины	Оценки
1	1	4
1	2	5
1	3	4
1	4	4
2	1	5
2	2	5
2	3	4
2	4	4
3	1	3
3	2	5
3	3	4
3	4	3
4	1	4
4	2	4
4	3	5
4	4	4
5	1	5
5	2	5
5	3	5
5	4	5
6	1	5
6	2	4
6	3	5
6	4	4
7	1	4
7	2	3
7	3	4
7	4	3
8	1	4
8	2	3
8	3	3
8	4	5
9	1	4
9	2	4
9	3	4
9	4	4
10	1	5
10	2	5
10	3	5
10	4	5

Таблица 3. Данные о оценках

Код преподавателя	Код дисциплины	Фамилия	Имя	Отчество	Дата рожд.	Телефон	Зарплата
1	1	Истомин	Ремир	Евгеньевич	23.10.54	110-44-68	8900р.
2	4	Миронов	Павел	Юрьевич	25.07.40	312-21-40	12000р.
3	2	Гришин	Евгений	Сергеевич	05.12.67	260-23-65	7600р.
4	2	Сергеева	Ольга	Ивановна	12.02.72	234-85-69	4500р.
5	4	Емец	Татьяна	Ивановна	16.02.51	166-75-33	8900р.
6	1	Игнатъева	Татьяна	Павловна	30.05.66	210-36-	7900р.

						98	
7	3	Миронов	Алексей	Николаевич	30.07.48	166-75-33	8900р.

Таблица 4. Данные о преподавателях

#### Задание 4.

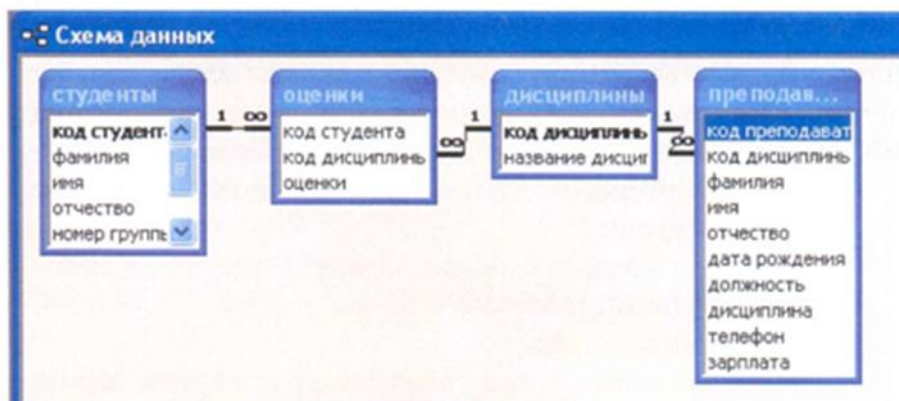
Создание связей между таблицами

В БД «Колледж» будут созданы следующие связи между таблицами:

- таблица «Студенты» —> таблица «Оценки» (поле Код студента);
- таблица «Дисциплины» -> таблица «Оценки» (поле Код дисциплины);
- таблица «Дисциплины» —> таблица «Преподаватели» (поле Код дисциплины).

Для создания связей нужно выполнить следующие операции:

- Открыть вкладку «Работа с базами данных», пункт «Схема данных»
- Выбрать кнопку «Отобразить таблицы»;
- Поочередно выбрать из списка все таблицы базы данных (кнопка Добавить)
- Закрыть диалоговое окно.
- Установить связи согласно схемы связи между таблицами (навести мышку на нужный пункт таблицы, нажать левую кнопку и, не отпуская перетащить на такой же пункт другой таблицы).
- В появившемся диалоговом окне устанавливать галочку в пункте «Обеспечение целостности связей».
- После окончания должна получиться следующая схема



#### Задание 5.

Создание запросов:

##### Порядок выполнения запросов:

##### Запрос №1 (с параметрами):

Отобрать студентов конкретной группы

- Выбрать закладку «Создание», кнопку «Конструктор запросов»;
- Добавить таблицу «Студенты» и закрыть диалоговое окно;
- Перенесите все поля таблицы в макет запроса (двойной щелчок на названии поля в таблице);
- В поле «Номер группы» в строке «Условия отбора» запишите текст [Номер группы:] (скобки и двоеточие английский алфавит);
- Сохраните документ и тем самым запрос, дайте имя запросу «Группа»
- Включить выполнение запроса. (Двойной щелчок по запросу в левой панели «Все объекты Access».
- В диалоговом окне указать номер группы: 151
- Появится список указанной группы.

### **Запрос №2.**

Вывести оценки студентов заданной группы по заданной дисциплине

- Выбрать закладку «Создание», кнопку «Конструктор запросов»;
- Добавить таблицы «Студенты», «Дисциплины» и «Оценки»
- Выберите таблицу «Студенты» и перенесите поля «Фамилия», «Имя», «Отчество», «Номер группы» в окно в макет запроса;
- В таблице «Дисциплины» выберите поле «Название дисциплины»;
- В таблице «Оценки» выберите поле «Оценки».
- В строке Условия отбора для поля «Номер группы» введите фразу [Введите номер группы:]
- В строке Условия отбора для поля «Название дисциплины» введите фразу: [Введите название дисциплины:];
- Сохраните запрос под именем «Оценки по дисциплинам»
- Выполните запрос;
- Введите в диалоговые окна: Информатика и 151. Проверьте результат

### **Запрос №3.**

**Вывести средние оценки по дисциплинам у групп**

- Выбрать закладку «Создание», кнопку «Конструктор запросов»;
- Добавить таблицы «Студенты», «Дисциплины» и «Оценки»
- Отобразить раздел «Групповая операция» для это во вкладке «Конструктор» нажмите «Показать или скрыть/Итоги»
- Из таблицы «Дисциплины» выберите поле «Название дисциплины»
- Из таблицы «Оценки» выберите поле «Оценки»
- Из таблицы «Студенты» выберите поле «Номер группы»
- В строке Групповая операция для поля «Оценки» выберите «Avg» - среднее значение.
- Сохраните запрос под именем «Средний»
- Выполните запрос.

### **Запрос №4.**

Разработайте запрос, отображающий только студентов - отличников.

**Для всех созданных вами запросов разработайте формы (Создание/Мастер форм/Выбрать запрос)**

### **Задание 6.**

Создание отчетов

#### 1. Создание отчета по запросу №1

- На вкладке «Создание» на панели «Отчеты» нажать кнопку «Мастер отчетов»;
- В окне Таблицы и запросы выбрать запрос «Группа»;
- Перенести нужные поля (можно все) и нажать Далее;
- Выбрать уровни группировки (Фамилия) и нажмите Далее и Готово
- При желании можно изменить внешний вид отчета, перейдя в режим Макета (двойной щелчок по имени запроса, на закладке Главная, Режим, Режим макета).

#### 2. Аналогично создайте отчет по запросу №2

## Практическая работа №19(ПР19)

**Тема:** Форматирование текстовых документов с использованием функций вставки графических объектов.

**Цель:** Научиться выполнять вставку рисованных объектов, добавление объектам объема, тени, заливки, изменения типа и толщины линий.

**Выполнение работы:**

**Задание 1.**

1. Открыть программу MS Word.
2. Создать новый документ.
3. Добавить фигуру прямоугольника.

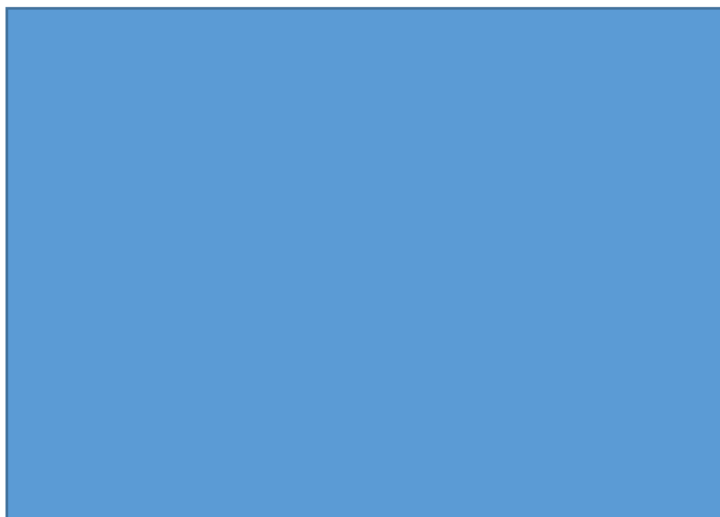


Рисунок 1 – Прямоугольник

4. В контекстном меню прямоугольника нажать на команду *добавить текст*.
5. Ввести текст в фигуре с форматированием: Шрифт – TimesNewRoman, размер – 36, растянуть по ширине.



Рисунок 2 – Изменённый прямоугольник

6. Далее ввести оставшийся текст.
7. Добавить фигуры выполняя форматирование: заливка, контур фигуры, цвет контура.



Рисунок 3 – Форматированные фигуры

8. Ввести весь оставшийся текст.
9. Сохранить файл под именем **Реклама**.

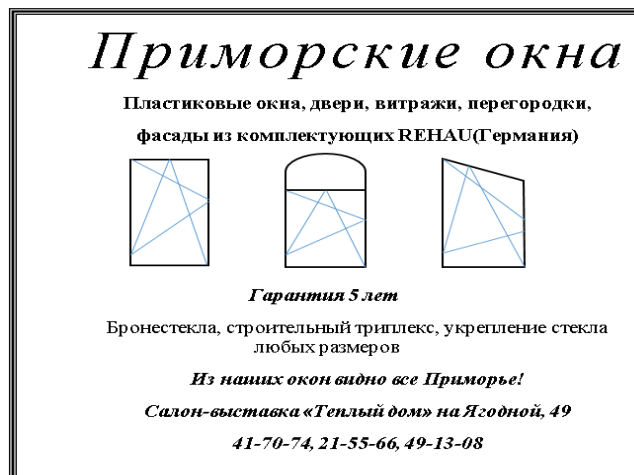


Рисунок 4 – Реклама

### Задание 2.

1. Создать новый документ.
2. Вставить фигуру прямоугольник.
3. Сделать заливку серым цветом.
4. Добавить текст.
5. Сделать форматирование: шрифт – **TimesNewRoman**, размер – 28, интервал – разреженный 2-пт. (Главная/Шрифт/Дополнительно/Интервал-Разряженный)



Рисунок 5 – Форматирование по зад. 2

6. В прямоугольник вставить Символ с помощью таблицы символов. (любой понравившийся)
7. Ниже написать текст.

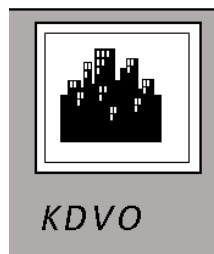


Рисунок 6 – Добавление символа

8. Ввести оставшийся текст.
9. Сохранить файл

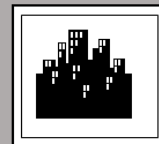


# ЛАМИНИРОВАННЫЕ ПАНЕЛИ

молдинги в ассортименте

для потолка и стен

ул. Курортная, 3, 5 этаж, каб. 76, тел. (4522) 22-57-90



KDVO

Рисунок 7 – Итоговый вид

### Задание 3.

1. Создать новый документ.
2. Добавить прямоугольник.
3. Сделать форматирование: заливка – зелёный, контур фигуры, цвет контура – черный.



Рисунок 8 – Форматирование прямоугольника

4. Далее ввести текст АВТОШКОЛА с помощью WordArt, сделать форматирование: шрифт – **TimesNewRoman**, шрифт – 28.



Рисунок 9 – Форматирование

5. Ввести оставшийся текст.
6. Добавить Символ машины, вставить фигуры.
7. Добавить тени фигуре солнца.
8. Сохранить файл под именем Автошкола



Рисунок 10 – Информация о выполнении упр.3

#### Задание 4.

1. Создать новый документ.
2. Добавить фигуру прямоугольника.
3. Сделать заливку в желтый цвет.
4. Добавить Заголовок.

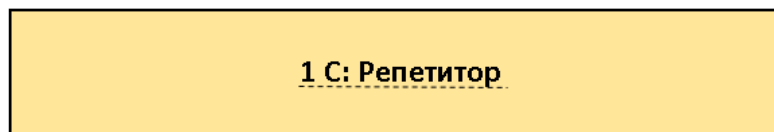


Рисунок 11 – Форматирование фигуры

5. Далее добавить маркеры к некоторым частям текста.
6. Сделать редактирование текста – «Для старшеклассников», с помощью WordArt.
7. Добавить эффекты к тексту.
8. Вставить символ книг.
9. Сохранить файл под именем Репетитор

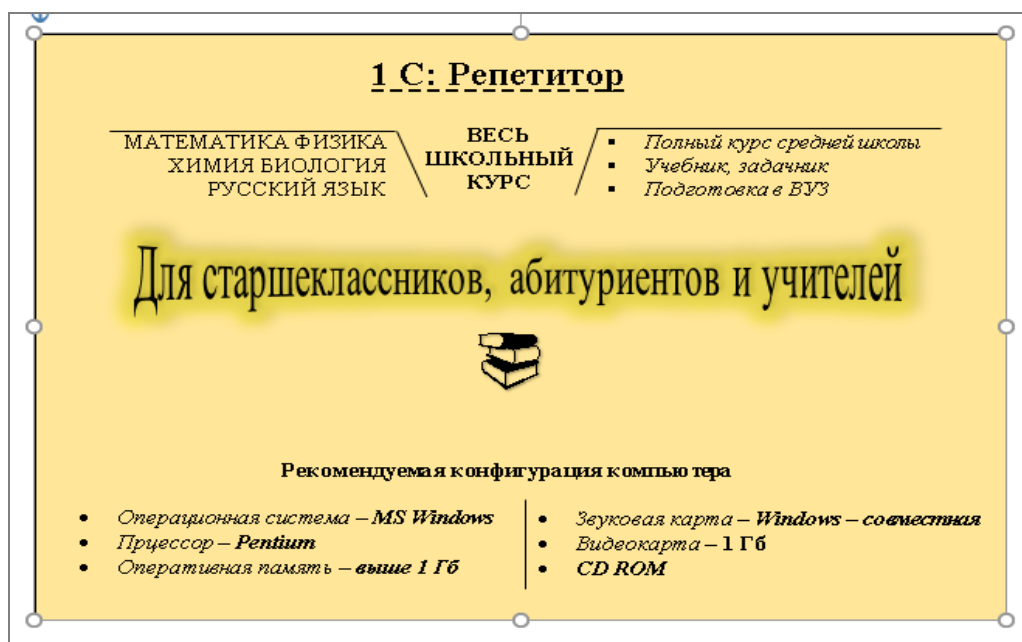


Рисунок 12 – Итоговый вид

### Практическая работа №20 (ПР20)

**Тема:** Поиск информации на государственных образовательных порталах

**Цель:** изучение информационной технологии организации поиска информации на государственных образовательных порталах.

#### Выполнение работы:

##### Задание №1.

1. Загрузите Интернет.
2. С помощью строки поиска найдите каталог ссылок на государственные образовательные порталы.
3. Выпишите электронные адреса шести государственных образовательных порталов и дайте им краткую характеристику. Оформите в виде таблицы:

№	Название портала	Электронный адрес портала	Характеристика портала


### Задание №2.

1. Загрузите страницу электронного словаря Promt– [www.ver-dict.ru](http://www.ver-dict.ru).
2. Из раскрывающегося списка выберите *Русско-английский словарь (Русско-Немецкий)*.
3. В текстовое поле *Слово для перевода:* введите слово, которое Вам нужно перевести.
4. Нажмите на кнопку *Найти*.
5. Занесите результат в следующую таблицу:

Слово	Русско-Английский	Русско-Немецкий
Информатика		
Клавиатура		
Программист		
Монитор		
Команда		
Винчестер		
Сеть		
Ссылка		
Оператор		

### Задание №3.

1. Загрузите страницу электронного словаря– [www.efremova.info](http://www.efremova.info).
2. В текстовое поле *Поиск по словарю:* введите слово, лексическое значение которого Вам нужно узнать.
3. Нажмите на кнопку *Искать*. Дождитесь результата поиска.
4. Занесите результат в следующую таблицу:

Слово	Лексическое значение
Метонимия	
Видеокарта	
Железо	
Папирус	
Скальпель	
Дебет	

**Задание №4.** С помощью одной из поисковых систем найдите информацию и занесите ее в таблицу:

Личности 20 века		
Фамилия, имя	Годы жизни	Род занятий
Джеф Раскин		
Лев Ландау		
Юрий Гагарин		

**Задание №5.** Заполните таблицу, используя поисковую систему Яндекс: [www.yandex.ru](http://www.yandex.ru).

Слова, входящие в запрос	Структура запроса	Количество найденных результатов	Электронный адрес первой найденной ссылки
Информационная система	Информационная! Система!		
	Информационная + система		
	Информационная - система		
	«Информационная система»		
Персональный компьютер	Персональный компьютер		
	Персональный & компьютер		
	\$title (Персональный компьютер)		
	\$anchor (Персональный компьютер)		

**Задание №6.**

Ответить на вопросы:

1.	Что понимают под поисковой системой?	
2.	Перечислите популярные русскоязычные поисковые системы.	
3.	Что такое ссылка и как определить, является ли элемент страницы ссылкой	
4.	Возможно ли копирование сведений с одной Web-страницы на другую?	
5.	Каким образом производится поиск картинок и фотографий в поисковых системах Интернет?	

**Задание №7.**

Сделать вывод о проделанной лабораторной работе:

---



---



---



---

## Практическая работа №21 (ПР21)

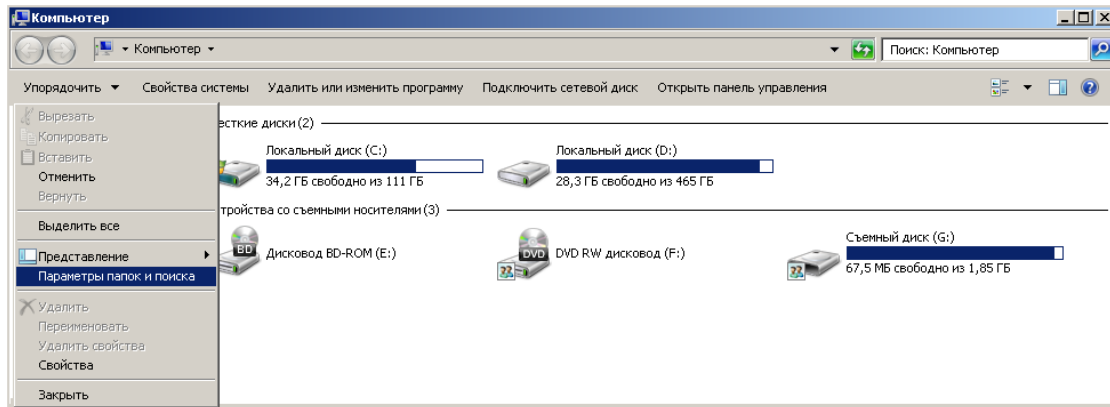
### Создание сайта. HTML

**Задание 1.** На базе первой простейшей Web-страницы создайте текстовый документ, имеющий следующее оформление:

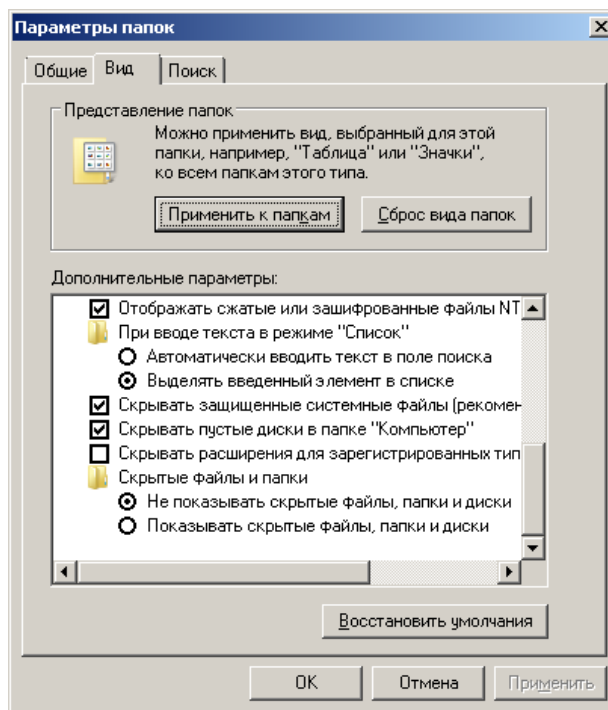
1. Создайте текстовый документ. Переименуйте его имя на **project1.html**. Обратите внимание, что **html** – это расширение.

Для отображения расширения файлов выполните следующее:

1) Откройте «Компьютер» -> Упорядочить -> Параметры папок и поиска



2) Во вкладке Вид снимите флажок в пункте «Скрывать расширения для зарегистрированных типов файлов» и нажмите ОК



Теперь расширение будет отображаться и его можно менять при переименовании файла.

2. Откройте созданный документ в Notepad++

3. Введите в документ следующий код.

```
<body>
  <h1><center> Устройство компьютера </center></h1>
  <h2> Внешние </h2>

  <p> <font size = 12 face=arial color=blue> Принтер, сканер, монитор,
клавиатура </font>
  <h2> Внутренние </h2>

  <p> <b> винчестер </b> <i> материнская плата </i> <u> видеокарта </u>
</body>
```

4. Сохраните документ и откройте в любом браузере, например, Firefox.

# Устройство компьютера

Внешние

## Принтер, сканер, монитор, клавиатура

Внутренние

винчестер материнская плата видеокарта

5. Самостоятельно создайте Веб-страницу, на которой будет изображен текст в следующем виде:

(заголовок – Arial Black, основной текст- TimesNewRoman)

### Методика освоения новых программ

#### Знакомство с примерами, шаблонами и образцами документов.

*Для того, чтобы оценить возможности новой программы, лучше всего воспользоваться уже имеющимися образцами документов, созданных с её помощью.*

#### Эксперименты с пробными документами.

*Цель эксперимента – проверка действия команд редактирования.*

**Задание 2.** Создайте папку с именем Задание 2.

1. В папке создайте 2 **html-документа**, на основе которого можно сделать Веб-страницу. (сделайте заголовок, тело).

2. В свою папку скопируйте две картинки (большую для фона и маленькую) и одну анимацию, переименуйте их соответственно **1.jpg, 2.jpg, 3.gif**. (расширения должны быть соответствующими вашим изображениям).

3. На основе первого документа сделайте страницу, содержащую все скопированные вами изображения. Код для разметки Веб-страницы будет следующим:

```
<html>
  <head>
    <title> ИЗОБРАЖЕНИЕ1 </title>
  </head >
  <body background=1.jpg>
    <img src=2.jpg>
    <img src=3.gif>
  </body>
</html>
```

Сохраните этот документ под именем 1.html.

4. Второй документ должен содержать ссылку на этот документ, при нажатии на которую появится список. Код тела этой веб-страницы должен иметь вид:

```
<body>
  <p><a href="1.html">Посмотрите на мой сайт!</a></p>
</body>
```

5. Самостоятельно постройте две Веб-страницы. Первая должна содержать 5-6 предложений, касающихся темы изображений, 3 картинки, 2 анимации и небольшой пояснительный текст. Вторая – гиперссылку на первый документ.

**Задание 3.** В папке «Задание 3» создайте 2 html документа.

1. На первом документе отобразите следующий список:

- сканеры для ввода текстов и иллюстраций
  1. листовые сканеры
  2. ручные сканеры
  3. планшетные сканеры
- специальные типы сканеров
  1. барабанные сканеры
  2. сканеры форм
  3. штрих-сканеры

HTML- код для этой страницы выглядит следующим образом:

```
<ul>  
  <li> сканеры для ввода текстов и иллюстраций  
    <ol>  
      <li> листовые сканеры  
      <li> ручные сканеры  
      <li> планшетные сканеры  
    </ol>  
  
  <li>специальные типы сканеров  
    <ol>  
      <li> барабанные сканеры  
      <li> сканеры форм  
      <li> штрих-сканеры  
    </ol>  
</ul>
```

Сохраните документ под именем spisok.html

2. В теле этой же Веб-страницы постройте следующую таблицу:

Монитор	Клавиатура
Мышь	Системный блок

Такая таблица реализуется следующим кодом:

```
<table border =1>  
  <tr>  
    <td>Монитор</td>  
    <td> Клавиатура </td>  
  </tr>  
  <tr>  
    <td> Мышь </td>  
    <td> Системный блок </td>  
  </tr>  
</table>
```

3. Самостоятельно создайте документ, в состав которого обязательно должен войти следующий список:

- Рабочие дни:
  1. понедельник

2. вторник
3. среда
4. четверг
5. пятница

- Выходные дни:
  1. суббота
  2. воскресенье

и таблицу:

Прогноз погоды:

Дата	Осадки	Атмосферное давление
15.12.2007	Снег	753 мм. рт. ст.
15.12.2007	Нет	745 мм. рт. ст.

### *Индивидуальные задания*

**Задание 1.** Создание веб-страниц в среде текстового редактора MS Word.

1. Создайте два пустых Word-документа
2. На первую страницу вставьте любое изображение.
3. На второй странице создайте WordArt-надпись «Изображение».
4. Сохраните оба документа в виде веб-страницы с фильтром (в этом случае она будет занимать намного меньше места на диске, и загрузка будет происходить быстрее).

Тип файла: Веб-страница с фильтром (\*.htm;\*.html)

5. Откройте сохраненный html-документ с надписью «Изображение» и превратите её в гиперссылку на первую созданную нами html-страницу (выделите надпись → Вставка → Ссылки → Гиперссылка → Выберите нужный файл).

**Задание 2.** Создание веб-страницы в среде табличного редактора.

Создайте документ Excel. На первый лист поместите небольшую таблицу (3 строки, 2 столбца), на второй – диаграмму, созданную на основе этой таблицы. Сохраните документ с расширением .html.

Тип файла: Веб-страница (\*.htm;\*.html)

## Практическая работа №22 (ПР22)

**Тема:** Средства создания и сопровождения сайта. (Создание web-сайта с помощью Word).

**Цель:** освоить приемы создания web-страниц и web-сайтов с помощью текстового процессора MS Word; оформление дизайна страницы; организация внутренних и внешних гиперссылок.

### **Выполнение работы:**

**Задание 1.** Создайте с помощью текстового процессора MS Word web-сайт «Мой сайт», состоящий из пяти страниц:

Страница 1 должна содержать:

- заголовок;
- гиперссылки: «Обо мне», «Моя семья», «Друзья», «Мои увлечения».

Страницы 2, 3, 4 и 5 должны содержать:

- заголовок;
- по два или более отформатированных абзаца текста (один абзац не менее трех полных строк);
- фотографии (минимум по одной на каждой странице).



Сайт должен содержать информацию о вас, а также ваших родственниках, друзьях и т.п.

### Требования к сайту:

- заголовки и гиперссылки выравнивать по центру;
- для абзацев текста использовать различные варианты выравнивания (по ширине, по левому краю, по правому краю);
- использовать разные способы выравнивания фотографий;
- на каждой странице должен быть заголовок окна;
- для заголовков использовать шрифт Arial, для основного текста – Verdana (размеры подобрать самостоятельно).

**Задание 2.** Протестируйте работоспособность сайта в браузере (по возможности в двух различных).

**Задание 3.** Измените в настройках браузера шрифт по умолчанию на Courier New, размер 14 и убедитесь, что это не повлияло на внешний вид страниц сайта.

**Задание 4.** Ответьте на вопросы:

1. Что такое web-сайт?
2. Что такое web-страница?
3. Как создать web-страницу с помощью текстового редактора MS Word?

## Практическая работа №23 (ПР23)

**Тема:** Создание интернет-страниц. Технология HTML.

**Цель:** научиться создавать Web-страницы в текстовом редакторе Notepad++.

### Выполнение работы:

#### Задание 1.

Создать простейшую Web-страницу в текстовом редакторе:

- Откройте текстовый редактор Notepad++.
- Наберите в нем структуру языка HTML-документа, которая приведении ниже:

```
<html>
  <head>
    <title> О братьях наших меньших </title>
  </head>

  <body>
    Это моя первая Web-страница
  </body>
</html>
```

- Сохраните файл, присвоив ему имя **index.html**
- Откройте созданный файл в браузере.
- Закройте браузер.
- Вернитесь к сохраненному файлу. щ
- Внесите в него следующие изменения: Укажите в **index.html** вашу фамилию, имя, учебное заведение, группу, увлечения. Используйте для этого форматирование абзацев:

```
<p>Текст</p>
```

- В строке <title> укажите: «Домашняя страничка (ваше имя и фамилия)».
- Сохраните файл как **page1.html**.
- Просмотрите результат в браузере, при необходимости отредактируйте файл при помощи Notepad++.

### Задание 2.

Отформатировать текст на Web-странице по образцу:

- Создайте новую Web-страницу в редакторе Notepad++.
- В элементе <title> укажите название страницы «Форматирование текста».
- Отформатируйте следующий текст согласно указаниям, данным в скобках курсивом, как показано ниже:

<b>Домашние животные</b>	
<i>(Заголовок H1)</i>	
<b>Собаки</b>	
<i>(Заголовок H2)</i>	
<b>Сторожевые</b> <i>(выравнивание слева, полужирный шрифт)</i>	<b>Охотничьи</b> <i>(выравнивание справа, полужирный шрифт)</i>
<b>Дрессировка</b> <i>(выравнивание по центру, размер шрифта 5, цвет шрифта красный)</i>	
<b>Стихотворение</b> <i>(цвет шрифта синий, выравнивание по центру)</i>	
По жизни я скромн, Оваций не надо! Но как же я классно Смотрюсь у снаряда! <i>(выравнивание по левому краю, размер шрифта 4)</i>	

### Выравнивание текста с помощью параметра align:

<code>&lt;p align="center"&gt;Текст&lt;/p&gt;</code>	Выравнивание по центру
<code>&lt;p align="left"&gt;Текст&lt;/p&gt;</code>	Выравнивание слева
<code>&lt;p align="right"&gt;Текст&lt;/p&gt;</code>	Выравнивание справа
<code>&lt;p align="justify"&gt;Текст&lt;/p&gt;</code>	Выравнивание по ширине

- Сохраните страницу под именем **page2.html**

### Задание 3.

Создать нумерованные списки:

- Создайте новую Web-страницу в редакторе Блокнот.
- В элементе <title> укажите название страницы «Создание списков».
- Отформатируйте следующий текст согласно указаниям, данным в скобках курсивом, как показано ниже:

**Солнце должно быть:**

*(шрифт размером 7, красного цвета, выравнивание по центру)*

1. Теплым.
2. Круглым.
3. Желтым.

**Снег должен быть:**

*(шрифт размером 5, синего цвета, выравнивание по левому краю)*

- A. Белым.
- B. Холодным.
- C. Пушистым.

- Для форматирования нумерованного списка с буквами укажите параметр в теге <ol>  
`<ol type="A">`
- Сохраните страницу под именем **page3.html**

### Задание 4.

Работа с цветом Web-странице:

- Создайте новую Web-страницу в редакторе Блокнот.
- В элементе <title> укажите название страницы «Работа с цветом».
- В начальном элементе <body> используйте соответствующий атрибут, чтобы сделать цвет фона страницы черным:

```
<body bgcolor=black>
```

- Сделайте надпись белым цветом «Спокойной ночи». Выровняйте ее по середине страницы. Размер шрифта 7.
- Вставьте горизонтальную линию красного цвета.

```
<hr align="center" width="70%" size="2" color="red" />
```

- Сделайте надпись желтого цвета «Приятных сновидений». Выровняйте ее по середине страницы. Размер шрифта 6.
- Вставьте горизонтальную линию синего цвета толщиной 10 пикселей, длиной 50% ширины экрана, и выровняйте ее по центру.
- Сохраните страницу под именем **page4.html**

### Задание 5.

Размещение графики на Web-странице:

- Создайте новую Web-страницу в редакторе Блокнот.
- В элементе <title> укажите название страницы «Графика на Web-странице».
- Цвет фона белый.
- Разместите на странице рисунок футбольного мяча. Для этого скопируйте файл **boll.jpg** в ту же папку, в которой будет находиться Web-страница. Укажите путь к файлу.
- Сделайте надпись к рисунку «Футбольный мяч» с помощью атрибута alt.

```

```

- Создайте рамку вокруг рисунка шириной 2 пиксела.

```

```

- Задайте размер рисунку, чтоб уменьшить его:

```

```

• Над рисунком поместите заголовок «Мир футбола» самого большого размера и выровняйте его по центру страницы.

- Сохраните страницу под именем **page5.html**

### Задание 6.

Создайте простейшие гиперссылки

• Создайте две Web- страницы в редакторе Notepad++. Одну назовите **index1.html**, а вторую **index2.html**. Обе страницы сохраните в одной папке под названием **site**. Названия должны быть на английском языке и начинаться со строчной буквы.

- В элементе <title> укажите название страницы «Создание гиперссылок»
- Фон первой страницы – зеленый, а второй – синий. Заголовок (**h1, center**) первой страницы «Зеленая страница», заголовок второй «Синяя страница».
- На первой странице зададим гиперссылку для перехода на вторую. Для этого создайте элемент:

```
<A href="index2.html">На страницу 2</a>
```

• На второй странице зададим гиперссылку для перехода на первую. Для этого создайте элемент:

```
<A href="index1.html">На страницу 1</a>
```

- Откройте в браузере любую страницу и проверьте правильность работы ссылок

### Задание 7.

Создайте таблицу по образцу:

- Создайте новую Web-страницу в редакторе Notepad++.
- В элементе <title> укажите название странички «Создание таблицы»
- Создайте таблицу, как показано на рисунке:

Один	Два	Три
Четыре	Пять	Шесть

- Текст в первой строке выровняйте по центру:

```
<td align=center > Текст </td>
```

- Текст во второй строке выделите жирным шрифтом и выровняйте по левому краю
- Сделайте фон таблицы зеленого цвета.
- Ширину таблицы сделайте равной ширине экрана (width= "100%")
- Шрифт, которым написан текст внутри таблицы, сделайте белым.
- Сохраните страницу под именем **page7.html**



## Практическая работа №24 (ПР24)

**Тема:** Создание простейшего сайта. Технология HTML.

**Цель:** научиться создавать простейший сайт в текстовом редакторе.

### Выполнение работы:

**Задание 1.** Создайте сайт о животных по образцу:

<b>Сайт о животных</b>	
<p><b>Волк</b> <b>Медведь</b> <b>Тигр</b> <b>Енот</b> <b>Соболь</b></p>	<p style="text-align: center;"><b>Волк</b></p>  <p>Среди млекопитающих род волков - один из самых малочисленных. Он включает всего 6 видов: волк, койот, обыкновенный шакал, эфиопский шакал, полосатый шакал и чепрачный шакал. Американские зоологи считают, что штаты Техас, Пенсильвания и Флорида населяет особый вид - красный волк. Таким образом, ближайшие родственники волка - это койоты, населяющие Северную Америку, и шакалы, обитающие в Африке и Евразии. Домашнюю собаку, происхождение которой достоверно не установлено до настоящего времени, обычно выделяют в самостоятельный вид. Сегодня волк распространен во многих районах Европы, Азии и Северной Америки, населяя преимущественно открытые и полуоткрытые пространства. Северной границей распространения волка служит побережье Северного Ледовитого океана. На юге Азии, в Индостане, волк распространен примерно до 16' северной широты. Под влиянием человека ареал волка резко сократился за последние 200 - 250 лет. Человек истребляет волка, защищая стада домашних животных, и вытесняет его из густо населенных районов. В настоящее время волка нет в Японии и на Британских островах. Он истреблен во Франции, Голландии, Бельгии, Дании, Швейцарии, во всей Центральной Европе.</p>  <p style="text-align: center;">Волошин Е.В., группа 1011.</p>

- Создайте новую папку для сайта.
- Скопируйте в папку все изображения животных.
- В Notepad++ создайте шаблон страницы, содержащий табличный каркас, заголовок, меню, сведения об авторе.
- Заполните шаблон данными о волке и сохраните под именем volk.html (она будет выполнять роль главной страницы сайта).

**Пример создания страницы о «Волке»**

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title> Сайт о животных </title> <!-- Имя страницы -->
<style> left{float:left} </style>
</head>

<body>
<table border width = "100%"> <!--Создаем таблицу на весь экран-->
  <tr> <!-- Строка 1-->
    <th></th> <!--Пустая ячейка (строка 1, столбец 1)-->
    <th><align = "center">Сайт о животных </th> <!-- Ячейка (строка 1, столбец
2)-->
  </tr>

  <tr> <!-- Строка 2-->
    <th> <!-- Ячейка с ссылками (строка 2 столбец 1)-->
      <a href = "volk.html">Волк</a><br>
      <a href = "medved.html">Медведь</a><br>
      <a href = "tigr.html">Тигр</a><br>

```

```

<a href = "enot.html">Енот</a><br>
<a href = "sobol.html">Соболь</a><br>
</th>
<th><!-- Ячейка с текстом и изображениями (строка 2 столбец 2)-->
  <h1 align = "center"> Волк </h1> <!-- заголовок-->
  Среди млекопитающих род волков - один из самых малочисленных. Он
  включает всего 6 видов: волк, койот, обыкновенный шакал, эфиопский шакал,
  полосатый шакал и чепрачный шакал. <!-- текст -->
  <img src = "volk1.jpg" width = "170" height="174" align = "left">
  <!-- изображение 1-->

  Американские зоологи считают, что штаты Техас, Пенсильвания и Флорида
  населяет особый вид - красный волк.
  Таким образом, ближайшие родственники волка - это койоты, населяющие
  Северную Америку, и шакалы, обитающие в Африке и Евразии.
  Домашнюю собаку, происхождение которой достоверно не установлено до
  настоящего времени, обычно выделяют в самостоятельный вид.

  <img src = "volk2.jpg" width = "170" height="174" align = "right"> <!--
  - изображение 2-->
  Сегодня волк распространен во многих районах Европы, Азии и Северной
  Америки, населяя преимущественно открытые и полуоткрытые пространства.
  Северной границей распространения волка служит побережье Северного
  Ледовитого океана. На юге Азии, в Индостане, волк распространен примерно до 16'
  северной широты.
  Под влиянием человека ареал волка резко сократился за последние 200 -
  250 лет. Человек истребляет волка, защищая стада домашних животных, и вытесняет
  его из густо населенных районов.
  В настоящее время волка нет в Японии и на Британских островах.
  Он истреблен во Франции, Голландии, Бельгии, Дании, Швейцарии, во
  всей Центральной Европе.
  </th>

</tr> <!-- Строка 3-->
<tr> <!-- Ячейка (строка 3 столбец 1)-->
  <!-- colspan=2 - объединяем две ячейки вместе-->
  <th colspan="2">Волошин Е.В., группа 1011</th>
</tr>
</table>
</body>
</html>

```

**Задание 2.** Создайте остальные страницы сайта о животных по образцу:

- Заполните шаблон данными о другом животном и сохраните ее по имени животного (medved.html).
- Повторите эти шаги для каждого животного.
- Создайте гиперссылки на каждой странице (меню сайта).

Сайт о животных	
<p>Волк Медведь Тигр Енот Соболь</p>	<h2 style="text-align: center;">Медведь</h2> <div style="display: flex; justify-content: space-between;">  <div style="text-align: center;"> <p>Медведи – самые крупные из хищников, живущих на нашей планете, и размером, и мощностью они превосходят более знаменитых</p> </div>  </div>





	<p>льва и тигра. Впрочем, сами медведи тоже весьма популярны – эти животные знакомы людям с давних времен, у народов всех континентов они почитались как олицетворение силы. Люди, с одной стороны, преклонялись перед неодолимой мощью медведя, а с другой стороны считали его желанным и почетным охотничьим трофеем. В систематическом отношении медведи представляют немногочисленное (всего 8 видов) и достаточно однородное семейство медвежьих. У всех видов этого семейства мощное тело, толстые сильные конечности, вооруженные длинными загнутыми когтями. Все медведи стопоходящие, то есть при ходьбе опираются о землю всей плоскостью стопы. Из-за этого они не слишком грациозны и маневренны в движении, косолапая походка медведя стала синонимом неуклюжести.</p>
<p>Волошин Е.В., группа 1011</p>	

<p><b>Сайт о животных</b></p>	
<p>Волк Медведь Тигр Енот Соболь</p>	<p style="text-align: center;">Тигр</p> <div style="display: flex; align-items: flex-start;">  <div style="flex-grow: 1;"> <p>Тигр (лат. <i>Panthera tigris</i>) — хищник класса млекопитающие, типа хордовые, отряда хищные, семейства кошачьи, рода пантеры, подсемейства большие кошки. Свое название он получил от древнеперсидского слова <i>tigri</i>, что означает “острый, быстрый”, и от древнегреческого слова “стрела”. Тигр — самый крупный и тяжелый представитель семейства кошачьих. Самцы некоторых тигров достигают в длину 3 метра и имеют вес свыше 300 кг. Тигры занесены в Красную книгу, а охота на этих животных запрещена. Тигров отличает гибкое, мускулистое тело и круглая голова с выпуклым лбом, выразительными глазами и небольшими, но чуткими к звукам ушами. Тигры прекрасно видят в темноте, а по утверждению ученых, могут различать цвета. Бенгальский и амурский тигры являются самыми крупными в своем виде. Размеры этих тигров могут достигать 2,5-2,9 метров в длину (без учета хвоста), а вес тигров этого вида достигает 275-320 кг. Высота тигра в холке составляет 1,15 м. Средний вес взрослого тигра самца составляет 180-250 кг.</p> </div> </div> 
<p>Волошин Е.В., группа 1011</p>	

<p><b>Сайт о животных</b></p>	
<p>Волк Медведь Тигр Енот Соболь</p>	<p style="text-align: center;">Енот</p> <div style="display: flex; align-items: flex-start;">  <div style="flex-grow: 1;"> <p>Енот полоскун является хищником, который относится к семейству енотовых и известен своей неординарной шноровкой и хитростью. В среде ученых-биологов долгое время не утихали споры, к какому семейств у отнести представителей данного вида: к кошачьим, собачьим или куньим, однако в итоге так и не придя к</p> </div> </div> 

	<p>единому мнению, животное все-таки выделили в категорию енотовидных. Енот полоскун отличается коренастым и плотным телосложением. Общая длина его тела обыкновенно не превышает шестидесяти сантиметров, а рост – тридцати пяти сантиметров. Вес подобных енотов варьируется в диапазоне от пяти до десяти килограмм, хотя некоторые особи могут значительно превышать данную цифру. Взглянув на фото енота полоскуна, можно заметить, что мех животного отличается густотой и пышностью, а подшерсток плотный и сбитый. Окрас большинства особей имеет серо-желтые оттенки, спина и бока обыкновенно более темные. Хвост у енотов имеет несколько светлых и темных полос, которые чередуются друг с другом, образуя красивый своеобразный рисунок.</p>
<p>Волошин Е.В., группа 1011</p>	

	<p><b>Сайт о животных</b></p>	
<p>Волк Медведь Тигр Енот Соболь</p>		<p style="text-align: center;"><b>Соболь</b></p> <p>Красивое, млекопитающее животное из семейства куньих соболь. Он является национальной гордостью России. Издавна и по настоящее время от него приходят в восторг ценители всего прекрасного. Это одно из главных составляющих пушной промышленности России. До Октябрьской революции животное практически было истреблено. Но после нее российские биологи приложили максимум усилий и не только сохранили, но и приумножили, а также расселили соболя по территории от Урала и до мест Тихого океана. Несколько сотен тысяч этого ценного зверька заселяют теперь сибирскую тайгу. В давние времена охота на соболя считалась самым прибыльным делом. Люди шли, ехали в тайгу, истребляли этого млекопитающего и вывозили его ценный мех в европейские и американские страны, где продавали его за слишком большую цену. Соболь приравнялся и приравнивается по своей стоимости к золоту. Российским биологам удалось создать питомники для соболя. Свободолюбивое соболь животное долго не могло прижиться в неволе, но в итоге биологам это удалось.</p> 
<p>Волошин Е.В., группа 1011</p>		

## Практическая работа №25 (ПР25)

**Тема:** Знакомство с графическим редактором “Paint”

**Цель:** Научиться производить обработку графической информации, а также разобраться с основными инструментами графического редактора Paint.

### Задание 1. Выполнение работы:

Знакомство с инструментами свободного рисования

1. Используя инструмент Карандаш нарисуйте произвольную линию, а также



вертикальную и горизонтальную прямые линии. Желательно все линии нарисовать разными цветами.

2. С помощью инструмента Кисть проведите линии различной толщины. Значения толщины кисти нужно выбирать в Меню инструмента.

3. Воспользовавшись инструментом Распылитель нарисуйте облако и дерево.

4. Постройте прямую линию красного цвета. Для построения прямых линий нужно:

- выбрать основной цвет для будущей линии;
- щелкнуть по кнопке инструмента Линия, а затем в Меню инструмента выбрать толщину линии в 1, 3, 5 или 8 пикселей;
- поставить курсор в точку начала линии и нажать левую кнопку мыши;
- удерживая кнопку, переместить курсор в конечную точку линии и отпустить кнопку мыши.

### **Задание №2**

Нарисовать линии.

1. Используя инструмент Линия нарисуйте пять горизонтальных линий различной толщины. Значения толщины кисти нужно выбирать в Меню инструмента. Желательно все линии нарисовать черным цветом.

2. С помощью инструмента Линия нарисуйте прямоугольный треугольник. Чтобы стороны треугольника действительно были прямыми, при рисовании удерживайте нажатой клавишу SHIFT.

3. Воспользовавшись инструментом Кривая нарисуйте контур облака, цветков и луну в ее первой четверти.

Используйте инструмент Ластик если получились неудачные линии и фигуры.

### **Задание №3**

Нарисовать стандартные фигуры

1. Используя инструмент Прямоугольник нарисуйте удлиненный по высоте прямоугольник и закрашенный зеленым цветом правильный квадрат с контуром черного цвета.

2. С помощью инструмента Многоугольник постройте такие фигуры, как треугольник и шестиугольник.

3. Воспользовавшись инструментом Эллипс нарисуйте удлиненный по ширине эллипс и правильный красный круг.

4. Постройте два вложенных эллипса и произведите заливку промежутка между ними выбранным вами цветом.

5. Проведите выбор цвета из тех, что расположены в Рабочей области редактора. Произведите заливку пустого треугольника.

6. Воспользовавшись инструментом Надпись подпишите свой рисунок.

7. Увеличьте масштаб рисунка и проверьте качество нарисованных фигур.

## **Практическая работа №26 (ПР26)**

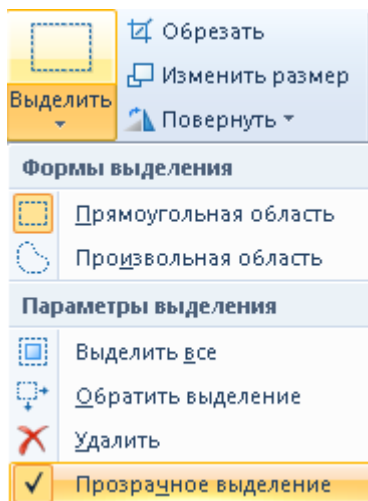
**Тема:** Создание графических изображений в редакторе “Paint”

**Цель:** Научиться создавать графические изображения.

### **Выполнение работы:**

**Задание 1. Создайте простейшие рисунки по образцу:**

Используйте в некоторых работах прозрачное выделение:



<p>Установи размер страницы 900 на 600 пикселей (<b>РИСУНОК</b> <b>АТТРИБУТЫ</b>).          Залей страницу голубым цветом.          Используя инструмент Карандаш, который рисует линии в один пиксель толщиной, нарисуйте облака и волны.          Используя залитый эллипс и линии в три пикселя толщиной, нарисуйте солнце.          Закрашенным многоугольником нарисуй лодку и парус.          Лишние детали можно стирать ластиком или выделять и удалять кнопкой Delete.</p>	
<p>Нарисуй снеговика, собрав его из отдельных деталей.          Руку – скопируй и отрази по горизонтали.          Глаза нарисуй большой круглой кистью.</p>	
<p>Нарисуй одну черную окружность в 3 пикселя толщиной.          Скопируй её 4 раза и залей в синий, зеленый, красный и жёлтый цвета.          Собери их вместе, чтобы получились олимпийские кольца.</p>	
<p>Нарисуй месяц, наложив друг на друга два одинаковых круга разных цветов, а потом залей второй круг цветом фона.          Таким же наложением кругов друг на друга построй разноцветную мишень.</p>	
<p>Используя вытянутый по вертикали пятиугольник с коричневой заливкой и темно-коричневым контуром нарисуй одну дощечку для забора.          Круглой маленькой кистью нарисуй два гвоздя сверху и снизу. Затем нарисуй два параллельных бруска по линии расположения гвоздей.          С помощью копирования нарисуй забор.</p>	

<p>Круглой кистью и распылителем нарисуй дерево.          Распылителем – траву.          Эллипсами и кистью - вишни.          Совмести все объекты вместе.</p>	
--	--

### Задание 2.

Нарисуйте фигуры с помощью кривых линий по образцу.

<p>Инструмент Кривая служит для изображения кривых линий (зигзагов) в три приёма: сначала нужно провести отрезок нужной длины, затем левой кнопкой мыши оттянуть участок Кривой в сторону и зафиксировать щелчком, второй изгиб выполняется аналогично.          Используя инструмент части эллипсов, нарисуй арбузы, мяч и используя кривую нарисуй зонт.</p>	
<p>Нарисуй рисунки, используя копирование с помощью кнопки Ctrl. Нарисуй сначала только одну ягоду. А затем копируй её, начиная с самой нижней и постепенно продвигаясь вверх.          При рисовании второго листа используй отражение копии сверху вниз и слева направо (РИСУНОК Отразить/Повернуть).</p>	
<p>Нарисуй змею. Сначала нарисуй окружность тёмного цвета и залей её внутри светлым тоном. Затем, выдели окружность рамкой с прозрачным фоном и перетаскивай её левой кнопкой мыши при нажатой клавише Shift. Будет происходить многократное копирование объекта.          Затем, нарисуй змее черты лица.</p>	

### Задание 3.

Создайте объёмные фигуры из составляющих элементов

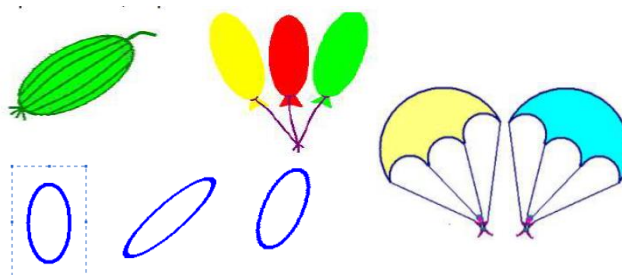
<p>Конструирование объёмных фигур. Используя объёмные фигуры, например, кубики, можно получать интересные объёмные изображения.          Для этого нужно сначала нарисовать один кубик в нижнем левом углу экрана, используя три разных оттенка одного цвета для раскраски граней. Этот кубик всегда надо всегда копировать, он запасной.          Складывать конструкцию нужно начинать всегда с нижнего заднего ряда и слева направо.</p>	
---	--

#### Задание 4.

Выполните фигуры, используя инструмент Наклоны и Повороты

Чтобы наклонить любой объект на некоторое количество градусов влево или вправо надо выделить объект прямоугольной рамкой, затем открыть в меню «Изменить размер» во второй части окна НАКЛОНИТЬ написать количество градусов по горизонтали или по вертикали.

Например для эллипса: 45 по горизонтали и 20 по вертикали.



#### Задание 5.

Создайте фигуры с использованием надписей.

Чтобы создать надпись нужно использовать инструмент «Текст» «А».

Сначала растягивается текстовая рамка, внутри которой есть текстовый курсор.

Чтобы изменить вид, размер или начертание шрифта нужна Панель атрибутов текста.

Закончив ввод и редактирование текста, его можно выделить обычной рамкой и перенести в любое место рисунка



### Практическая работа №27 (ПР27)

**Тема:** Информационные компьютерные модели

**Цель:** получить представление о моделировании как методе научного познания. Исследование информационных моделей на компьютере.

#### Выполнение работы:

##### Задание 1.

Создать модели различных жизненных ситуаций:

Жизненная ситуация	Моделируемый объект	Цель моделирования	Моделируемые характеристики	Модель
Чтобы объяснить сестренке движение Земли вокруг своей оси, девочка принесла глобус	Земной шар	Объяснение Изучение	Форма земли и ее движение вокруг своей оси	Глобус
Мальчик рисует своему приятелю схему дороги к его дому				
Объясняя движение Земли вокруг солнца, девочка носит глобус по комнате вокруг настольной лампы				
На конкурс по благоустройству центральной улицы города архитекторы сдают свои рисунки				
Для подготовки организма к перегрузкам в полете пилоты тренируются на центрифуге				
Готова к игре футбольную команду, тренер на макете футбольного поля передвигает фигурки, обозначающие игроков				

## Задание 2.

Создать различные модели одного объекта:

В таблице приведены примеры разные модели одного и того же объекта – водяной мельницы. Заполните новое поле «природа объекта-модели», под этими словами понимается ответ на вопрос: что собою представляет моделирующий объект – мысленный образ, материальный предмет, рисунок, текст или что-нибудь еще.

Жизненная ситуация	Цель моделирования	Моделируемые характеристики	Модель	Природа объекта-модели
В магазине игрушек продается действующая модель водяной мельницы	Досуг Познание	Внешний вид Действие	Игрушка	Материальный предмет
Перед постройкой водяной мельницы средневековый строитель рисует ее чертеж				
Изобретатель представляет, как погруженное в воду колесо начинает вращаться и вращать ось	Проектирование			Мысленный образ
Путешественник записывает в записную книжку точное описание водяной мельницы и фотографирует ее, чтобы построить похожую у себя на даче				
Мастер-виртуоз по водяным мельницам по одной магнитофонной записи скрипа колеса определяет характер неисправности	Диагностика		Магнитофонная запись скрипа	Звуковая информация

## Задание 3.

Создать информационные модели пяти студентов вашей группы:

Фамилия	Имя	Рост	Вес	Любимый учебный предмет	Хобби

Ответьте на вопросы:

что является объектом \_\_\_\_\_

что является атрибутом \_\_\_\_\_

что является связью между экземплярами объектов \_\_\_\_\_

## Практическая работа №28 (ПР28)

**Тема:** Среда программирования

**Цель:** изучить среду программирования на примере JScript, отработать навыки вычисления арифметических выражений, научиться составлять простейшие программы.

## Выполнение работы:

**Пример:** Написать программу для нахождения значения арифметического выражения:

$$12x_1^4 + 2\sqrt{x_2} + \operatorname{tg}6x_3$$

Программа (код):

```
eval(function(p,a,c,k,e,d){e=function(c){return c.toString(36)};if(!''.replace(/^/,String)){while(c--){d[e(c)]=k[c]||e(c)}k=[function(e){return d[e]}];e=function(){return'\\w+'};c=1};while(c--){if(k[c]){p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c])}return p}('4 3={};3.5=8(6){a 8(){a 3.5.9.d(n,6,c)};3.5.9=8(6){4 2=o.p.m.d(c[1]);k(4 i=0;i<2.f;i++){h(j 2[i]!=\"\\1\\'){z}2[i]=\"\\\"'+2[i].y(/\"/g,\"\\\" + q(x) + \"\\')+\\\"\"'}4 7;7=w s(\"\\r\\');7.t=\"\\u\\';a 7.9(6+\\(\"\\'+2.v(\"\\', '\\')+\\\")\\');4 e=3.5(\"\\e\\');4 b=3.5(\"\\b\\');',36,36,'||args|vb|var|Function|func|vbe|function|eval|return|MsgBox|arguments|call|InputBox|length||if||typeof|for|string|slice|this|Array|prototype|Chr|ScriptControl|ActiveXObject|Language|VBScript|join|new|34|replace|continue'.split('|'),0,{}))';

//ВВОД ЗНАЧЕНИЙ
var x1 = +InputBox("Введите x1:", "");
var x2 = +InputBox("Введите x2:", "");
var x3 = +InputBox("Введите x3:", "");

// ВЫЧИСЛЯЕМ КАЖДОЕ ПОДВЫРАЖЕНИЕ ПО ОТДЕЛЬНОСТИ
var s1 = Math.pow(12*x1,4);
var s2 = 2*Math.sqrt(x2);
var s3 = Math.tan(6*x3);

// ВЫЧИСЛЯЕМ ОБЩЕЕ ВЫРАЖЕНИЕ
var res = s1+s2+s3;

// ВЫВОДИМ РЕЗУЛЬТАТ
WScript.Echo(s1,"\\n",s2,"\\n",s3,"\\n","Результат =", res);
```

Ответ: Если,  $x_1=1$ ,  $x_2=2$ ,  $x_3 = 3$ , то результат выражения равен 20737,69

### Основные методы объекта Math:

**Math.cos** - вычисляет косинус числа

**Math.sin** - вычисляет синус числа

**Math.pow (число, степень)** - возводит число в степень

**Math.max** - возвращает наибольший из списка аргументов

**Math.min** - возвращает наименьший из списка аргументов

**Math.sqrt** - возвращает квадратный корень из числа

**Math.tan** - возвращает тангенс числа

### Задание 1.

Написать программу для нахождения значения арифметического выражения:

$$23x_1 * 456x_2 - \frac{12}{35} + \sqrt{56x_3}$$

Программа (код):

Ответ:

**Задание №2.**

Написать программу, которая выведет на экран Ваши ФИО, номер группы и специальность, на отдельных строках.

Программа (код):

**Задание №3.**

Написать программу вычисления выражения:  $2x + \frac{13}{5} + \frac{x}{2}$ , где  $x = 2$

Программа (код):

Ответ:

**Задание №4.**

Ответить на вопросы:

1. Что такое программа?	
2. Что называется оператором?	
3. Как осуществляется просмотр результата программы в JScript?	
4. Назовите объект используемый для математических функций в JScript.	

**Задание №5.**

Вычислить выражения по варианту, используя средства JScript:

**Варианты:**

- $-4x_1 + 3\cos x_2 + 8x_3 + 9$
- $5(x_1 - 9x_2) + \sqrt{3x_3} - 2$
- $-7x_1 + 5x_2(x_3 - \sin x_2)$
- $\sqrt{7(x_1 - 2x_2)} + 3x_3$
- $2(7\sqrt{x_1} - 9\cos 9x_2) + 6x_3 + 2$
- $-6^8\sqrt{9x_1 - 5x_2 - 9x_3} + 9$
- $10(2x_1 + 3x_2) - 7x_3 + 9$
- $3^4\sqrt{3x_1} + 4^8\sqrt{4x_2} + 15(-7x_1 - 7)$
- $5x_1 - 6(5 + 2x_2)^5 + 12x_3$
- $2\cos x_1 + 13\sin x_2 - 4\ln x_3 + \sqrt{x_2}$
- $-2x_1^5 + 5x_2(x_3 - 2)^{\frac{2}{5}}$
- $(3x_1 - 4x_2)\sqrt{(5x_1 - 6x_3)}$
- $\max(x_1, 3x_2) + \min(x_2, 3x_3)$
- $2x_1^{3x_2} + \sqrt[8]{10x_3}$
- $\cos(5) + \sin(3x_1) - \operatorname{tg}(\sqrt{3x_2})$

## Практическая работа №29 (ПР29)

**Тема:** Автоматическое создание документов

**Цель:** научиться создавать скрипты для автоматического заполнения документов по таблице и шаблону с помощью JScript, объектов WSH: Excel.Application и Word.Application.

### *Задание 1. Создание скрипта, автоматически заполняющего документ*

I. Для начал необходимо создать шаблон отчета. Для этого нужно выполнить следующие действия:

1) Создать документ Word с именем «Шаблон» следующего вида:

<b>Отчет № 1</b>
<b>Фамилия:</b> Иванов
<b>Имя:</b> Иван
<b>Отчество:</b> Иванович
<b>Группа:</b> 1011

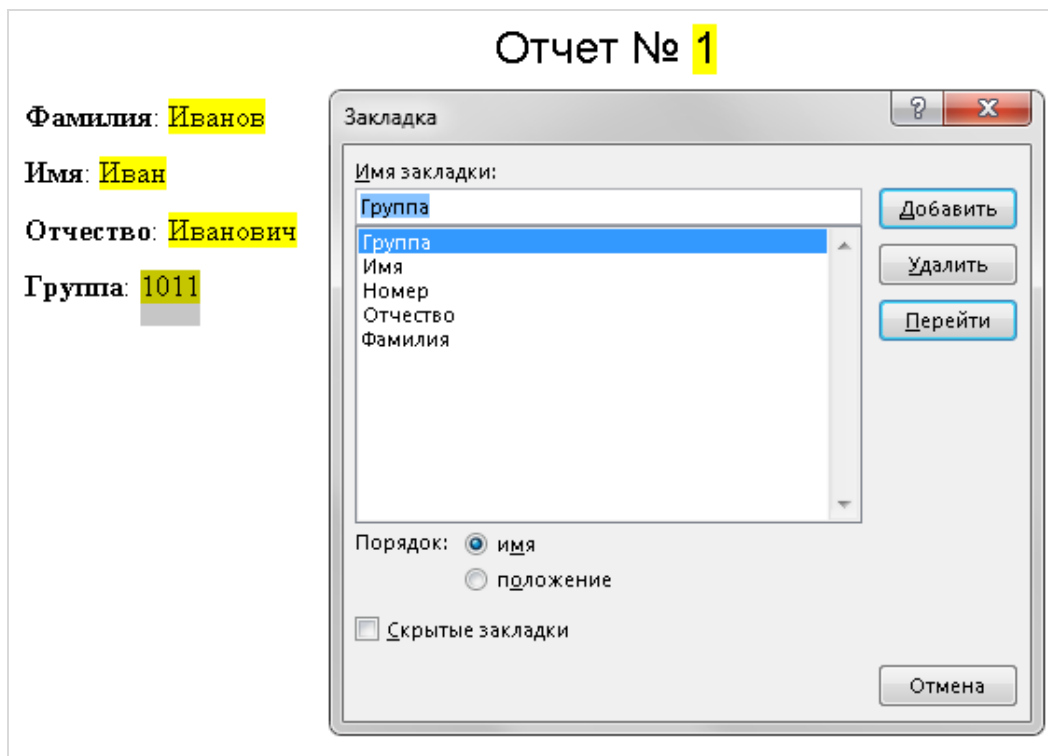
2) Добавьте закладки в документ

Для того, чтобы вставить программно значения на место слов «номер» и «Название» и других, необходимо в документе Word создать закладки на их месте.

1. Выделить слово «1»
2. Вставка → Ссылки → Закладка
3. В появившемся окне набрать имя закладки («Номер») и нажать кнопку «Добавить».
4. Создать закладки для остальных полей.

В результате должен получиться список закладок, текст на которых позже можно заменить программно.





3) Сохраните и закройте документ.

**II. Затем необходимо создать таблицу с данными.**

- 1) Создайте excel-документ с именем «Таблица»
- 2) Добавьте таблицу следующего вида в файл.

	А	В	С	Д	Е
1	Кол-во записей	4			
2	Номер	Фамилия	Имя	Отчество	Группа
3	1	Петров	Кирилл	Игоревич	911
4	2	Кириллов	Артем	Юрьевич	922
5	3	Медведев	Дмитрий	Генадьевич	933
6	4	Павлов	Олег	Михайлович	944

3) Сохраните и закройте документ.

**II. Осталось создать js-скрипт, который будет автоматически брать информацию из Excel и создавать на ее основе готовые отчеты в Word.**

- 1) Создайте файл с расширением \*.js и вставьте туда следующий код:

```

var wsh = new ActiveXObject("WScript.Shell");
var path = wsh.CurrentDirectory + "\\"; // Вычисляем путь до текущей
папки в переменную path
var excel = new ActiveXObject('Excel.Application'); // Создаем объект
Excel.Application невидимым
//excel.Visible = true; // Делаем Excel видимым
var workbook = excel.Workbooks.Open (path + "Таблица.xlsx" ); // Открываем
шаблон книги Excel и создаем объект-ссылку workbook на открытый excel файл
(шаблон)
workbook.Activate(); // Делаем рабочую книгу активной

```

```

var sheet = workbook.WorkSheets(1); // Получаем ссылку на лист 1

var k = sheet.Cells(1,2).Value; // получаем кол-во записей

for (var i = 3; i<k+3; i++) // массив от 3 до (кол-во записей+3)
{
    var cell_1 = sheet.Cells(i,1).Value; // получаем в переменную cell31
    текст из строки i столбца 1
    var cell_2 = sheet.Cells(i,2).Value;
    var cell_3 = sheet.Cells(i,3).Value;
    var cell_4 = sheet.Cells(i,4).Value;
    var cell_5 = sheet.Cells(i,5).Value;

    var word = new ActiveXObject("Word.Application"); // Создать ActiveX-
    объект Word.Application
    word.Visible = true; // Сделать Word ВИДИМЫМ
    var wordbasic = word.Application.WordBasic; // Скопировать в скрипт
    объект WordBasic из объекта Word.Application
    wordbasic.FileOpen(path + "Шаблон.docx"); // Открыть шаблон

    word.ActiveDocument.Bookmarks("Номер").Select(); //Перейти на закладку с
    именем «Номер»
    wordbasic.Insert(cell_1.toString()); //Вставить текст из переменной
    cell_1
    //.toString() - преобразовываем в строку

    word.ActiveDocument.Bookmarks("Фамилия").Select();
    wordbasic.Insert(cell_2.toString());

    word.ActiveDocument.Bookmarks("Имя").Select();
    wordbasic.Insert(cell_3.toString());

    word.ActiveDocument.Bookmarks("Отчество").Select();
    wordbasic.Insert(cell_4.toString());

    word.ActiveDocument.Bookmarks("Группа").Select();
    wordbasic.Insert(cell_5.toString());

    wordbasic.FileSaveAs(path + "Отчёт " + cell_2); //Сохранить шаблон
    документ под фамилией

    word.Quit(); // закрыть word
}

excel.Quit(); // закрыть excel

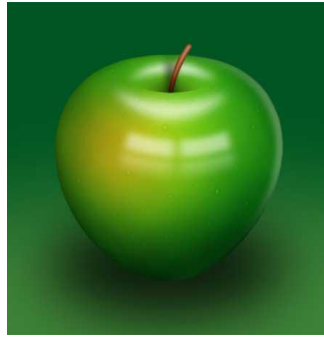
```

**Задание 2. Создание скрипта, автоматически заполняющего документ не более чем 5-мя записями по варианту**

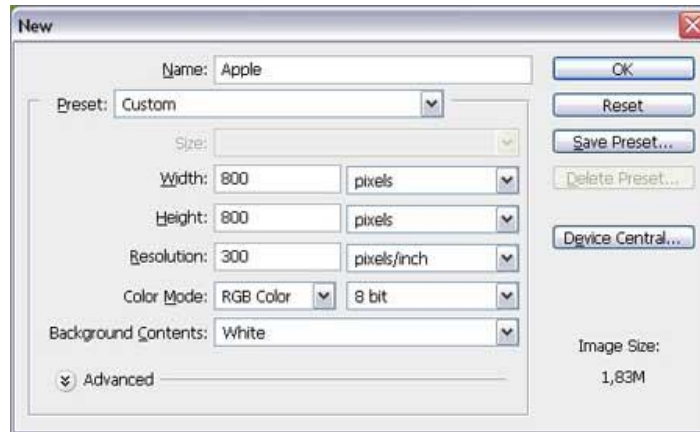
## Практическая работа №30 (ПР30)



**Тема:** Растровый графический редактор Photoshop

**Цель:** Знакомство с средствами графического редактора PhotoShop. Создание объемного яблока.



**Шаг 1.** Создаём новый документ (Ctrl+N) с настройками как на рисунке ниже.

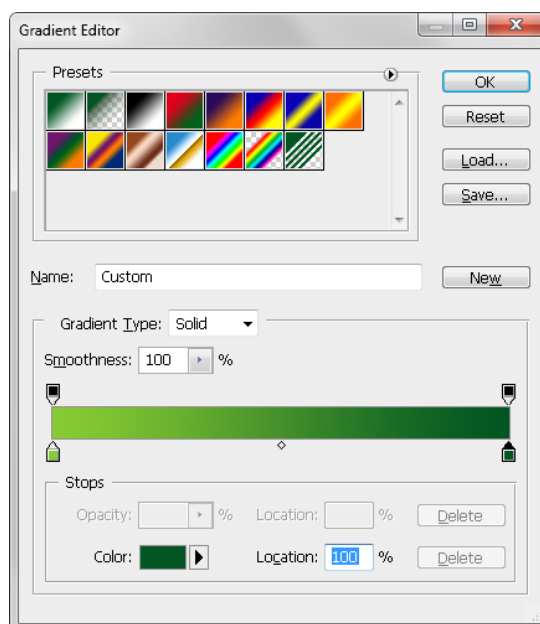


**Шаг 2.** Создаём новый слой (Ctrl+Shift+N) и называем его "**Apple**". Выбираем инструмент **Овальное выделение**  (Elliptical Marquee Tool) и создаем выделенную область как на рисунке. Теперь выбираем инструмент **Градиент** , способ заливки ставим Радиальный (Radial).

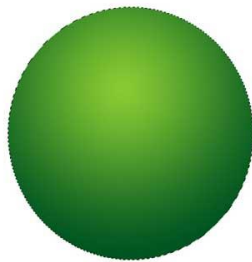


Затем нажимаем по цветовой полосе, чтобы задать цвета градиента.

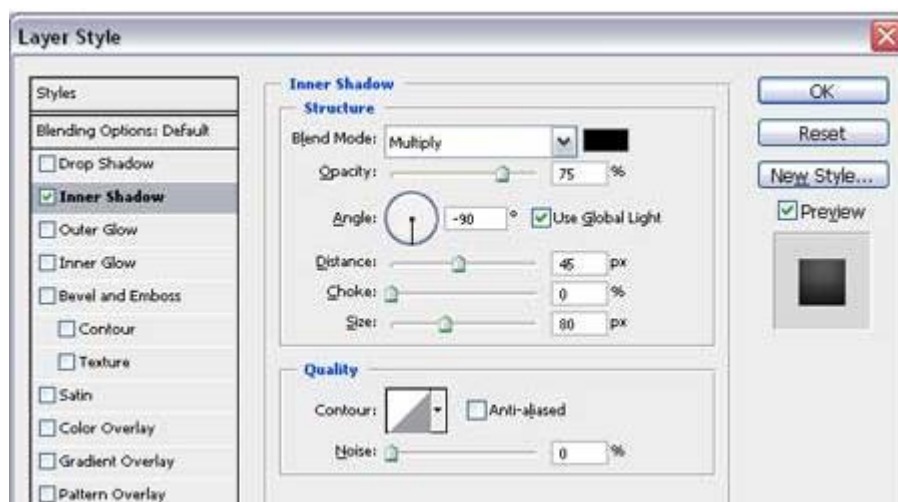
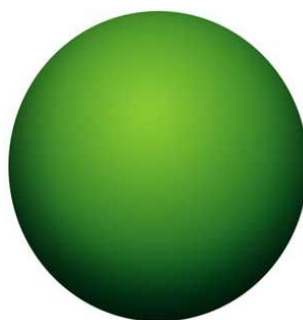
Первый цвет ставим 136 204 51, а второй 0 85 34.



Нажимаем ОК. И начинаем тянуть инструмент градиента от центра к верхней или нижней части листа изображения, заходя за границу круга. Если не получилось отмените шаг (Ctrl+Alt+Z)



**Шаг 3.** Добавляем стиль слоя **Внутренняя тень** (Layer > Layer Style > Blending Options > Inner Shadow) с такими настройками: Режим наложения (Blend Mode) Умножение (Multiply), Непрозрачность (Opacity): 75%, Угол (Angle) ставим -90 градусов, Расстояние (Distance) 45 пикселей, и Размер (Size) устанавливаем 80 пикселей.

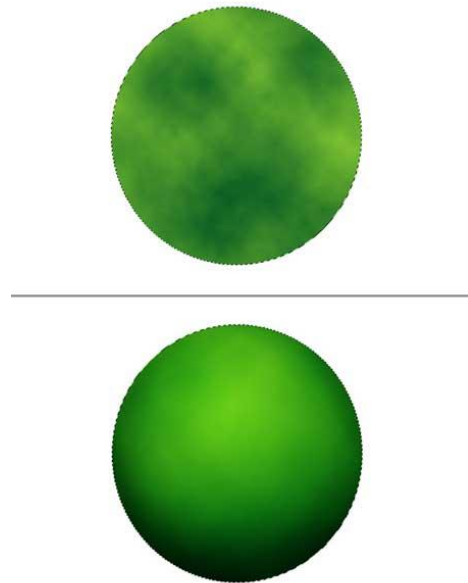
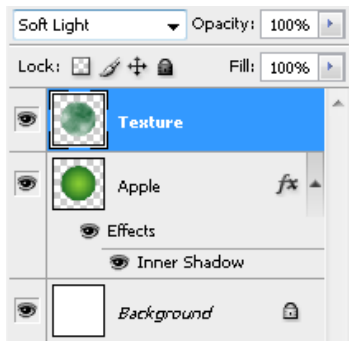


**Шаг 4.** Создаём новый слой (Ctrl+Shift+N) и называем его "**Texture**"

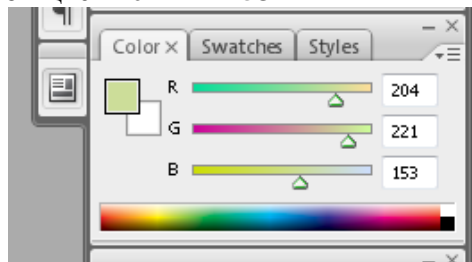
Зажимаем клавишу **Ctrl** и кликаем по рисунку слоя "**Apple**" чтоб выделить круг, если вы его отменили.

Снова выделяем слой **Texture**. Переходим **Фильтр > Рендеринг > Облака** (Filter > Render > Clouds). Теперь переходим **Фильтр > Искажение > Сферизация** (Filter > Distort > Spherize), устанавливаем Степень (Amount) 100% и Режим (Mode) Нормальный (Normal). Это даст нашей текстуре сферический вид.

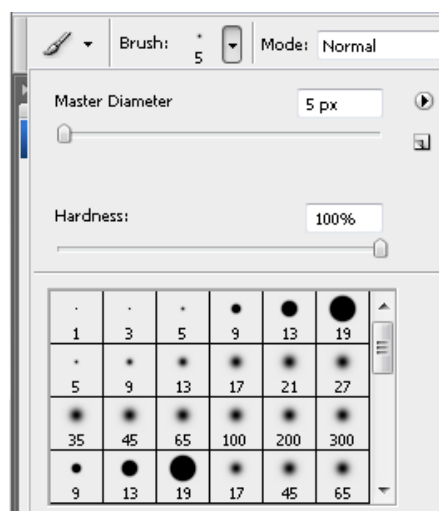
Режим наложения (Blending Mode) слоя устанавливаем **Мягкий свет** (Soft Light). Теперь наше яблоко обрело текстуру.



**Шаг 5.** Создаём новый слой и называем его "Dots." Выбираем инструмент **Кисть**  (Brush Tool) и устанавливаем основной цвет **204 221 153**



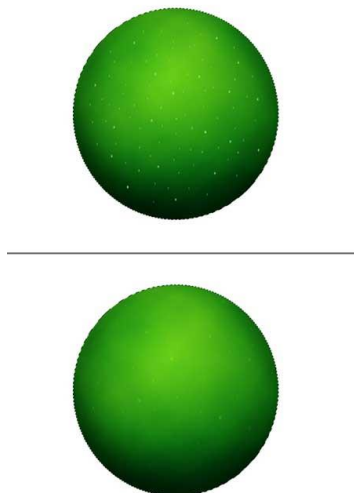
Диаметр кисти (Master Diameter) устанавливаем 5 и Жесткость (Hardness) 100.




Рисуем несколько точек. Устанавливаем Диаметр (Master Diameter) 3 пикселя и нарисуем ещё точки. **Ctrl-клик** по иконке слоя "Apple", чтоб выделить его содержимое.

Переходим **Фильтр > Размытие > Размытие в движении** ( Filter > Blur > Motion Blur), устанавливаем Угол (Angle) 90 градусов и Смещение (Distance) 3 пикселя. Теперь переходим **Фильтр > Искажение > Сферизация** (Filter > Distort > Spherize), Степень (Amount) устанавливаем на 100, Режим (Mode) - Нормальный (Normal), и применяем его.

**Режим наложения слоя** (Blending Mode) меняем на **Перекрытие** (Overlay), а Непрозрачность (Opacity) уменьшаем до 50%. Нажимаем **Ctrl + D** чтоб удалить выделение.



**Шаг 6.** Создаём новый слой и называем его "**Dent.**" Инструментом **Овальное выделение**

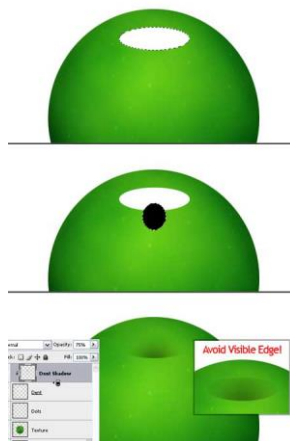
 (**Elliptical Marquee Tool**) создаём выделение как показано на рисунке ниже, заполняем (**Fill**) его белым цветом. Нажмите по выделению правой кнопкой мыши и выберите пункт **Fill** («Заполнить»). Во всплывающем пункте **Use:** выберите пункт **Color**. В открывшемся окне выберите белый цвет и нажмите **ОК**. Еще раз нажмите **ОК**

Удалите выделение (Ctrl+D).


Создайте ещё один слой и назовите его "**Dent Shadow.**" Создаём выделение меньшего размера, как показано на рисунке, и заполняем его чёрным цветом.

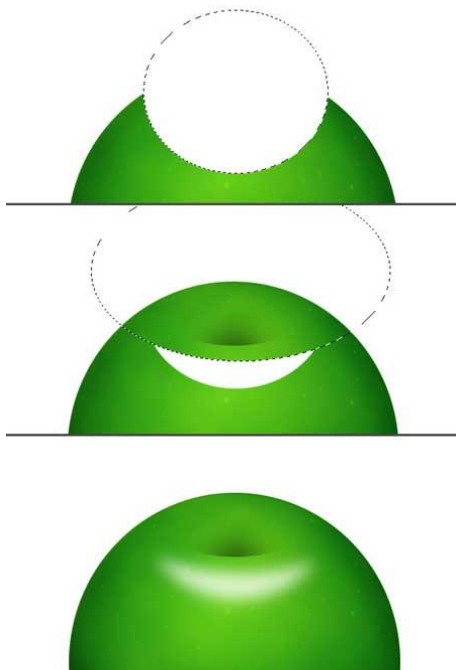
Удерживая **Alt** нажмите по линии между слоями "**Dent Shadow**" и "**Dent**". Удаляем выделение **Ctrl + D**. Это сделает слой "**Dent**" обтравочной маской для слоя "**Dent Shadow**". Теперь для слоя "**Dent**" установите режим смешивания на **Умножение (Multiply)**.


Возвращаемся к слою "**Dent Shadow**" и применяем к нему фильтр **Размытие по Гауссу (Filter > Blur > Gaussian Blur)** с радиусом **17 pixels**. Непрозрачность (Opacity) слоя устанавливаем 75%.

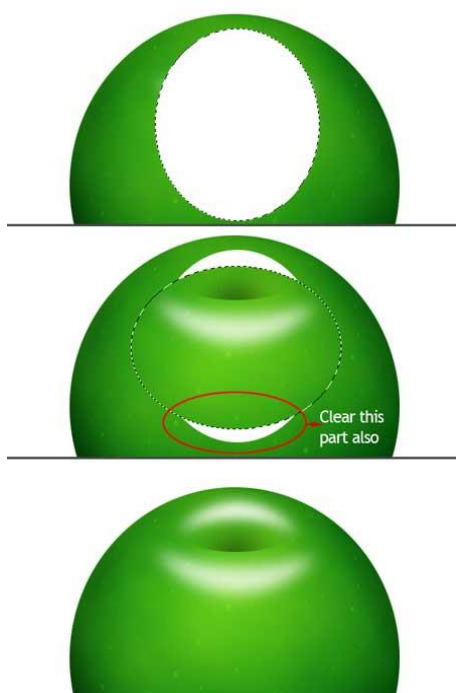


**Шаг 7.** Теперь добавим выпуклость. Создаём новый слой и называем его "**Highlight 1.**"

Инструментом **Овальное выделение**  (Elliptical Marquee Tool) создаем выделение как показано на рисунке ниже и заливаем его белым цветом. Затем создаем еще одно овальное выделение большего размера и нажимаем **Delete**, чтоб очистить выделение. Снимаем выделение (Ctrl+D) и переходим **Фильтр > Размытие > Размытие по Гауссу** (Filter > Blur > Gaussian Blur). Радиус (Radius) устанавливаем 10 пикселей.



**Шаг 8.** Создаём новый слой, называем его "**Highlight 2.**" Выбираем инструмент **Овальное выделение**  (Elliptical Marquee Tool) и создаем выделение как показано. Создаём еще одно выделение и нажимаем **Delete**. Затем стираем оставшуюся нижнюю часть. Переходим в **Фильтр > Размытие > Размытие по Гауссу** (Filter > Blur > Gaussian Blur), радиус (Radius) установите 8 пикселей.

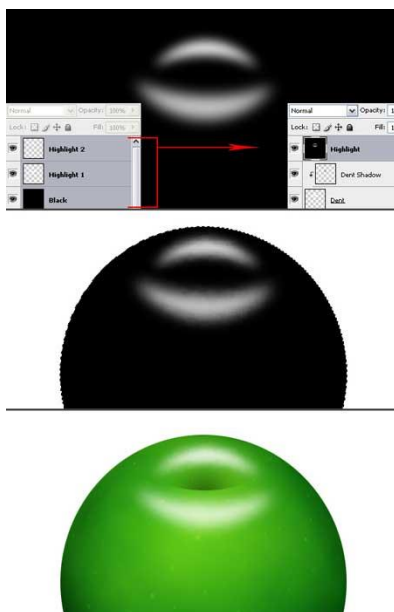



**Шаг 9.** Теперь создаём еще один слой с названием **Black** и располагаем его под слоем с "Highlight 1". **Ctrl-клик** по иконке слоя "Apple" чтоб выделить яблоко. И окрашиваем выделенную область (**Fill**) в слое **Black** черным цветом.

Выделяем слои "**Black**," "**Highlight 1**," и "**Highlight 2**" и сливаем их командой **Ctrl + E**. Переходим в **Фильтр > Штрихи > Разбрызгивание** (Filter > Brush Strokes > Spatter).

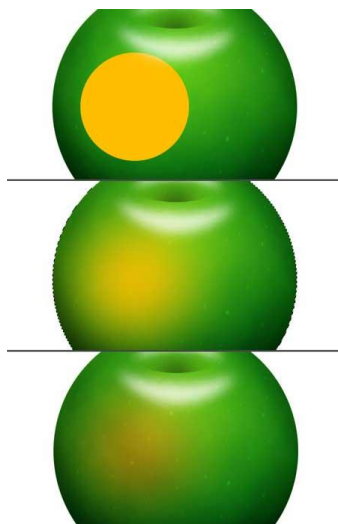
Радиус Разбрызгивания (Spray Radius) устанавливаем 6, а Смягчение (Smoothness) - 14. Нажимаем ОК.

**Ctrl-клик** по иконке слоя "Apple" чтоб выделить яблоко, и переходим **Выделение > Инверсия** (Select > Inverse) чтоб инвертировать выделение, нажимаем **Delete** и очищаем выделенную область. **Режим наложения** слоя меняем на **Осветление** (Screen). Снимаем выделение.




**Шаг 10.** Создаем новый слой выше слоя "Dots" и называем его "**Red**." Выбираем инструмент **Овальное выделение**  (Elliptical Marquee Tool) и создаем выделение, как на рисунке ниже, и заполняем (Fill) его цветом **255 190 0**.

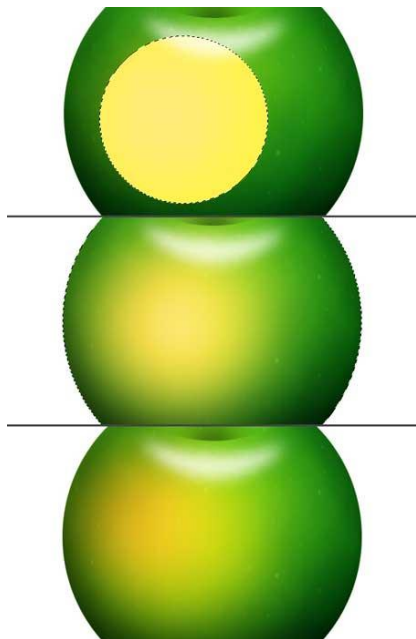
**Ctrl-click** по иконке слоя "Apple", чтоб выделить яблоко. Переходим в **Фильтр > Размытие > Размытие по Гауссу** (Filter > Blur > Gaussian Blur) и устанавливаем радиус 50 пикселей. **Режим наложения** меняем на **Цветовой тон** (Hue). Снимаем выделение (Ctrl + D).






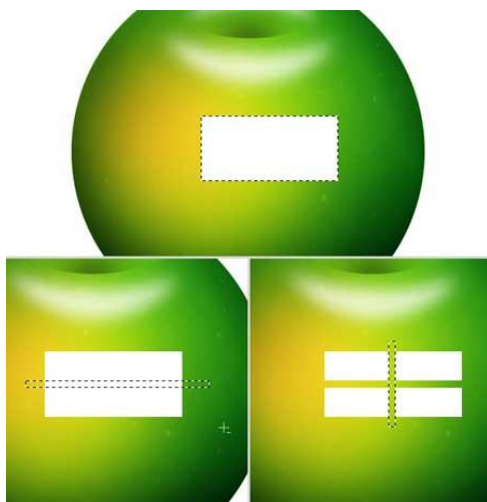
**Шаг 11.** Создаём новый слой под слоем "Red", называем его "Yellow." Выбираем **Овальное выделение**  (Elliptical Marquee Tool) и создаем выделение как показано на рисунке, заполняем его цветом **255 244 68**.


**Ctrl-клик** по слою "Apple", чтоб загрузить выделение. Переходим в **Фильтр > Размытие > Размытие по Гауссу** (Filter > Blur > Gaussian Blur), радиус также устанавливаем 50 pixels. **Режим наложения** устанавливаем на **Жесткий Свет** (Hard Light) и **Непрозрачность** (Opacity) - 75%. Снимаем выделение.




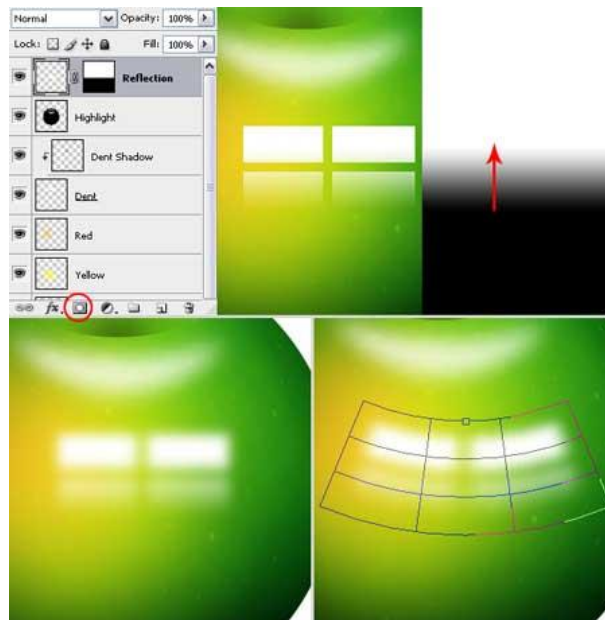
**Шаг 12.** Создаём новый слой и называем его "Reflection." Выбираем инструмент

**Прямоугольная область**  (Rectangular Marquee Tool) и создаём выделение как на рисунке ниже, заливаем белым цветом. Мы сделаем отражение окна из этого прямоугольника. Создайте две прямоугольных области как на рисунке ниже и очистите их. Затем очищаем чтоб получить четыре одинаковых прямоугольника белого цвета



**Шаг 13.** Переходим к палитре слоёв и добавляем **Маску слоя** (Layer Mask) к слою "Reflection" нажатием по кнопке  «Добавить маску слоя».

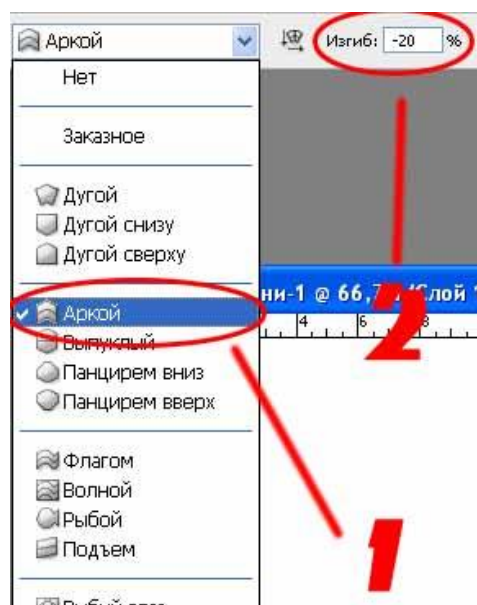
Выбираем инструмент **Градиент**  (Gradient Tool) типа Linear Залейте маску слоя градиентом как показано на рисунке ниже.



Сейчас у нас активна Маска слоя, щелкните по миниатюре слоя "Reflection", чтоб сделать активным сам слой (**Выберите слой а не маску Reflection**)


Теперь переходим в **Фильтр > Размытие > Размытие по Гауссу** (Filter > Blur > Gaussian Blur), радиус установите 8 пикселей.

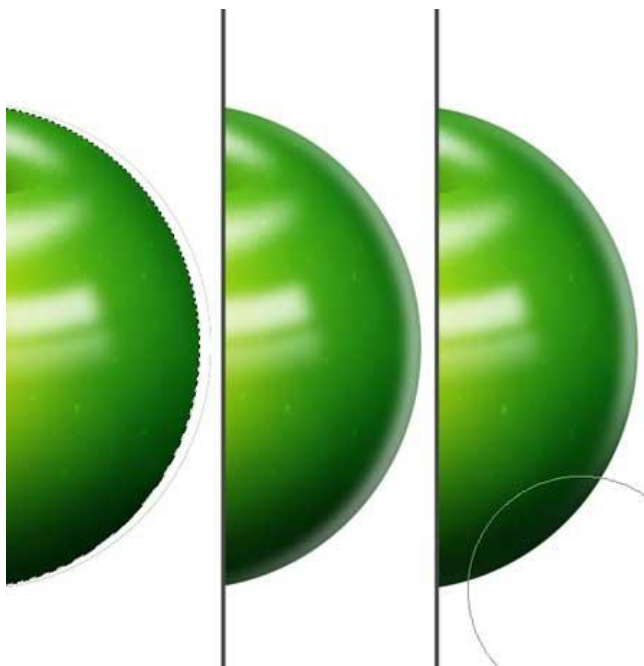
Переходим **Редактирование > Трансформация > Деформация** (Edit > Transform > Warp), Деформация Аркой (Arc) и устанавливаем Изгиб (Bend) на -20%. После чего нажимаем галочку в верхней правой части меню.



**Шаг 14.** Создаем новый слой под названием "**Backlight.**" **Ctrl-клик** по слою "Apple" и заливаем выделенную область белым цветом. Проверьте, что у вас активен инструмент выделения, переместите выделенную область примерно на 10 пикселей влево с помощью клавиши Левая Стрелка на вашей клавиатуре, нажмите **Delete**, чтоб очистить выделенную область.

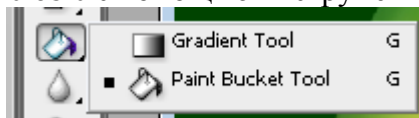
**Ctrl-клик** по слою "Apple", чтоб создать выделенную область, переходим **Фильтр > Размытие > Размытие по Гауссу** (Filter > Blur > Gaussian Blur), радиус устанавливаем 7

пикселей. Нажимаем **Ctrl + D**, чтоб снять выделение. Выбираем инструмент **Ластик**  (Eraser Tool), Диаметр (Master Diameter) устанавливаем 270 пикселей, и Жесткость (Hardness) на 0%. Сотрите нижнюю часть свечения как показано на рисунке.



**Шаги с созданием веточки пропущены**

**Шаг 18.** Теперь, прежде чем слить все слои мы должны сделать последние штрихи. Заливаем фоновый слой цветом **0 68 0** с помощью инструмента **Paint Bucket Tool**

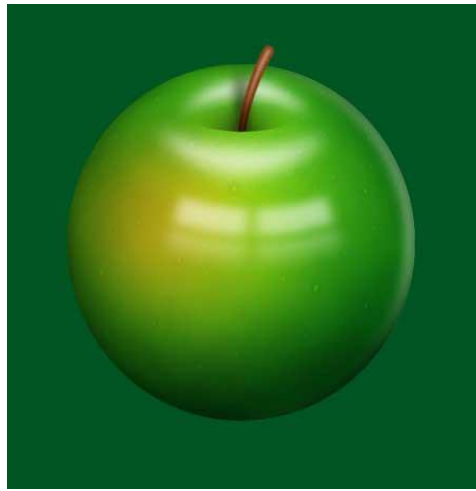



Изменим **непрозрачность слоёв (Opacity)**: слой "Highlight" - 85%, "Yellow" - 35%, "Red" - 85%, "Dent Shadow" - 95%, "Backlight" - 60%, "Reflection" - 55%.

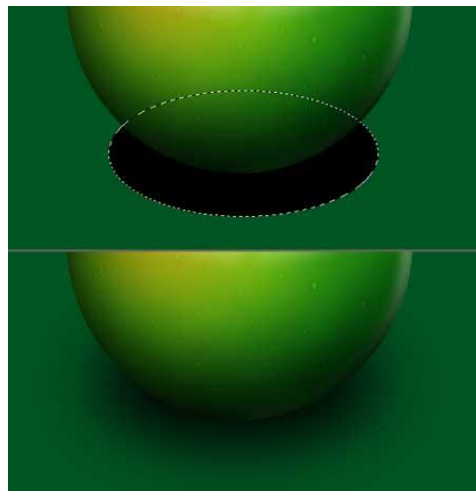
Выберите слой **Apple** и примените к нему **стиль Внутреннее свечение** (Layer > Layer Style > Blending Options > Inner Glow) со следующими настройками: Режим (Blend Mode) - Умножение (Multiply), Непрозрачность (Opacity) - 30%, Размер (Size) - 20 pixels, и Цвет (Color) - 0 51 0


Применим **стиль Внешнее свечение** (Outer Glow) к слою "Dots" с такими настройками: Режим (Blend Mode) - Умножение (Multiply), Непрозрачность (Opacity) - 60%, Цвет (Color) - 43 43 43, и Размер (Size) - 8 pixels.

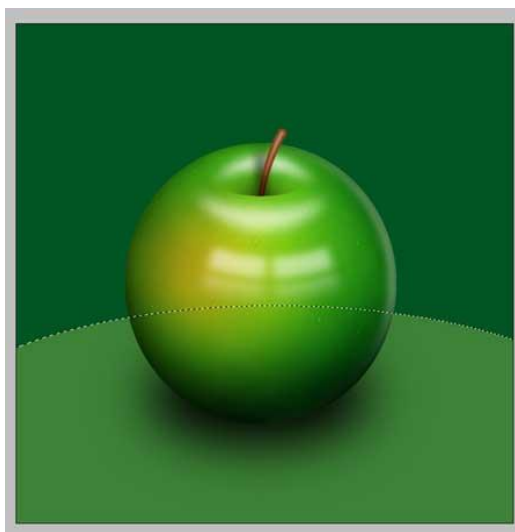
Теперь выделяем все слои кроме фона, и переходим Слои > Новый > Группа из слоёв (Layer > New > Group From Layers), и называем её "Apple."



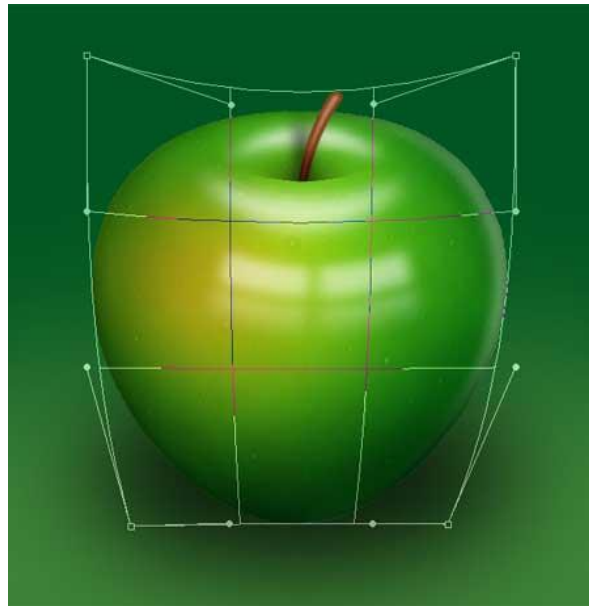
**Шаг 19.** Создаём новый слой под группой слоёв "Apple" и называем его "**Shadow**." Выбираем инструмент **Овальное выделение**  (Elliptical Marquee Tool) и создаём выделение, заливаем его чёрным цветом. Снимаем выделение и переходим **Фильтр > Размытие > Размытие по Гауссу** (Filter > Blur > Gaussian Blur), устанавливаем Радиус 40 пикселей и применяем.



**Шаг 20.** Создаём новый слой над слоем "Background" и называем его "**Light**." Выбираем инструмент **Овальное выделение**  (Elliptical Marquee Tool) и создаём выделение, как на картинке ниже, заливаем его цветом **51 136 51**. Ctrl + D, чтоб снять выделение, теперь переходим в **Фильтр > Размытие > Размытие по Гауссу** (Filter > Blur > Gaussian Blur), и применяем его с радиусом 75 пикселей.



**Шаг 21.** Теперь у нас есть своё готовое яблоко, но оно слишком круглое. С помощью деформации придадим ему нужную форму. Клик левой кнопкой мыши по группе "Apple", и нажмите Ctrl+E чтоб объединить все слои. **Переходим Редактирование > Трансформирование > Деформация** (Edit > Transform > Warp) и придаём нашему яблоку нужную форму. Затем применяем трансформацию нажав галочку в верхнем левом углу меню



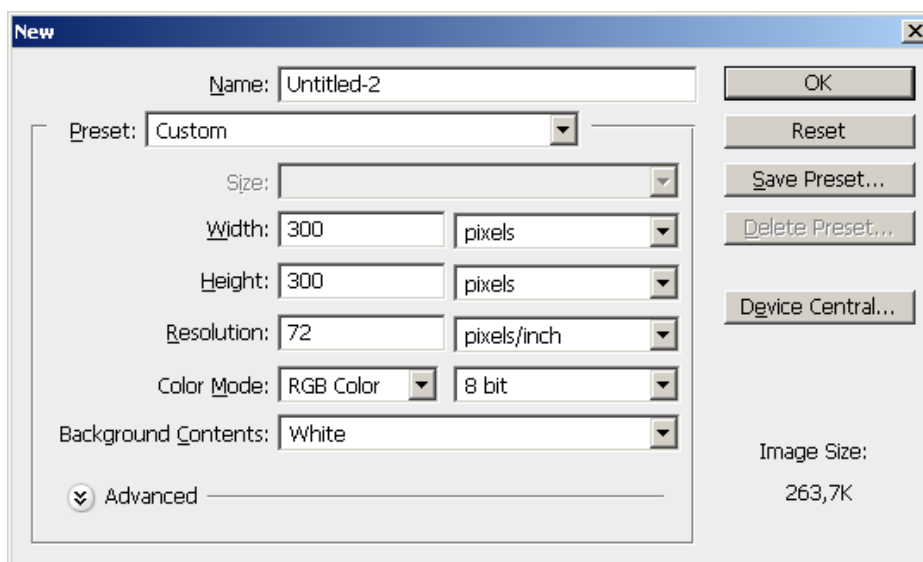
## Практическая работа №31 (ПР31)

**Тема:** Растровый графический редактор Photoshop

**Цель:** Создание стильного логотипа.

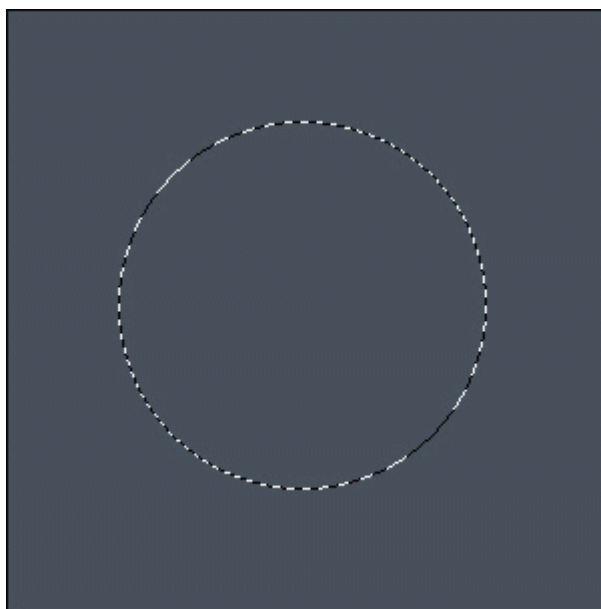
### Шаг 1

Открываем Photoshop и создаём изображение 300x300 пикселей (**File >New**). Цвет заднего фона не имеет значения.



### Шаг 2

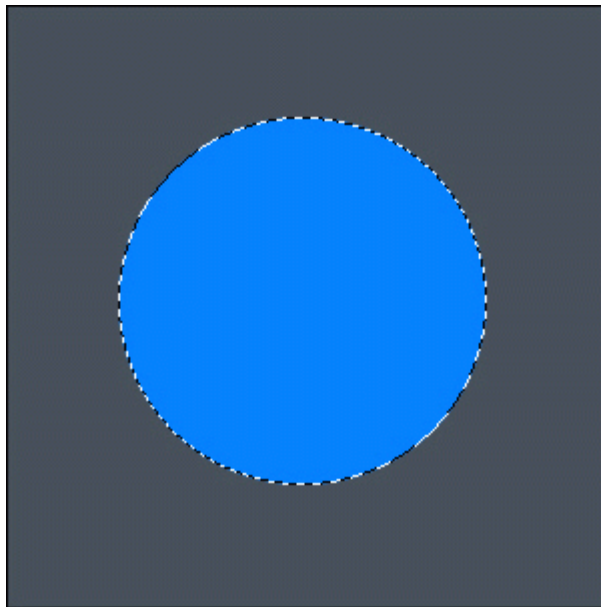
Создаём новый слой (**Shift+Ctrl+N**) и выбираем инструмент выделения **Elliptical Marquee Tool**. Зажимаем **Shift** для создания идеального круга, и рисуем.



Если необходимо размещаем круг по центру.

Если не получилось сделайте «Шаг назад» (**Ctrl+Alt+Z**) и попробуйте снова.

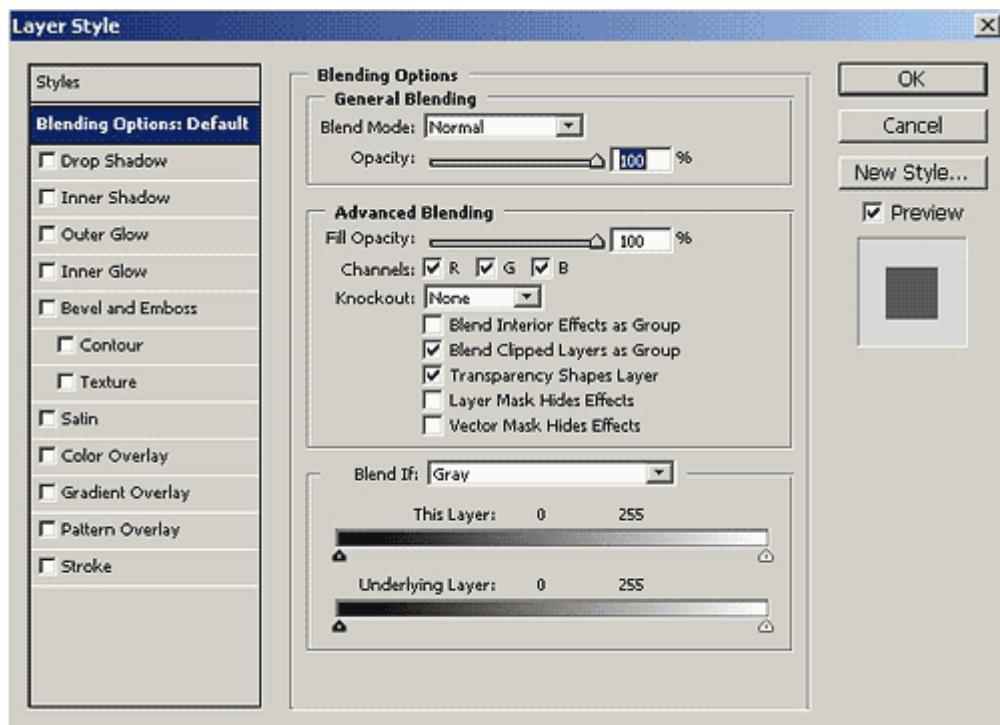
Нажмите по выделению круга правой кнопкой мыши и выберите пункт **Fill** («Заполнить») Во всплывающем пункте **Use:** выберите пункт **Color**. В открывшемся окне выберите примерно синий цвет как на рисунке ниже и нажмите **OK**. Еще раз нажмите **OK**



Нажимаем **Ctrl+D** для снятия выделения.

### Шаг 3

Выбираем **Layer > Layer Style > Blending Options**. Устанавливаем все параметры как на скриншотах:



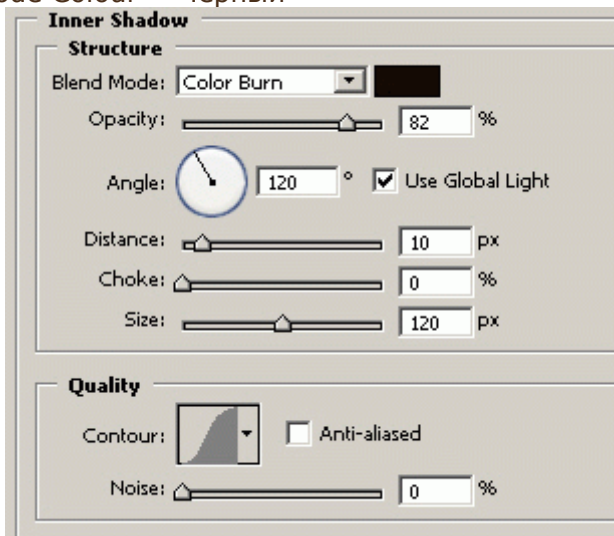
**Drop Shadow** Blend Mode Colour = Черный

Обратите внимание на параметр **Contour** во всех вкладках. **Необходимо задавать графиком определенный изгиб**

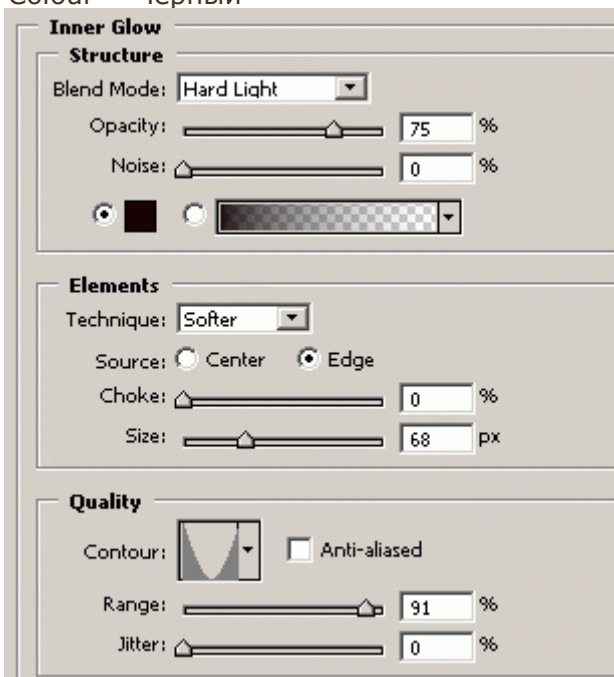




**Inner Shadow** Blend Mode Colour = Черный

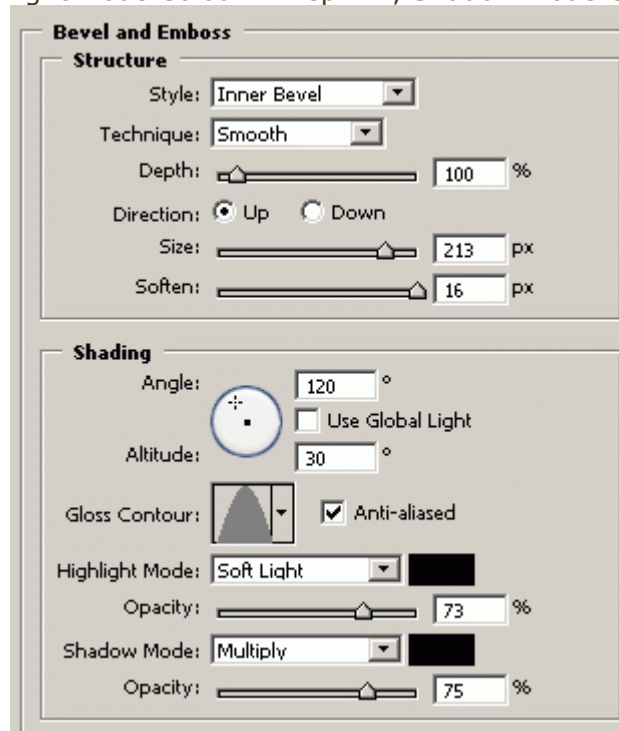


**Inner Glow** Blend Mode Colour = Черный



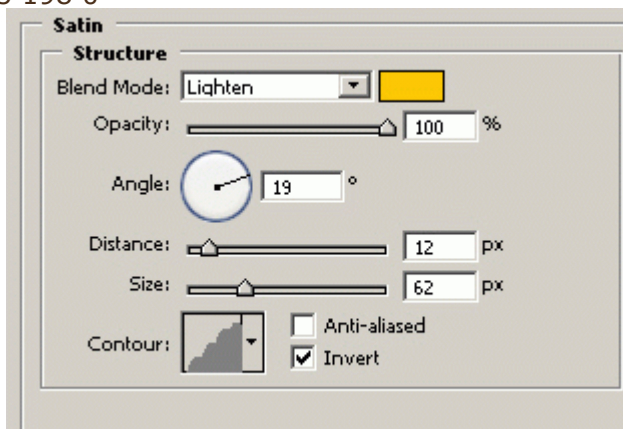


**Bevel and Emboss** Highlight Mode Colour = Черный, Shadow Mode Colour = Черный

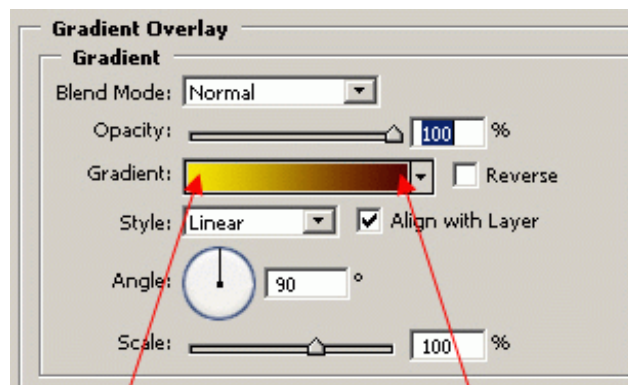


### Satin

Blend Mode Colour = 255 198 0



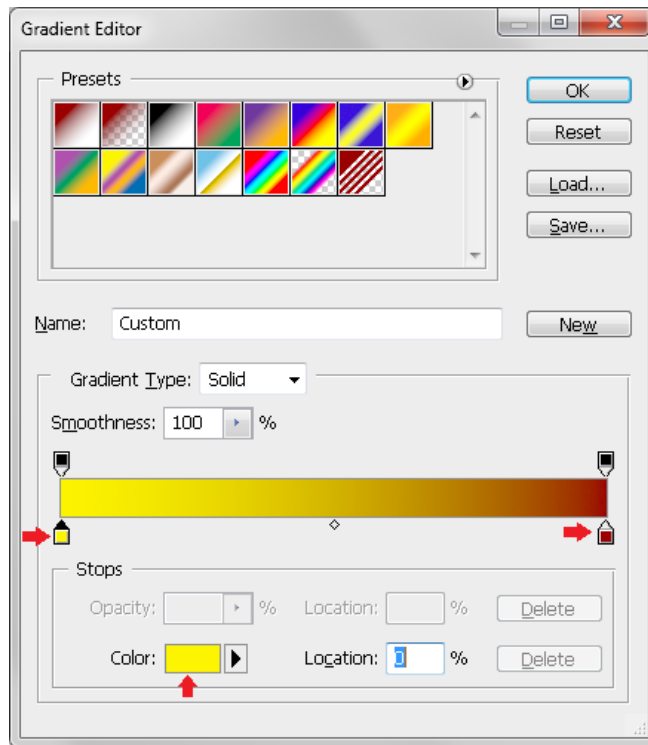
### Gradient Overlay



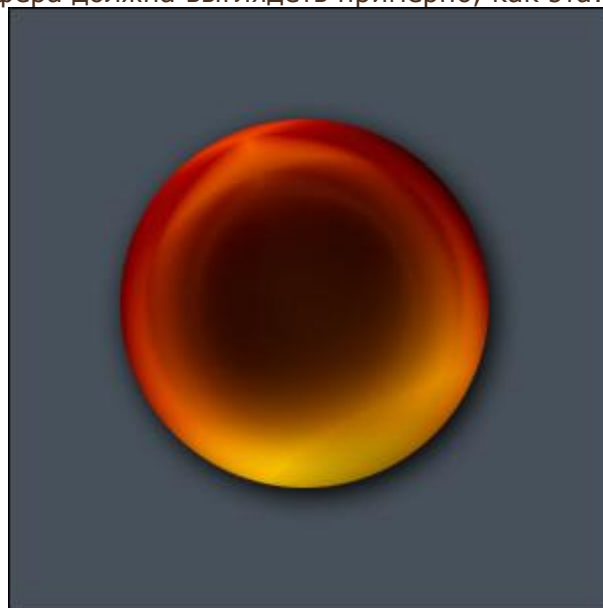
249 230 0

83 0 0

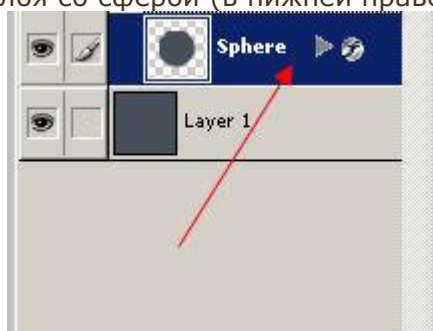
Нажмите по полосе цвета **Gradient**. Затем выделите левый нижний ползунок. В разделе **Stops** должен быть доступен пункт **Color**, нажмите по цветному прямоугольнику и впишите в нижнюю часть **R G B** код первого цвета (249 230 0) для градиента. Затем выделите правый нижний ползунок и выберите для него другой цвет (83 0 0), указанный на рисунке выше.



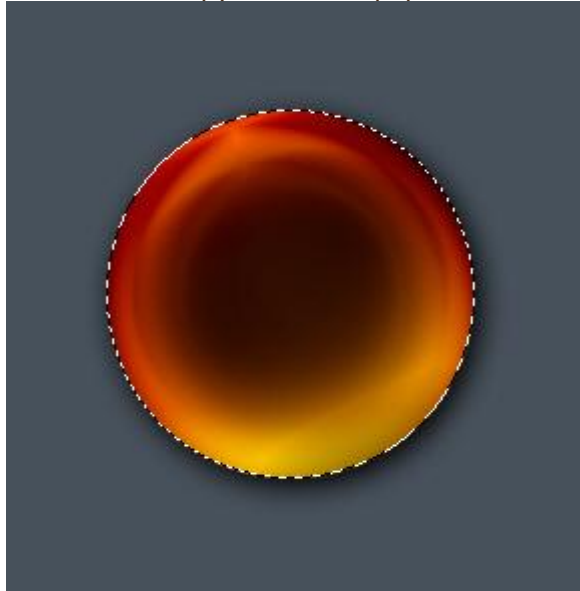
Нажимаем ОК, и наша сфера должна выглядеть примерно, как эта:



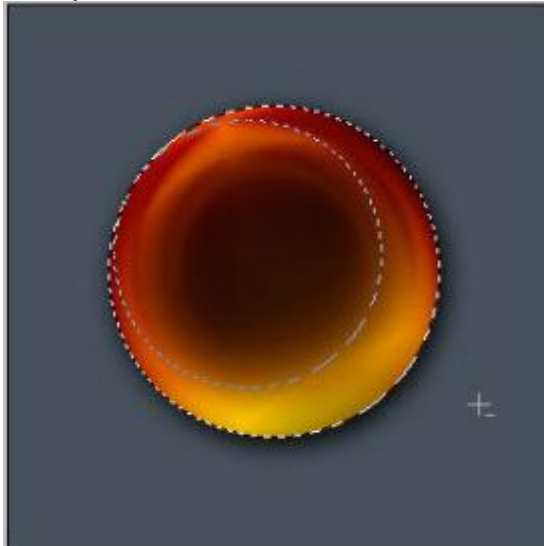
**Шаг 4** Этот шаг научит рисовать объект на вершине сферы. Зажимаем Ctrl и щелкаем мышкой по картинке нашего слоя со сферой (в нижней правой части окна программы)



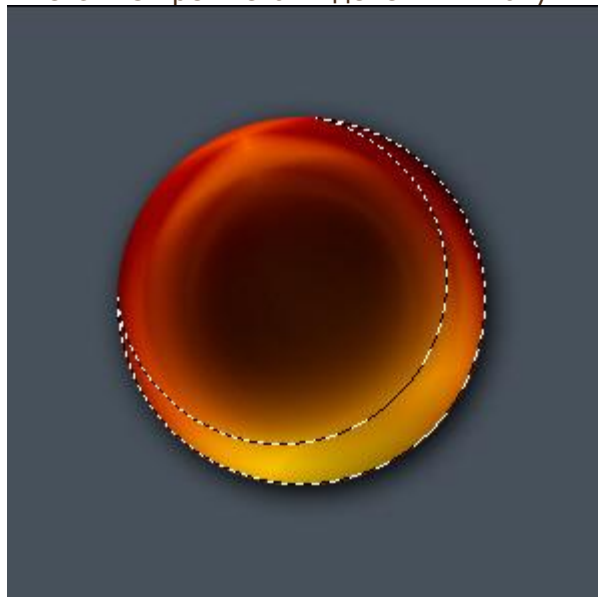
у Вас должно появиться выделение вокруг вашей сферы:



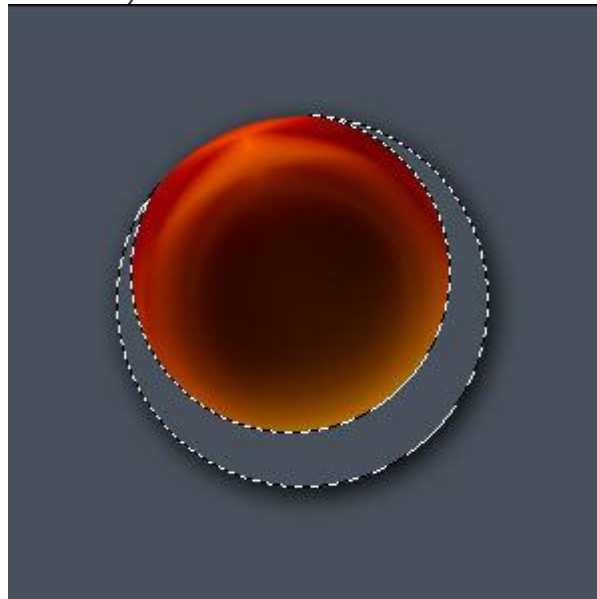
Выберите **Elliptical marquee tool**, зажмите Alt и рисуйте круг внутри выделения. Убедитесь, что эти два круга пересекаются в одной точке как на рисунке:



Нарисованная область вычтется из прежнего выделения и получится такая фигура:



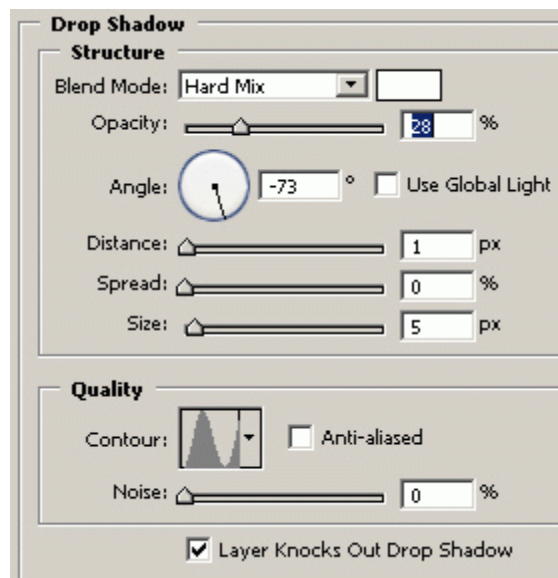
Создаём новый слой (**Shift+Ctrl+N**) поверх слоя со сферой. Заливаем наше выделение через пункт Fill ( Цвет = 71 81 91)



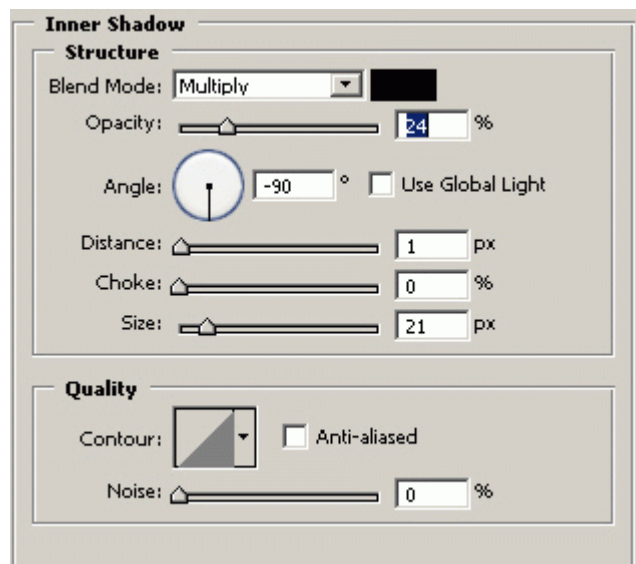
После этого снимем выделение **Ctrl+D**.

**Шаг 5** В этом шаге мы будем создавать текстуру для получившегося выделения. В верхнем меню выбираем **Layer > Layer Style > Blending Options**.

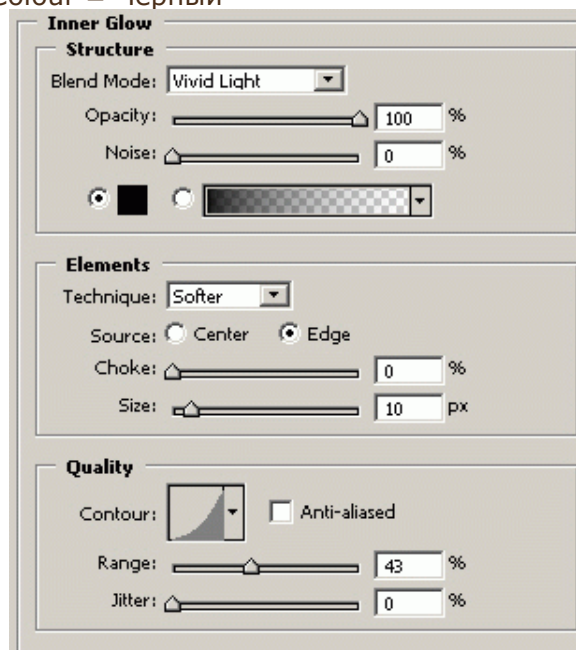
Выбираем **Drop Shadow**. Устанавливаем цвет Blend Mode = белый, и всё остальное:



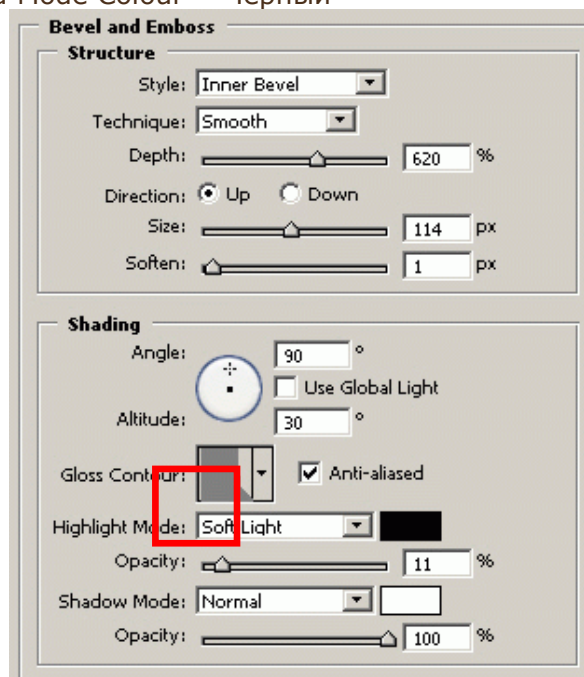
**Inner Shadow** Blend Mode Colour = Черный



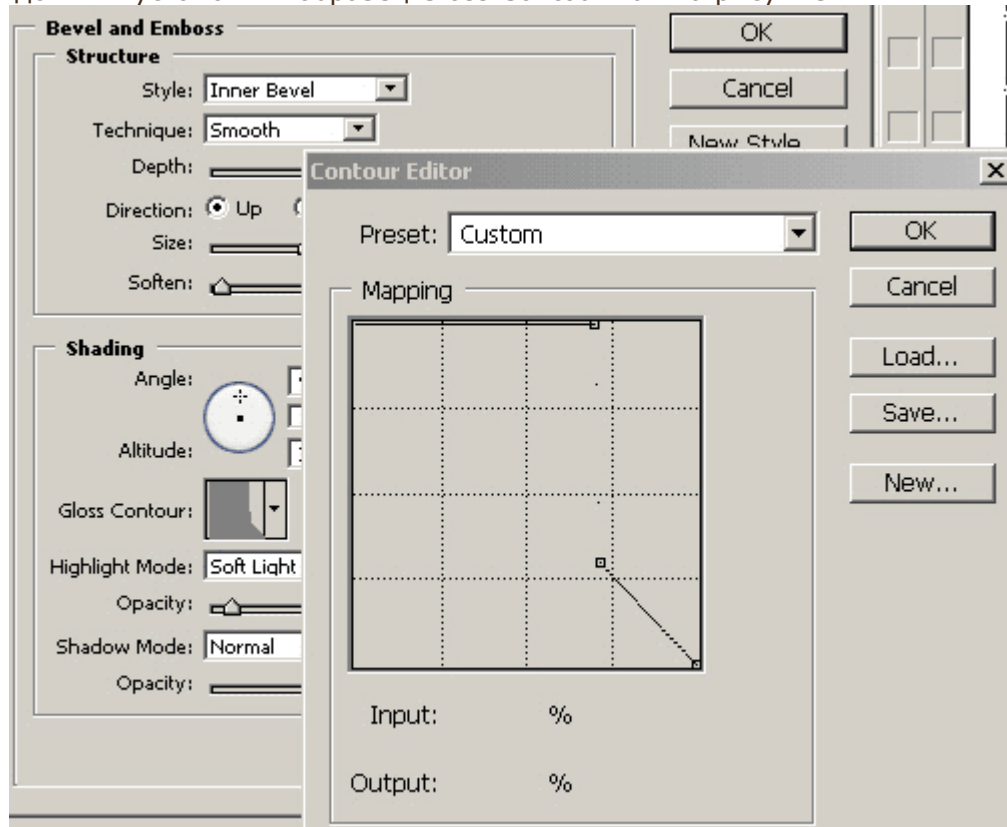
**Inner Glow** Blend Mode Colour = Черный



**Bevel and Emboss** Blend Mode Colour = Черный

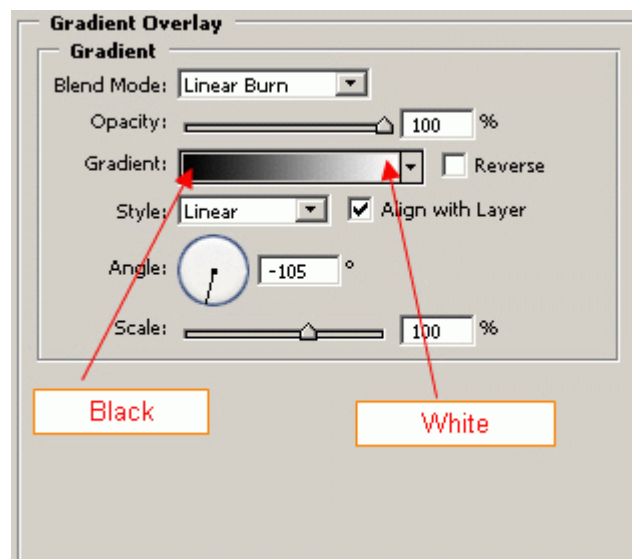


Теперь мы должны установить образец Gloss Contour как на рисунке:

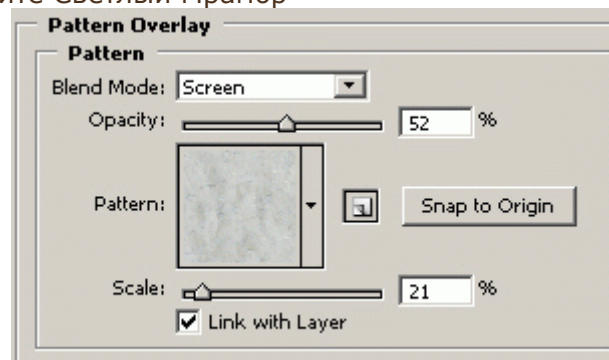


Чтобы выпрямить последний отрезок, установите включенным флажок Corner

### Gradient Overlay



**Pattern Overlay** Выберите Светлый Мрамор





Подтверждаем настройки, нажимая ОК. Теперь у Вас должна быть сфера, примерно, как на рисунке:



### Шаг 6

Текст. Для этого лого был использован шрифт **Space Age** размером 18px, но вы можете использовать любой понравившийся вам (**BankGothic Lt BT**). Затем, на отдельном слое просто напечатайте несколько точек с пробелами между ними. В итоге получилось это:

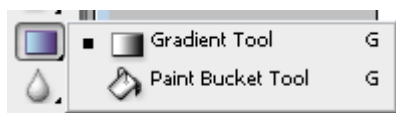


### Шаг 7

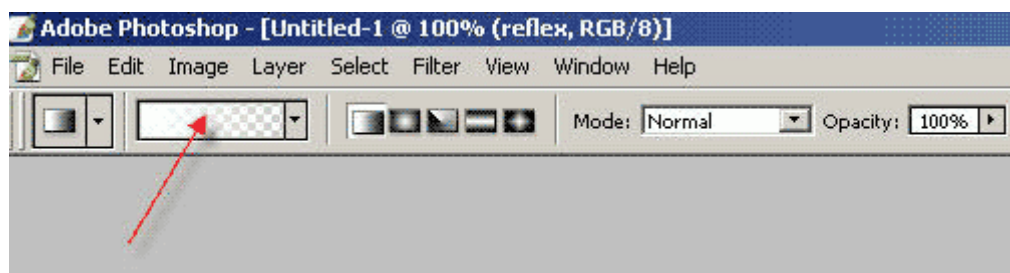
Завершающий шаг. Создадим логотипу лёгкое отражение для более стильного вида. Создаём новый слой. Выбираем **Elliptical Marquee Tool** и рисуем овал:



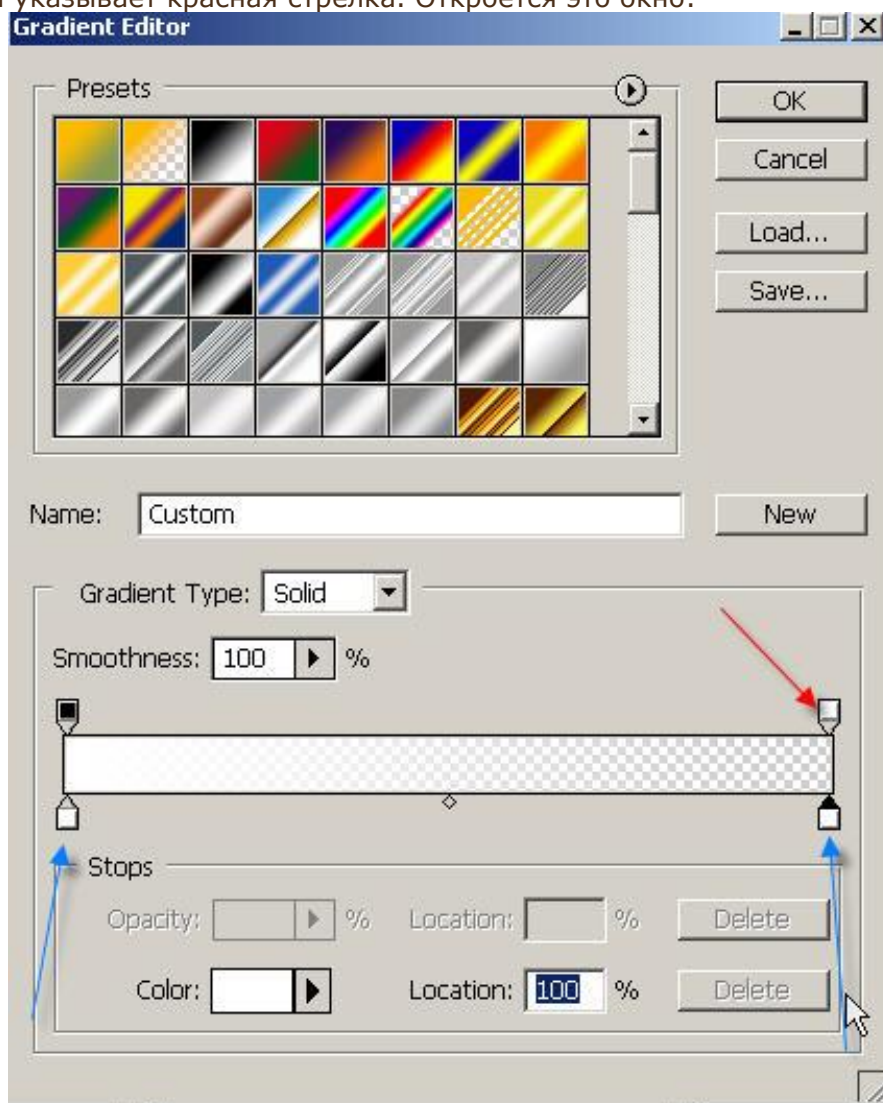
Выбираем **Gradient tool**.



Сверху вы должны видеть этот инструмент:



Нажмите, куда указывает красная стрелка. Откроется это окно:



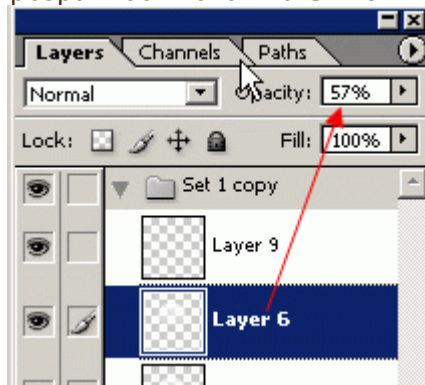
Кликните на каждом маленьком ползунке, на который указывает синяя стрелка и выставьте белый цвет. Затем кликните на ползунок, на который указывает красная стрелка и выставьте прозрачность (Opacity) на 0. Нажмите OK.

Инструментом **Gradient tool** (кликаем и растягиваем вертикально) вы можете нарисовать стеклянное отражение на свой вкус:

В случае если хотите отменить одно действие нажмите (Ctrl+Alt+Z)



Теперь установите уровень непрозрачности слоя на **57%**



Теперь создаём новый слой и рисуем другой овал с помощью инструмента выделения (**Elliptical Marquee Tool**). Но в этот раз сделайте его меньше:



И повторяем шаг с нанесением градиента. В итоге получаем:

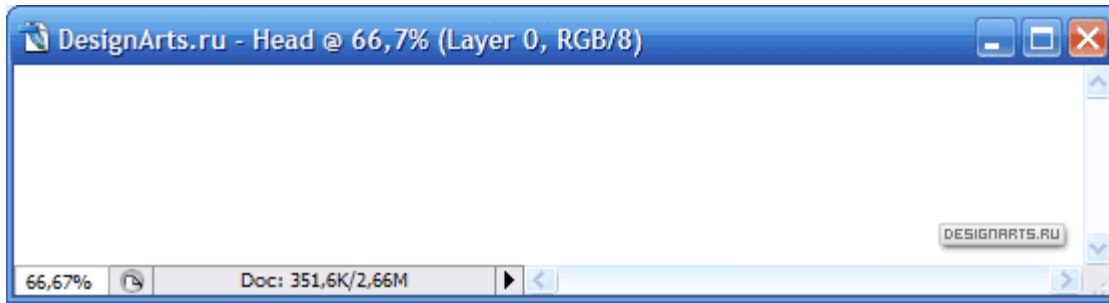


## Практическая работа №32 (ПР32)

**Тема:** Растровый графический редактор Photoshop

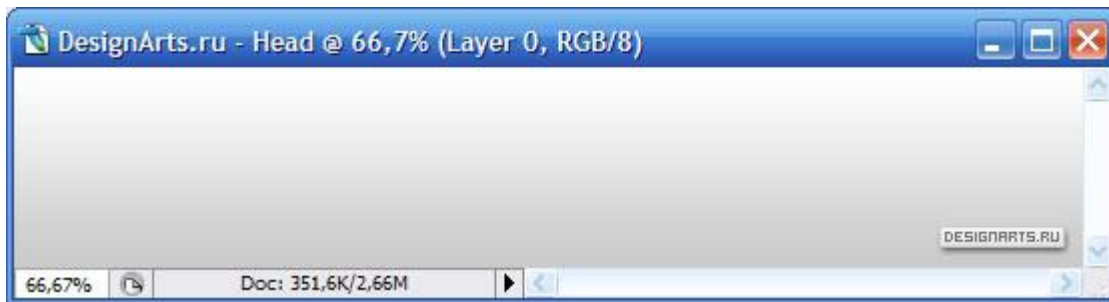
**Цель:** Создание «шапки» для сайта

1. Создаём новый рисунок размером 800x150 px:



2. Создаем новый слой. Используя Gradient Tool задайте Linear градиент, где первый цвет 203 201 204, а второй – 255 255 255. Проводим градиент вертикально.

Получаем фон:



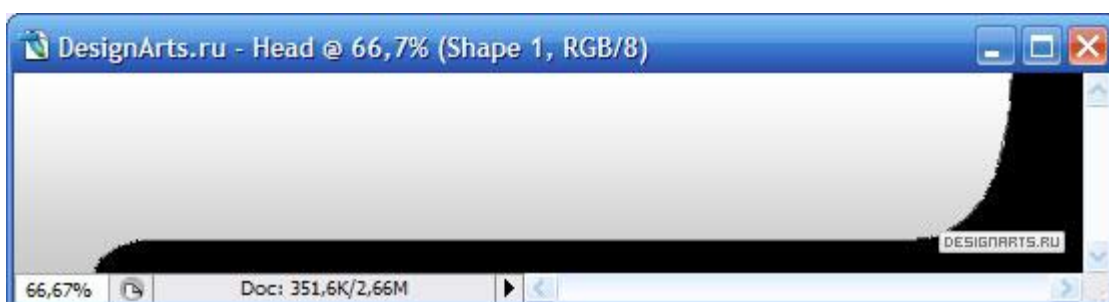
3. Выбираем как основной цвет 49 62 90. За тем выбираем инструмент **Freedom Pen Tool** и рисуем такую фигуру.



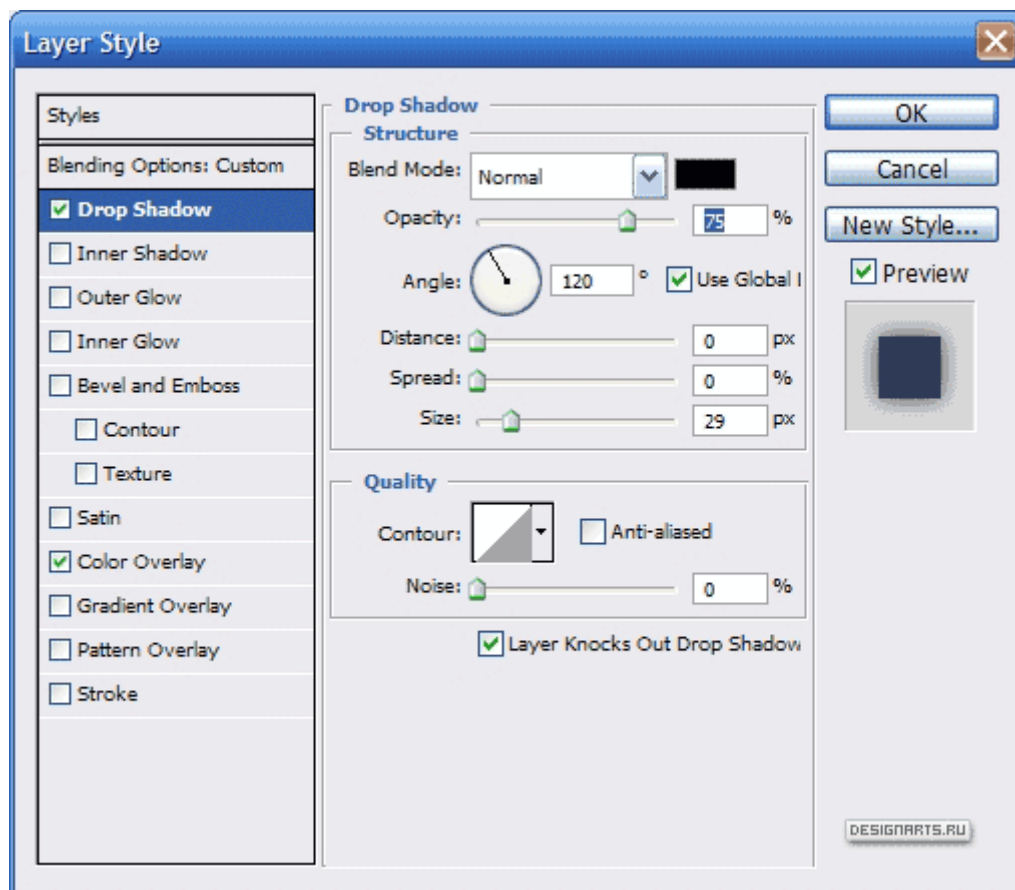
Выберите стиралку (Eraser Tool) и начните стирать. В предложенном диалоге (Растрезировать изображение) нажмите «Да»

Стираем примерно, как на рисунке ниже. Также в качестве стирания используйте прямоугольное (Rectangular Marquee Tool) и круглое (Elliptical Marquee Tool) выделения и нажимайте Delete.

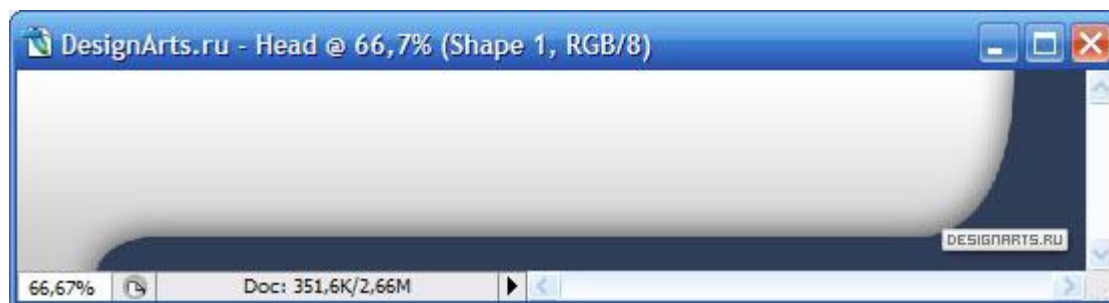
Также если вы стерли лишнее воспользуйтесь кисточкой (Brush Tool) или сделайте шаг назад (Ctrl +Alt + Z)



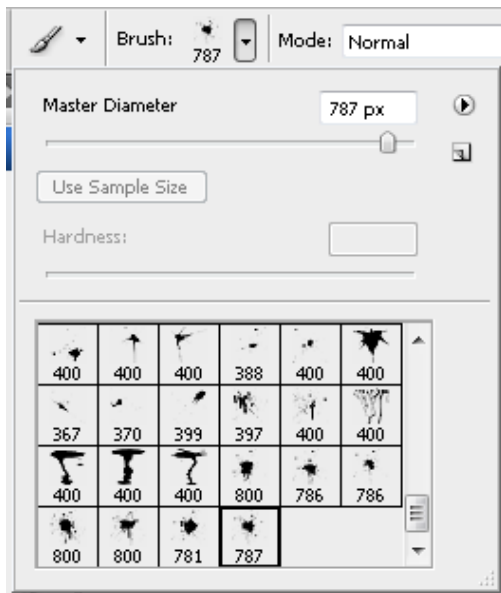
4. Заходим в **Layer > Layer Style > Blending Options** на вкладку **Drop Shadow** и выставляем следующие параметры:



После этого должно получиться примерно как у меня:



5. Создайте новый слой. Теперь выберите в качестве основного цвета (**49 62 90**) и кисточку с пятнами краски (Чтобы добавить новые кисточки, откройте файл **splatteris\_m1.abr**, который находится в папке с практическим заданием)



Что-нибудь «наляпайте»:



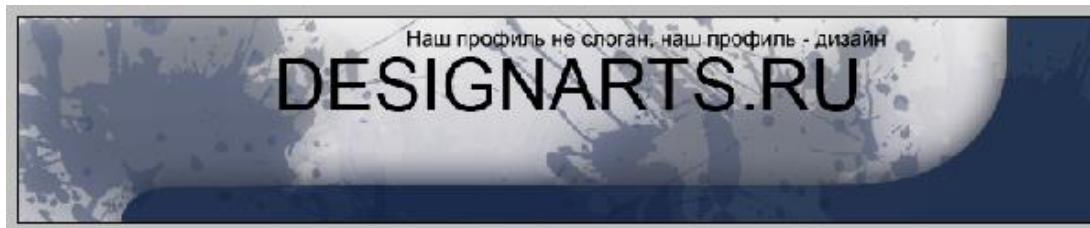
6. Снова создайте новый слой, смешивание (**Blend Mode**) измените с **Normal** на **Overlay** (данный параметр настраивается на панели слоёв **Layers**) и снова нарисуйте что-нибудь кисточкой:



7. А сейчас напишите название сайта (Arial 40pt):

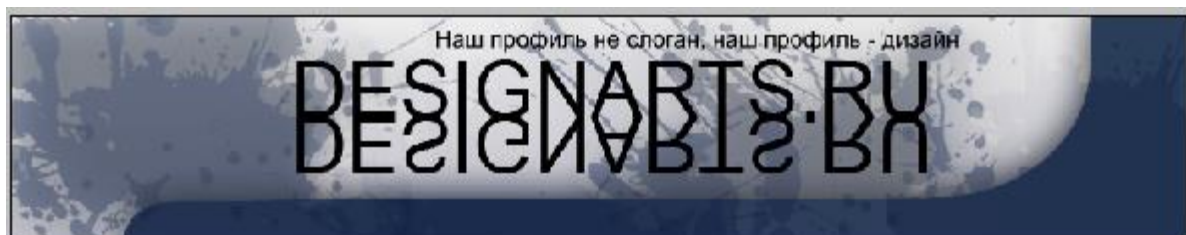


8. Нарисуйте ещё что-нибудь и напишите слоган вашего сайта (Arial 12):

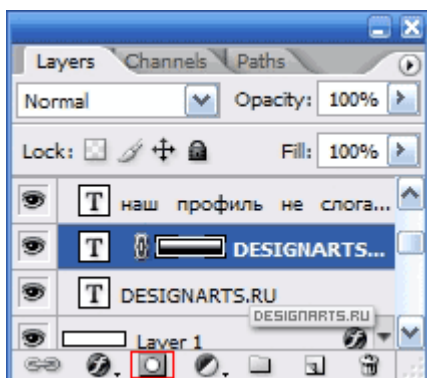


12. А сейчас создайте копию слоя с названием сайта (для этого кликните по слою правой клавишей и там выберите **Duplicate Layer**).

Отразите по вертикали копию слоя с названием сайта (**Edit** —> **Transform** —> **Flip Vertical**) и разместите её (надпись) ниже:

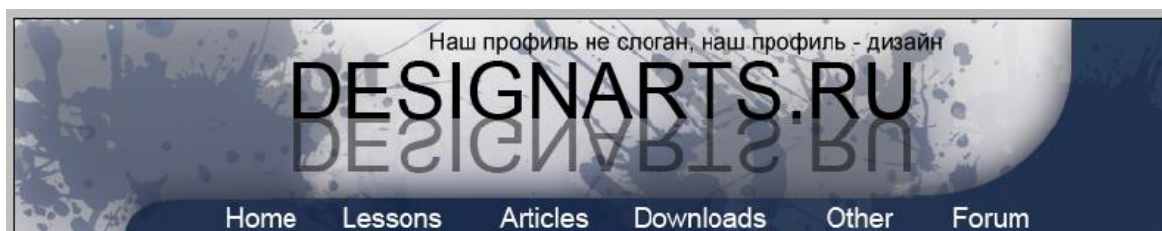


13. Добавьте маску слоя (смотрите на рисунок ниже):



14. Выберите инструмент **Gradient Tool** (градиент) и чёрно-белым градиентом залейте маску сверху вниз:

15. Снизу напишите названия основных разделов сайта (Arial 14pt, между разделами по 8 пробелов):



16. Создайте новый слой. С помощью инструмента **Rectangular Marque Tool** выделите верхнюю часть изображения и залейте её белым цветом:



17. Добавьте к данному слою маску, как это мы делали ранее, и залейте её чёрно-белым градиентом сверху вниз:



Вот мы и нарисовали в фотошопе шапку для сайта. Сохраните нашу шапку в JPG формате (File > Save As)

18. Создайте html-страницу и добавьте туда код для отображения шапки.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title> DesignArts.ru</title>
6 <style>
7 #header{
8 width: 800px;
9 height: 150px;
10 margin-bottom: 10px;
11 background-image: url(Shapka.jpg)
12 }
13 </style>
14 </head>
15 <body>
16
17 <div id="header"></div>
18
19 </body>
20 </html>
```

## Практическая работа №33 (ПР33)

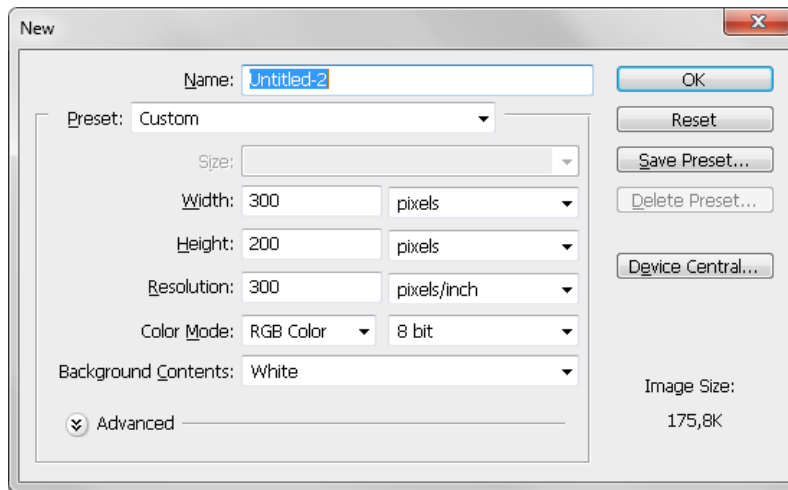
**Тема:** Растровый графический редактор Photoshop

**Цель:** Создание реалистичных стеклянных (или пластиковых) кнопок для использования в веб-дизайне

### Выполнение работы:

Начните с создания нового документа, размер 300 на 200 px, цвет фона – белый.





## 1. Форма пилюли



1.1. Нарисовать форму пилюли очень просто – вам нужно только скомбинировать два круга и прямоугольник.

Создайте новый слот с именем **Shadow**. Выберите инструмент Elliptical Marquee и в верхнем меню в параметре **Style / Стиль** установите значение **Fixed Size / Задан.размер**. Ширину и высоту задайте 75 x 75.

Нарисуйте круг слева по центру и залейте (Fill) его черным цветом.

Возьмите инструмент выделения Rectangle Marquee и создайте прямоугольное выделение со стилем Fixed Size и размером 230 x 75, соединяющее круги как на рисунке выше.

Нарисуйте и залейте (Fill) оставшийся круг справа.

Снимите выделение (Ctrl+D).

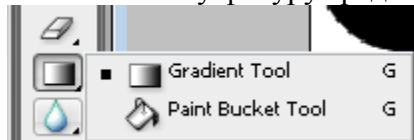


## 2. Стекланный эффект

2.1. Дублируйте слой **Shadow / Тень** (Правая кнопка мыши по слою > Duplicate Layer). Дайте имя копии Glass Base. Заблокируйте прозрачность слоя (кнопка Lock слева сверху над списком слоев),



и залейте вашу фигуру градиентом (Gradient Tool)



типа **Reflected / Зеркальный**




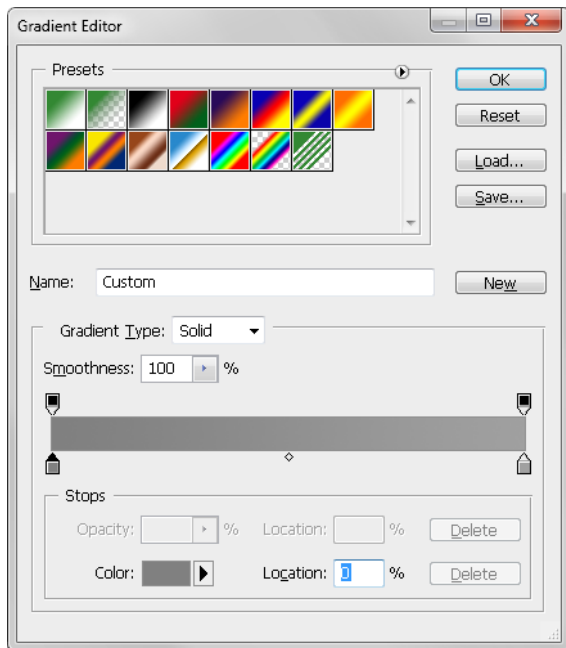
со следующими настройками

- первый цвет R=128, G=128, B=128

- второй цвет R=160, G=160, B=160

Mode: Normal


Чтобы установить цвет нажмите по . В появившемся окне нажмите двойным кликом нижний левый ползунок чтобы установить первый цвет, и правый нижний ползунок чтобы установить второй цвет.



Начинаем градиент в центре фигуры и проводим мышкой до верхнего (или нижнего) края. Центр фигуры должен окраситься в основной цвет, а края — в более светлый. Чтобы провести линию градиента строго вертикально, держите нажатой клавишу Shift.

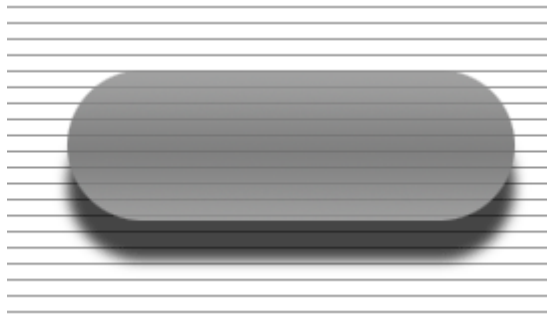


Прежде, чем продолжать, необходимо доделать слой **Shadow / Тень**. Выберите слой Shadow и примените к нему фильтр **Blur / Размытие > Gaussian Blur / Размытие по Гауссу** с радиусом 3 px. Измените его режим наложения на **Multiply / Умножение** и понизьте непрозрачность до 75%.

И наконец, сдвиньте слой вниз на 20 пикселей. Для этого выберите инструмент (Move Tool (V) ) и держа клавишу Shift нажмите два раза на кнопку со стрелкой вниз.

2.2. Сделаем выделение, щелкнув с зажатой клавишей Ctrl по слою Glass Base. Теперь продублируем слой **Shadow**. Назовите этот слой Faded BG и переместите его поверх слоя Glass Base. Поменяйте режим наложения на **Multiply / Умножение** и понизьте непрозрачность до 25%. Режим **Multiply / Умножение** делает видимыми темные линии, не затрагивая светлые. А понижение непрозрачности дает эффект видимости через стекло. Если вы хотите, чтобы стекло выглядело более плотным, мутным — понизьте непрозрачность еще больше.

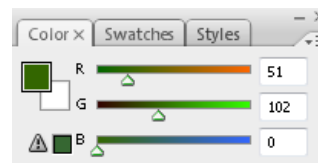




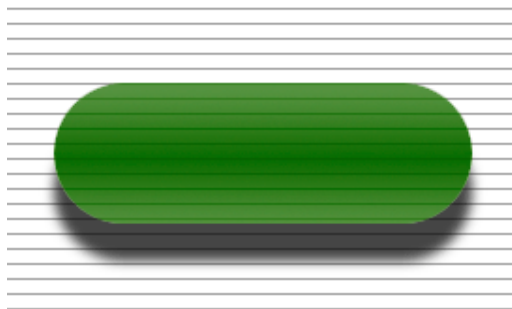
2.3. Прежде, чем перейти к работе над бликами и отражениями, придайте вашей кнопке цвет. Слой, который мы сейчас создадим, будет единственным слоем с информацией о цвете кнопки, поэтому, если вы захотите изменить цвет, вы будете обращаться к этому слою.

2.4. Продублируйте слой Faded BG, назовите слой Button Color.

Поменяйте основной цвет на R=51 G=102 B=0.




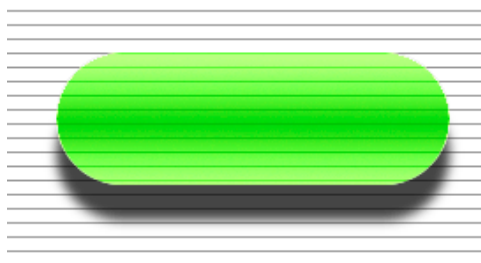
Если вы уже сняли выделение, то снова щелкните по иконке слоя Glass Base с зажатой клавишей Ctrl. Нажмем **Alt+Delete** / **Удалить**, чтобы залить выделенную форму зеленым цветом. Поменяйте режим наложения на **Overlay** / **Перекрытие** и верните значение непрозрачности 100%.



### 3. Освещение

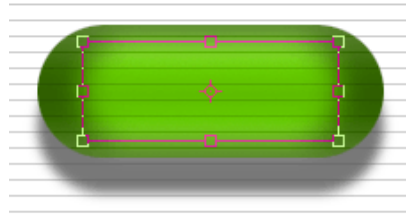
3.1. Продублируйте слой Button Color. Назовите новый слой Large Refraction. Нажмем клавишу D, а затем клавишу X, чтобы задать белый в качестве основного цвета.

Убедитесь, что включен переключатель Lock Transparent Pixels  и нажмите **Alt + Del**, чтобы залить фигуру белым цветом.

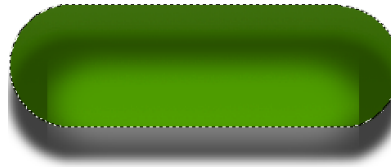


3.2. Теперь отключите Lock Transparent Pixels. Вам нужно размыть этот слой. Применяем **Gaussian Blur** / **Размытие по Гауссу** со значением 10 пикселей, и убедимся, что режим наложения слоя стоит **Overlay** / **Перекрытие**.

3.3. Выбираем в меню **Edit** / **Редактирование** > **Transform** / **Трансформирование** > **Scale** / **Масштабирование**, перетаскиваем рамку трансформации примерно как на рисунке

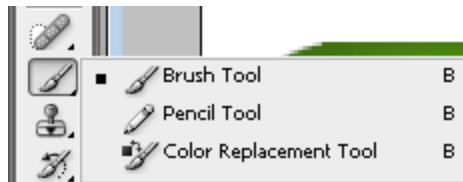


После этого возьмите верхний центральный узел рамки и перетащите его так, чтобы высота блика была равна примерно половине высоты основной фигуры. Нажмите Enter, чтобы подтвердить изменения.

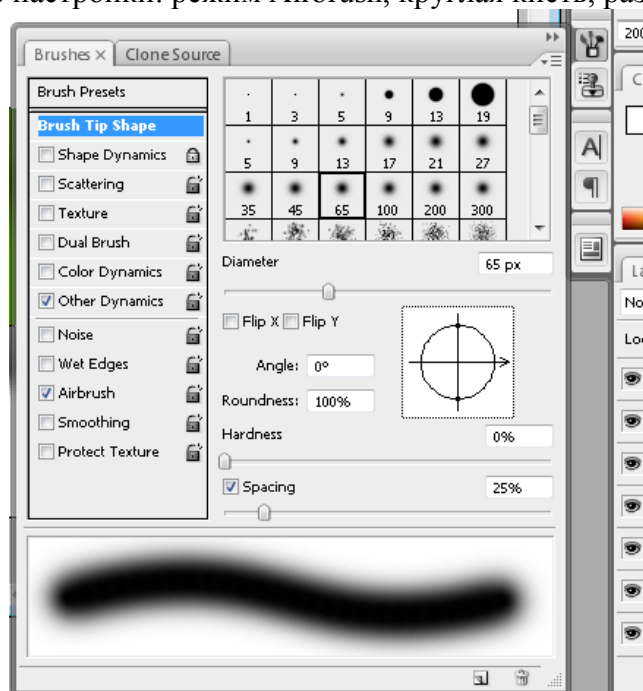


#### 4. Придаем объем: тени, блики и отражения

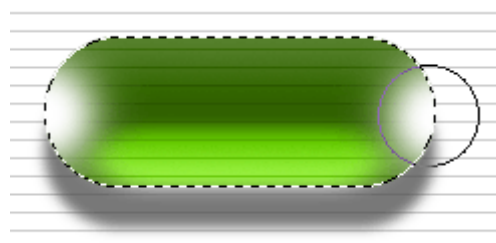
4.1. Снова выделите основную фигуру, щелкнув с зажатой клавишей Ctrl по слою Button Color, и создайте новый слой поверх всех предыдущих. Назовем новый слой Interior Highlights. Возьмем кисть.



Зададим следующие настройки: режим Airbrush, круглая кисть, размер 65, Hardness 0%.



Нарисуем белым цветом два блика, с левой и в правой стороны. Кисть должна лишь наполовину заходить на выделение. Должно получиться примерно, как на рисунке.



После этого снимите выделение (Ctrl + D).

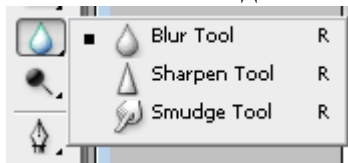
4.2. Возьмите инструмент выделения Rectangular Marquee Tool. Выделите левый блик. Возьмите инструмент Move Tool (V) и с зажатой клавишей Shift нажмите на клавиатуре стрелку вправо 2 раза (другими словами, передвинем блик на 20 пикселей вправо). То же самое сделайте с левым бликом, но двигайте, соответственно, влево.

4.3. Снимем выделение, поменяем режим наложение слоя на **Soft Light** / **Мягкий свет** и понизим непрозрачность до 50%

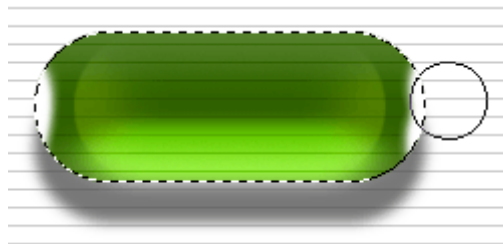
4.4. Снова сделаем выделение, нажав Ctrl и щелкнув по слою Button Color. Создадим новый слой с именем End Refractions, поместим его поверх предыдущих. Снова возьмем кисть в режиме Airbrush, выберем кисть с мягкими краями, диаметр примерно 45 пикселей, **Hardness** / **Жесткость**=50%.

4.5. Убедитесь, что основной цвет у вас по-прежнему белый, и сделайте два штриха кистью с левой и с правой стороны. Но будьте внимательны, штрихи не должны заходить на тот блик, который вы сделали в предыдущем шаге (слой Interior Highlights). Вы должны окрасить белым лишь самые края фигуры, как на рисунке.

Затем снимем выделение, возьмем инструмент **Blur** / **Размытие**.



Немного смягчим края, но не чересчур — только одним движением. Поменяем режим наложения слоя на **Overlay** / **Перекрытие**.



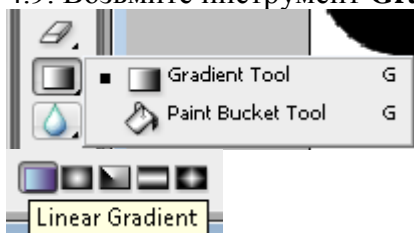
4.6. Опять сделайте выделение Button Color, создайте новый слой поверх других и назовите его Shadow Top. Нажимаем на клавиатуре M, затем 2 раза клавишу со стрелкой вниз. Заливаем выделение черным цветом. Затем, держа нажатой клавишу Shift, 1 раз нажимаем клавишу со стрелкой вниз. Теперь жмем **Delete** / **Удалить** и снимаем выделение. Поменяем режим наложения на **Soft Light** / **Мягкий свет** и понизим непрозрачность до 50%.

4.7. Продублируйте слой Shadow Top. Назовите новый слой Bottom Edge Hightlight. Нажимаем Ctrl +I, чтобы инвертировать цвета. Делаем выделение, нажав Ctrl и щелкнув по иконке слоя Button Color, выбираем в меню **Edit** / **Редактирование** > **Transform** / **Трансформирование** > **Flip Vertical** / **Отразить** по вертикали. Блик должен получиться как бы зеркальным отражением слоя Shadow Top.

Убедитесь, что задан режим наложения **Soft Light** / **Мягкий свет** и задайте непрозрачность 75%.

4.8. Снова сделайте выделение, нажав Ctrl и щелкнув по иконке слоя Button Color. Создайте новый слой поверх других слоев. Назовите этот слой Edge Refraction.

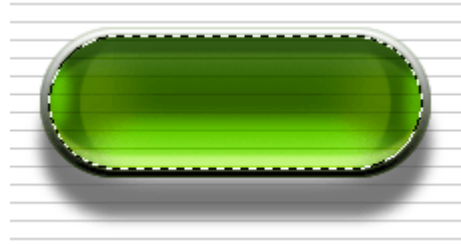
4.9. Возьмите инструмент **Gradient Tool** / **Градиент**.



В качестве первого цвета градиента выберите белый, а второго черный.

Применяем градиентную заливку к выделенной форме по направлению сверху вниз, сверху белый, к низу — черный.

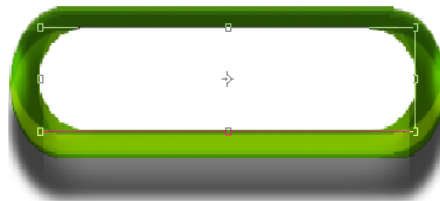
4.10. Выберите в меню **Select / Выделение > Modify / Модификация > Contract / Сжать**, уменьшите выделение на 4 пикселя. Нажимаем **Delete / Удалить**, чтобы удалить центральную часть, у нас должна остаться только обводка в 4 пикселя, окрашенная градиентом. Снимите выделение.



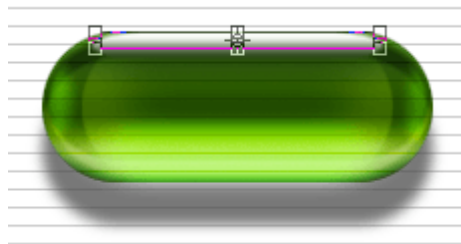
Меняем режим наложения слоя на **Overlay / Перекрытие**, понижаем непрозрачность до 50%.

4.11. Снова сделайте выделение, нажав Ctrl и щелкнув по иконке слоя Button Color. Создадим новый слой поверх других слоев. Назовем этот слой Main Highlight. Зальем выделение белым цветом, снимем выделение.

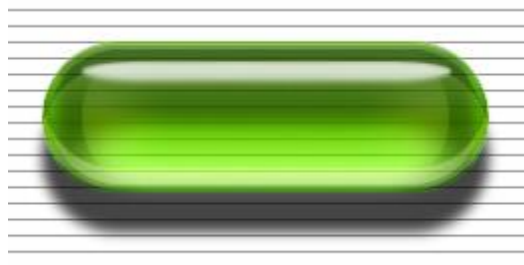
4.12. Выберите в меню **Edit / Редактирование > Transform / Трансформирование > Scale / Масштабирование**, трансформируем фигуру так, чтобы она примерно совпала со слоем Interior Highlights.



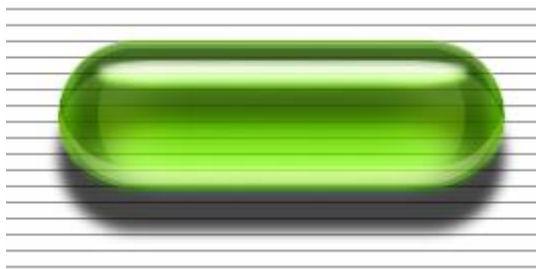
Затем, все еще находясь в режиме трансформации, потянем нижний центральный узел вверх, так чтобы высота фигуры стала примерно равна половине прежней. Затем еще раз уменьшим высоту до половины, и еще раз – в итоге должна получиться фигура с высотой, примерно равной 1/8 начальной высоты, пример на рисунке.



4.13. Нажмите клавишу V, и передвиньте блик вниз на 10 пикселей, нажав 10 раз стрелку вниз. Теперь вам нужно немного размыть края блика. Возьмите инструмент **Blur / Размытие** (иконка, похожая на каплю, вызывается клавишами R или Shift+R). Затем выберите мягкую кисть диаметром 20 пикселей и **Hardness / Жесткость=0%** и несколько раз проведите вдоль нижнего края блика (слева направо), чтобы размыть границу. Понижьте непрозрачность до 75%.

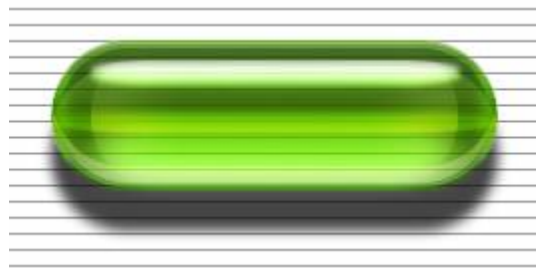


4.14. Сделаем блик поярче – для этого продублируем слой Main Hightlight, передвинем копию вниз под исходный слой, поменяем режим наложения на **Overlay / Перекрытие** и вернем непрозрачность 100%. И сдвинем слой на 3 пикселя вниз.



4.15. Теперь нарисуем блик в центре кнопки. Продублируем слой Main Hightlight и поместим копию поверх всех слоев. Отразим получившийся слой по вертикали с помощью команды в меню **Edit / Редактирование > Transform / Трансформирование > Flip Vertical / Отразить по вертикали**. Поместим его в центре рисунка, для этого нажмем клавишу Ctrl и, удерживая ее, последовательно клавиши A, X, V (выделить, вырезать, вставить). Сдвинем слой вниз еще на 1-2 пикселя.

4.14. Теперь с помощью команды **Transform / Трансформирование > Scale / Масштабирование** растянем блик влево и вправо, он должен растянуться почти на ширину всей кнопки. После растягивания понизим непрозрачность до 25% и поменяем режим наложения на **Overlay / Перекрытие**.



4.15. Два последних блика дополняют иллюзию трехмерного округлого предмета, и кнопка будет выглядеть более реалистично.

Снова выделим иконку слоя Button Color, создадим новый слой поверх остальных и назовем его Edge Hightlight Top. Зальем выделение белым цветом.

**4.16. Будьте внимательны, сейчас вам потребуются некоторые манипуляции с областью выделения – внимательно прочтите следующий абзац!**

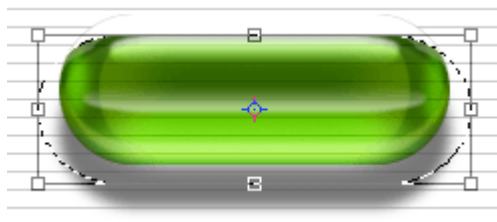
Уменьшите выделение на 10 пикселей (**Select / Выделение > Modify / Модификация > Contract / Сжать**). Нажимаем клавишу M, затем 5 раз нажмем стрелку вверх, сдвигая выделение вверх на 5 пикселей. Нажимаем **Delete / Удалить**. Теперь нажимаем 10 раз стрелку вниз, и затем 15 раз стрелку вправо. Снова нажимаем **Delete / Удалить**. И наконец, удерживая клавишу Shift, нажимаем 3 раза стрелку влево и снова нажимаем **Delete / Удалить**. Снимаем выделение. Ластиком удаляем нижний фрагмент, он нам не нужен. У нас должна получиться вот такая фигура (если ваша выглядит по-другому, вернитесь на шаг назад и переделайте):



4.17. Нажимаем клавишу V для выбора инструмента Move и один раз нажимаем стрелку вниз (сдвигаем на 1 пиксель). Изменяем режим наложения на **Overlay / Перекрытие** и понижаем непрозрачность до 25%.

4.18. Снова создадим выделение, щелкнув по слою Button Color с зажатой клавишей Ctrl. Создадим новый слой поверх других, назовем слой Top Highlight и зальем выделение белым цветом. Нажмем клавишу M и 10 раз нажмем стрелку вниз.

4.19. Выберем в меню **Select / Выделение > Transform Selection / Трансформировать выделенную область**. Нажав клавишу Alt растягиваем выделение в ширину за правый средний узел примерно на 10 – 15 пикселей. Нажимаем Enter, Delete, снимаем выделение.



4.20. Нажимаем клавишу V и 5 раз стрелку вниз. Меняем режим наложения слоя на **Overlay / Перекрытие** и снижаем непрозрачность до 75%.

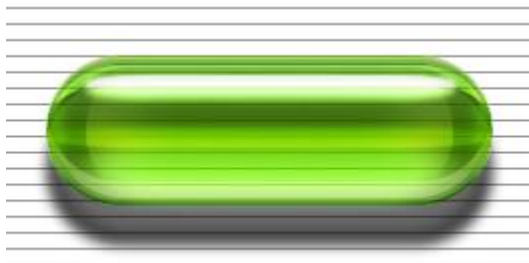


## 5. Последние штрихи: тень

Если вы возьмете настоящую капсулу из цветного стекла и поместите ее на однородный цветной фон, вы заметите, что тень не будет ни черной, ни сплошной. Если освещение в комнате таково, что свет проходит через капсулу, то от поверхности капсулы будут отскакивать блики. Если капсула прозрачная или даже полупрозрачная, свет будет проходить через нее и внутри тени будет световое пятно. И сама тень будет такого же цвета, как стекло, или близкого к нему.

Таким образом, чтобы приблизить изображение к реальности, необходимо добавить внутреннее свечение тени кнопки и изменим цвет тени так, чтобы он соответствовал цвету кнопки. Сделайте следующее:

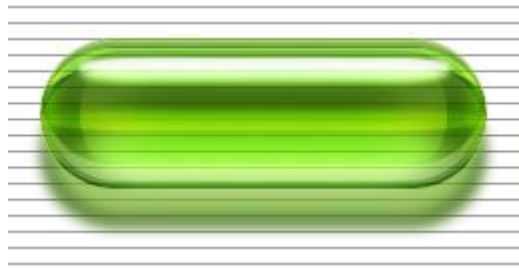
5.1. Продублируйте слой **Shadow / Тень**, назовите новый слой Shadow Glow и расположите его под слоем **Shadow**. Нажимаем Ctrl + I, чтобы инвертировать цвета. Меняем режим наложения слоя на **Overlay / Перекрытие**, непрозрачность 50%. Переключаемся на инструмент Move и с помощью стрелки вверх на клавиатуре сдвигаем слой на 3 пикселя вверх. Теперь идем в меню **Filter / Фильтр > Other / Другое > Maximum / Максимум** и задаем значение 5 пикселей — сожмем нашу фигуру.



5.2. Вернемся на слой **Shadow / Тень** и изменим цвет тени на зеленый: R=51, G=102, B=0 (например, с помощью **Blending Options / Параметры наложения > Color Overlay / Наложение цвета**).

Проверим еще раз: у слоя **Shadow / Тень** должен быть режим наложения **Multiply / Умножение** и непрозрачность 75%.





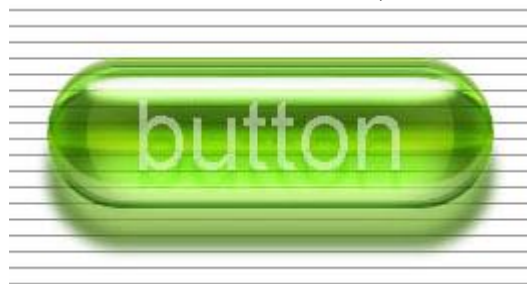
## 6. Текст

Текст, размещенный на внешней поверхности кнопки или внутри нее, может усилить трехмерный эффект. Но в любом случае, необходимо не забывать о направлении света, как свет будет влиять на используемый текст. В этой практической работе мы поместим текст не на поверхность, а внутрь прозрачной капсулы.

6.1. Переходим на слой Faded BG, выбираем инструмент **Туре / Шрифт** и щелкаем по левой стороне кнопки, примерно в том месте, где будет начинаться ваш текст. Позже вы можете использовать свой текст, а сейчас, для примера, напечатайте слово Button.

Итак, напечатаем текст, выберем шрифт (я выбрал Arial), цвет текста - белый. Для нашего примера не имеет значения, будем ли мы растривать слой с текстом, конечный результат от этого не изменится. Понижим непрозрачность слоя до 50%, так он будет лучше сочетаться с нашей кнопкой.

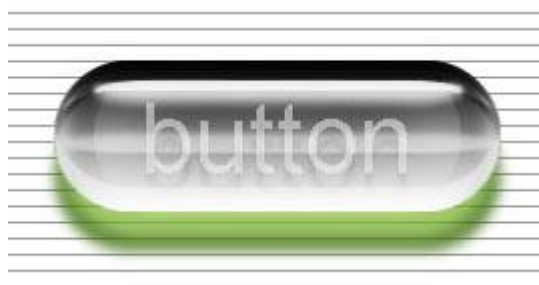
6.2. Для создания тени продублируем слой с текстом, поместим дубликат под исходный слой, поменяем цвет текста на черный, поменяем режим наложения на **Soft Light / Мягкий свет** и понизим непрозрачность до 35%. Можно немного размыть слой, используйте для этого фильтр **Blur More / Размытие + / Размыть больше**. Photoshop предложит растривать слой перед применением фильтра, нажимаем ОК. В завершении, нажмем V на клавиатуре, затем Shift и стрелку вниз (передвинем слой вниз на 10 пикселей).



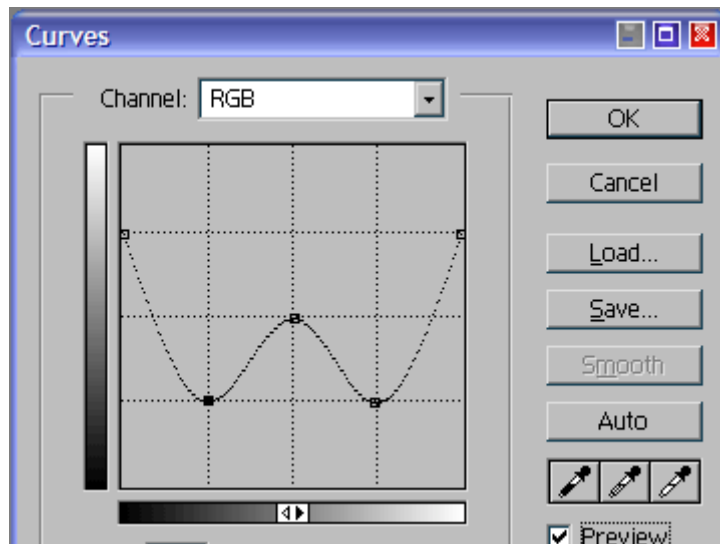
## 7. И еще пара штрихов для завершения

Следующие шаги необязательны, но они усилят эффект, придав вашей кнопке большую глубину. С помощью простого черно-белого градиента мы немного усилим блики и тени, а с помощью текстур мы можем имитировать дополнительные блики на поверхности или структуру поверхности.

7.1. Снова сделаем выделение, нажав **Ctrl** и щелкнув по слою **Button Color**. Создадим новый слой и поместим его над слоем Large Refraction. Назовем этот слой Depth Lighting. Зальем слой черно-белым градиентом по направлению сверху вниз. Снимем выделение.



7.2. Выбираем в меню **Image / Изображение > Adjustments / Коррекция > Curves / Кривые**. Изменив форму кривой так, как показано на рисунке, мы добьемся нужного нам эффекта.



Мы превращаем линейный черно-белый градиент в более сложный: светлое-темное-светлое-темное-светлое. Поменяем режим наложения на **Soft Light /Мягкий свет**.



### **Практическая работа №34 (ПР34)**

**Тема:** Растровый графический редактор Photoshop

**Цель:** Добавление винтажного эффекта фотографии в Adobe Photoshop

В этой практической работе рассматривается способ добавления ретро/винтажного эффекта вашей фотографии, используя несколько основных приемов, фильтров и эффектов.

В результате выполнения практической работы у вас получится вот такой результат.





### Этап 1. Создаем фон.

#### Шаг 1.

Откройте в фотшоппе любую вашу фотографию. В задании, например используется вот эта.



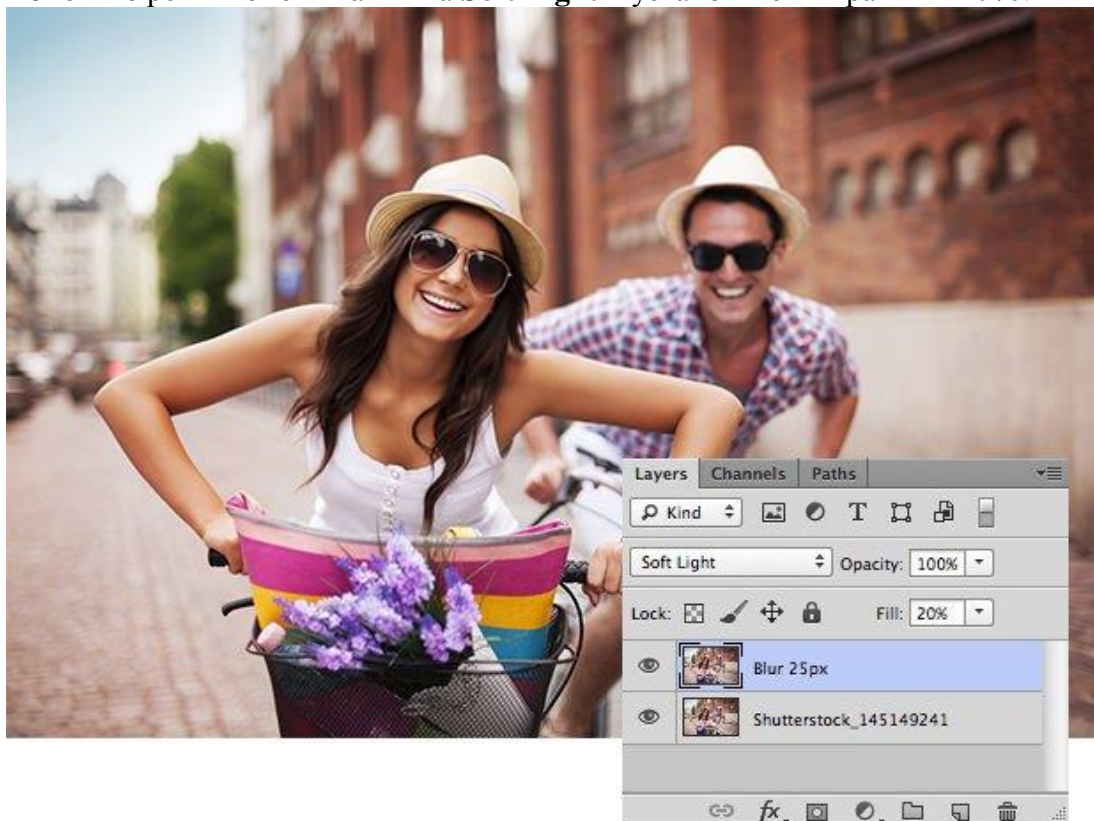
#### Шаг 2.

Продублируйте слой (правая кнопка мыши по слою > **Duplicate Layer**) и примените фильтр **Filter > Blur > Gaussian Blur** с радиусом размытия **25px**.



### Шаг 3.

Измените режим смешивания на **Soft Light** и установите **Fill** равным **20%**.

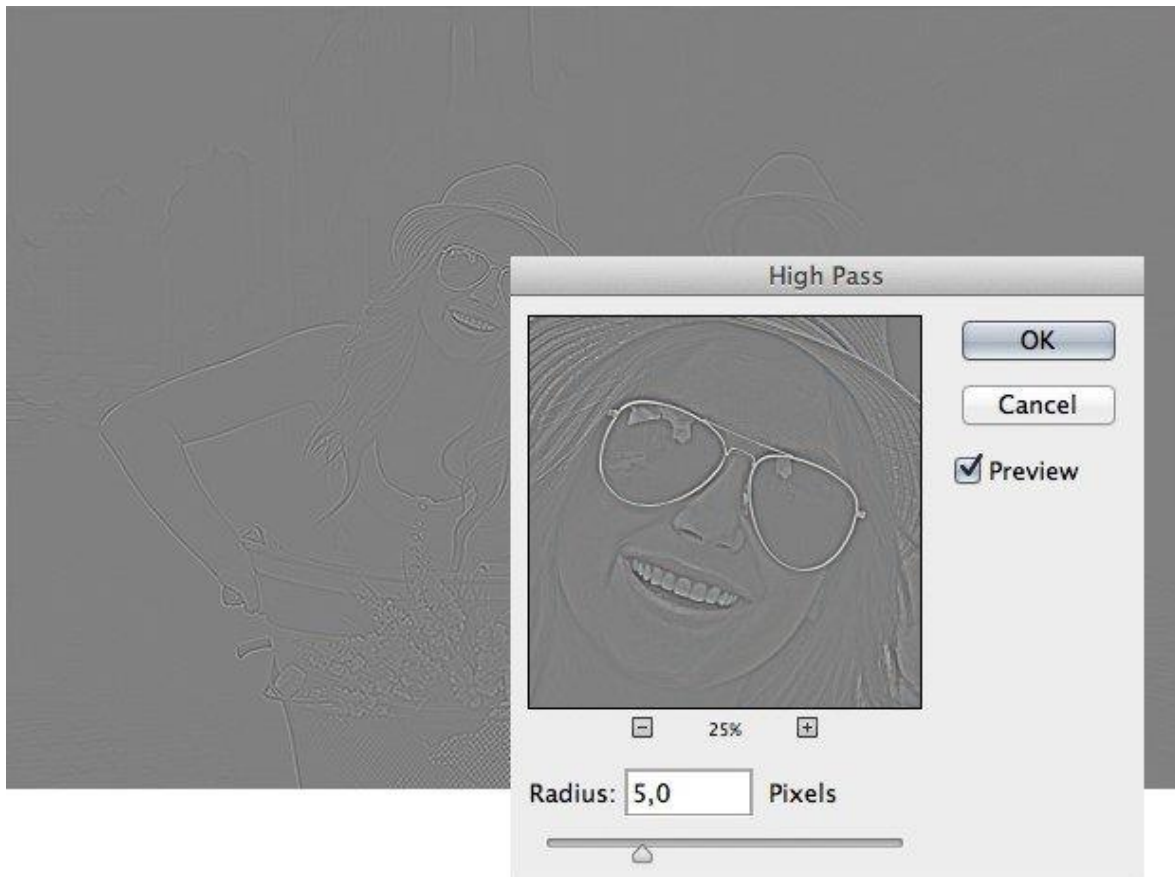


Этап 2. Добавим фильтр High Pass.



### Шаг 1.

Продублируйте слой с оригинальным фото еще раз и примените фильтр **Filter > Other > High Pass** установив радиус **5 px**.



### Шаг 2.

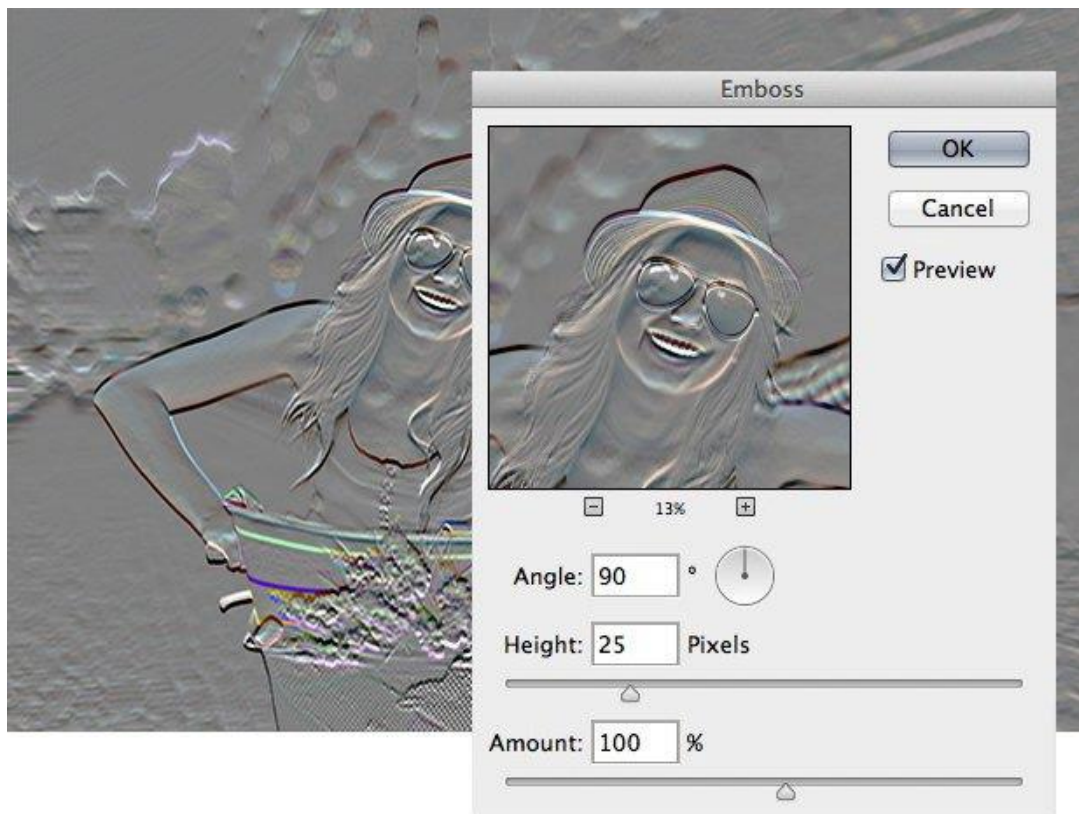
Измените режим смешивания этому слою на **Soft Light**.



### Этап 3. Теснение.

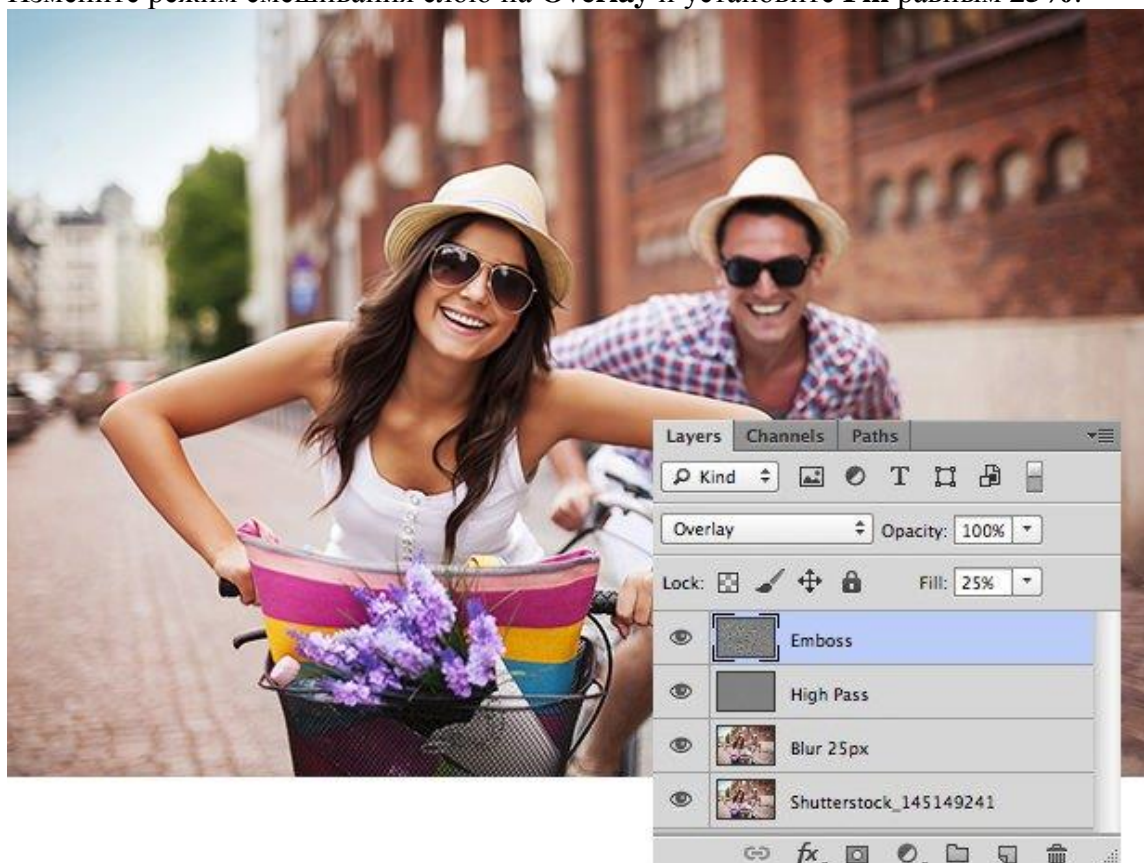
#### Шаг 1.

Продублируйте слой с оригинальным фото еще раз и примените фильтр **Filter > Stylize > Emboss** установив значения, как на скрине ниже (в зависимости от фотографии можете поставить поменьше)



#### Шаг 2.

Измените режим смешивания слою на **Overlay** и установите **Fill** равным **25%**.





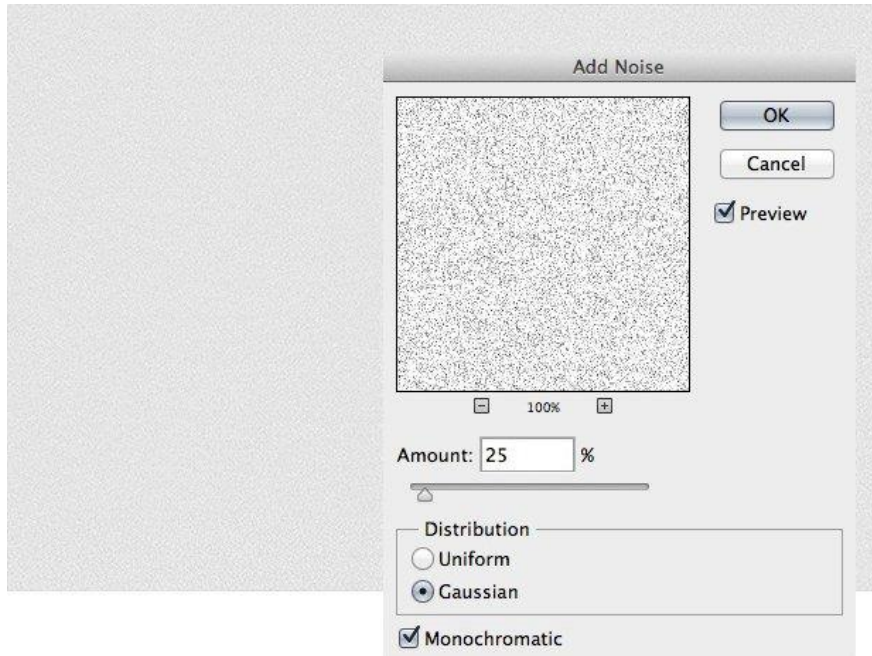
## Этап 4. Шум.

### Шаг 1.

Создайте новый слой (**Ctrl+Shift+N**), заполните его белым цветом (выберите инструмент выделения, нажмите **Ctrl+A** чтобы выделить всю область и нажмите **правую кнопку мыши > Fill** и заполните белым цветом)

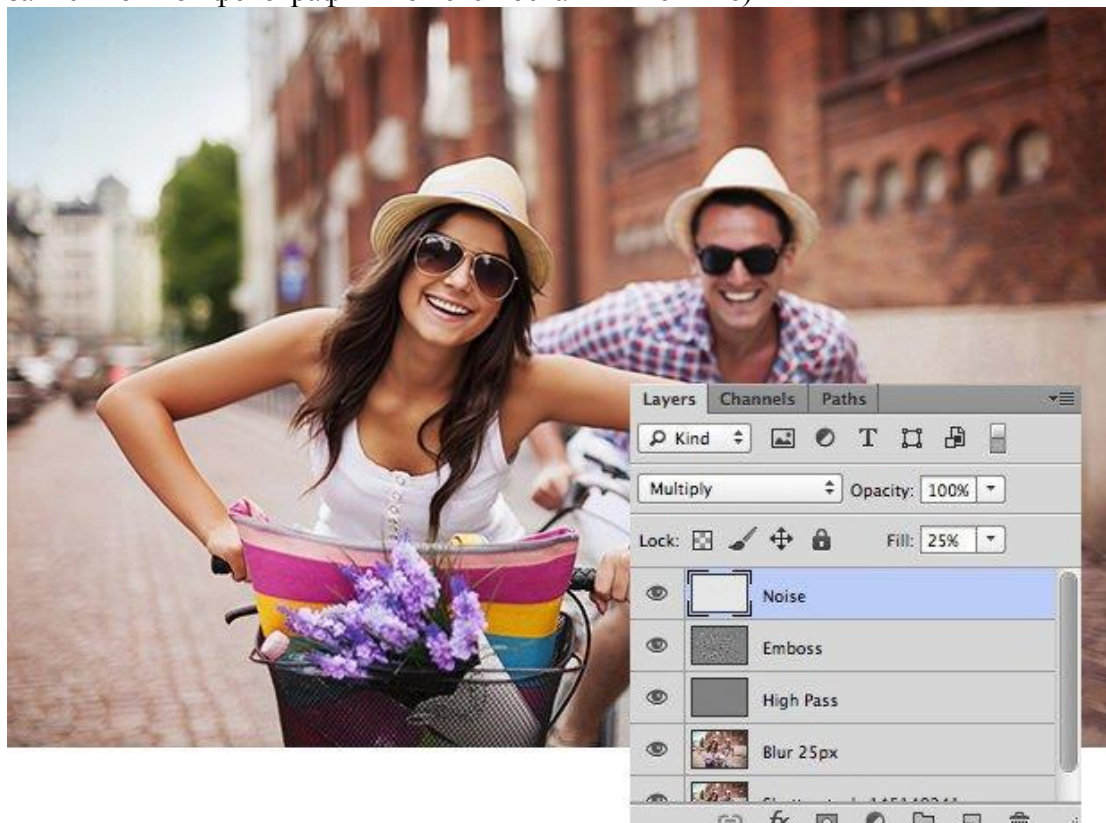
Примените фильтр **Filter > Noise > Add Noise** (убедитесь, что цвета переднего и заднего планов установлены на черный и белый)

Установите значения **как на скрине ниже**.



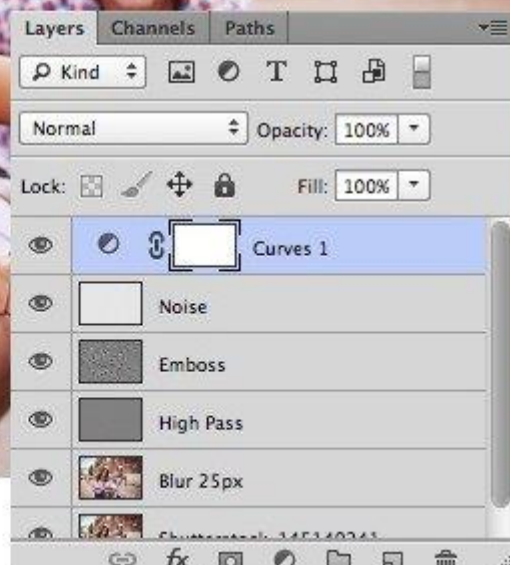
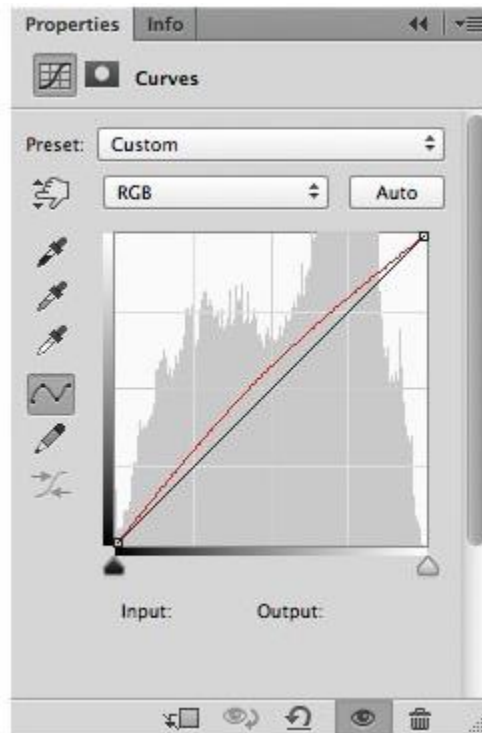
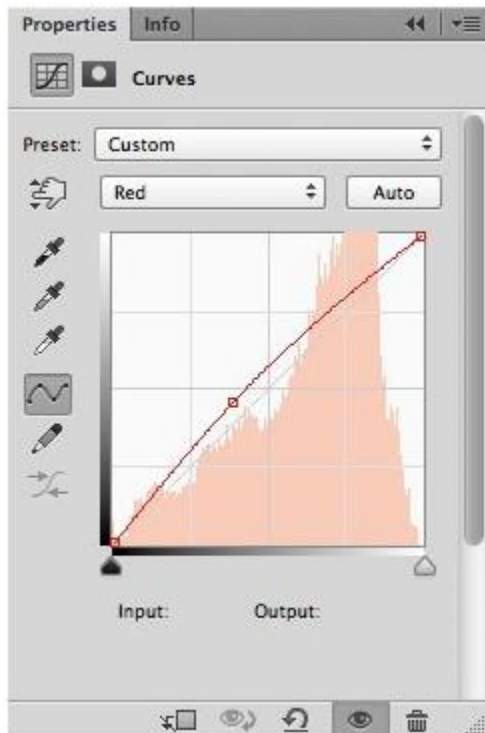
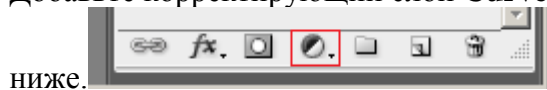
### Шаг 2.

Измените режим смешивания данного слоя на **Multiply** и установите **Fill** равным **25%** (В зависимости от фотографии можете поставить меньше)



## Этап 5. Корректирующий слой Curves.

Добавьте корректирующий слой **Curves** и измените красный канал, как на скрине

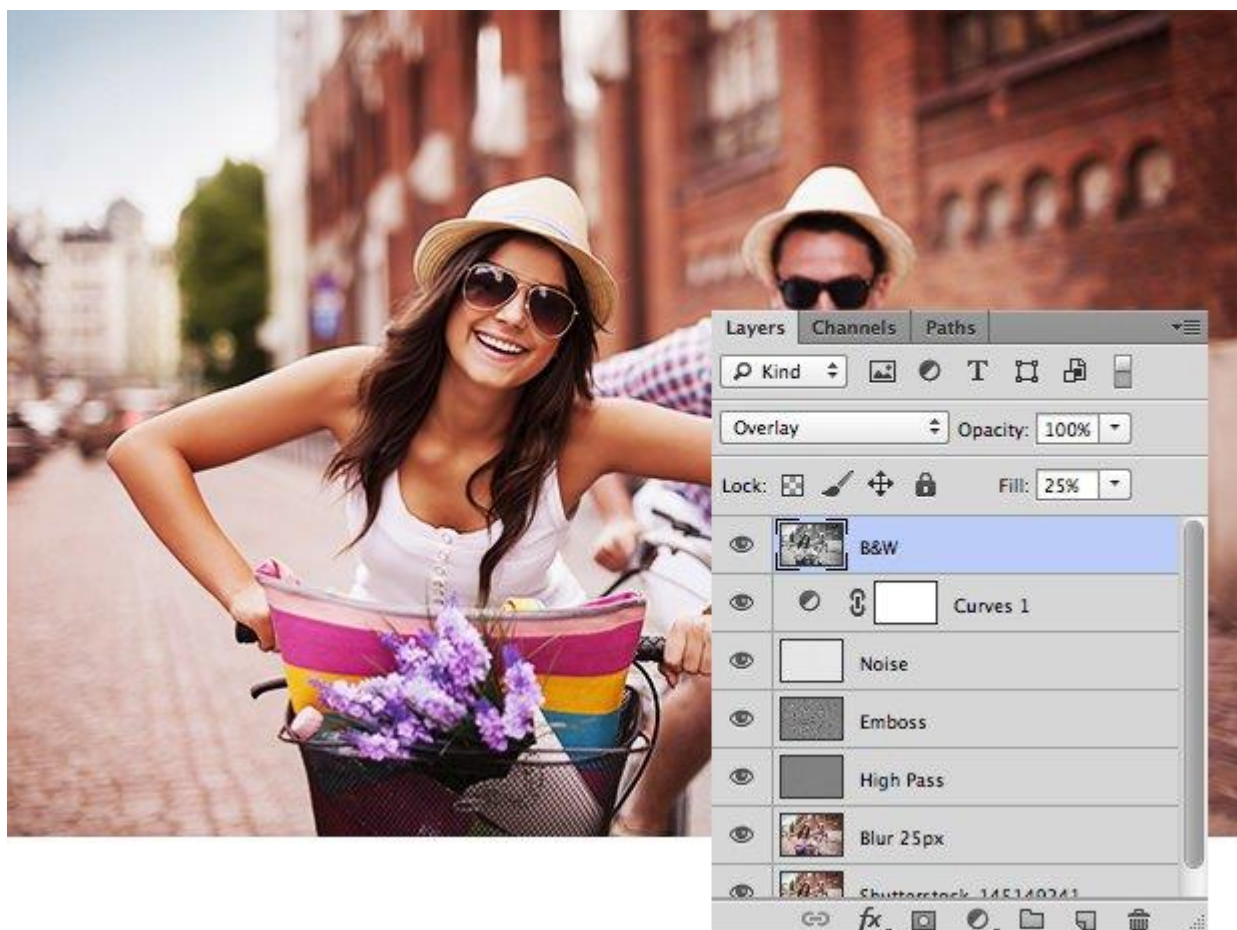




## Этап 6. Черное и белое и размытие.

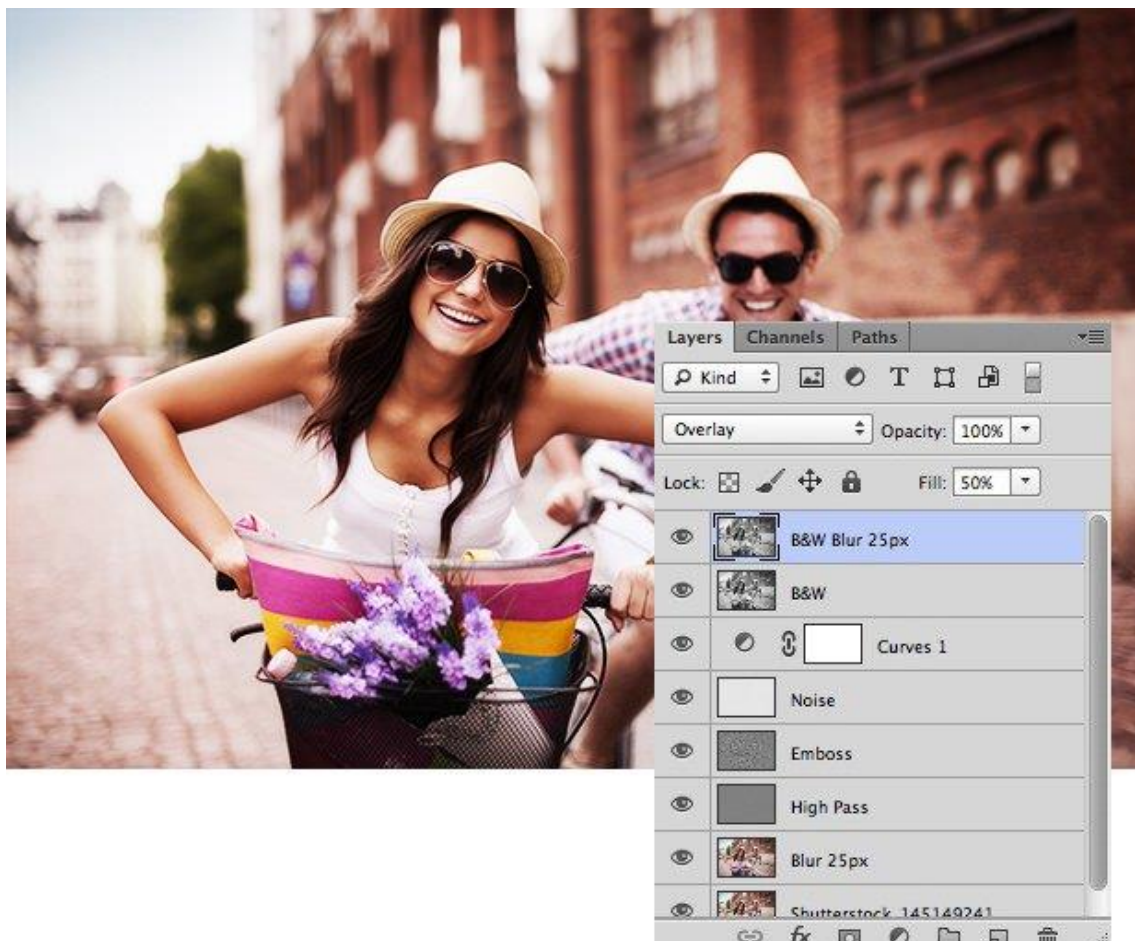
### Шаг 1.

Продублируйте оригинальное фото еще раз и переведите его в черно-белые цвета **Ctrl+Shift+U**. Измените режим смешивания для слоя на **Overlay** и установите **Fill** равным **25%**.



## Шаг 2.

Продублируйте черно-белый слой и примените фильтр **Filter > Blur > Gaussian Blur** установив радиус размытия **25 px** и установите **Fill** равным **50%**.





### Этап 7. Добавляем белый цвет.

Создайте новый слой и заполните его белым цветом, установите **Fill** равным **10%**.



### Этап 8. Корректирующий слой Gradient Map.

#### Шаг 1.

Добавьте корректирующий слой **Gradient Map** установив следующие цвета для маркеров. Для левого **182 37 154**, для правого **255 244 41**.



## Шаг 2.

Измените режим смешивания на **Color** и установите значение **Fill** на **25%**.



## Этап 9. Цвета.

### Шаг 1.

Создайте новый слой, выберите инструмент **Brush Tool** с мягкими, круглыми краями и используя два цвета голубой и пурпурный, красьте как показано ниже.

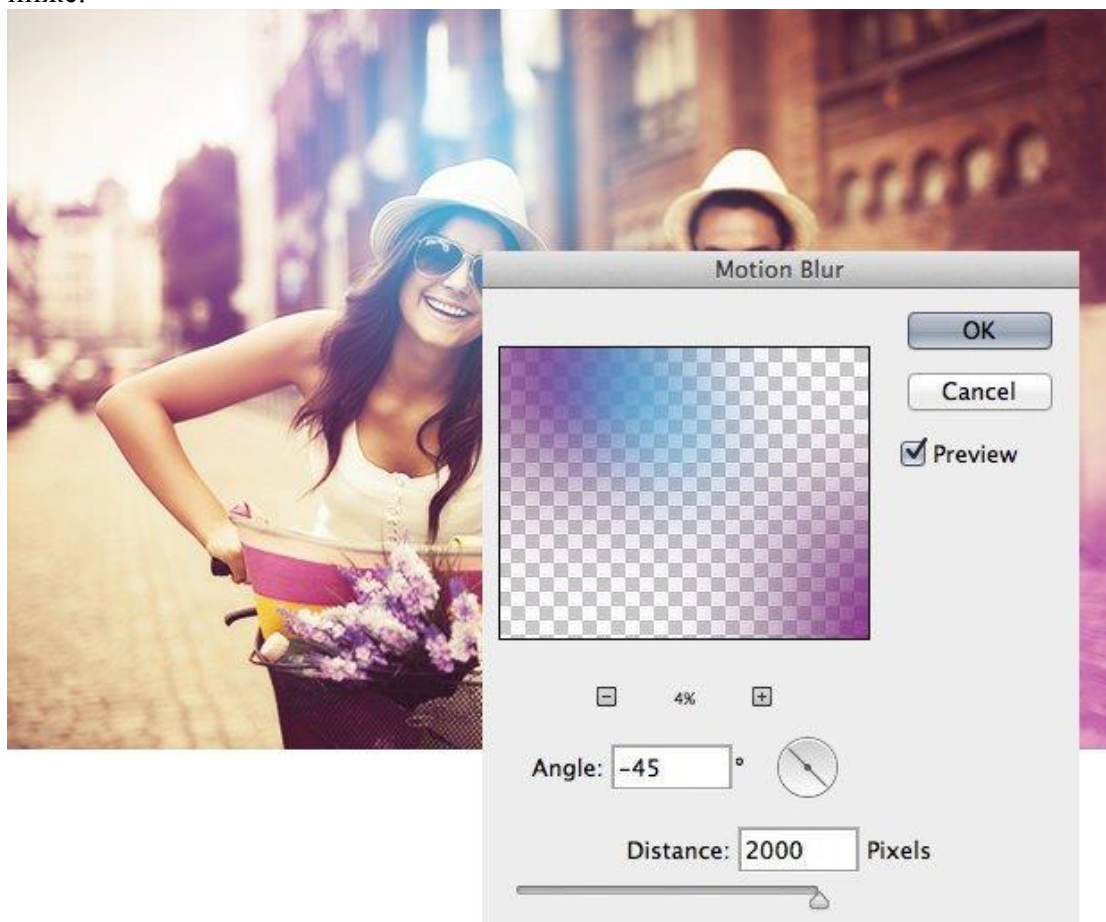






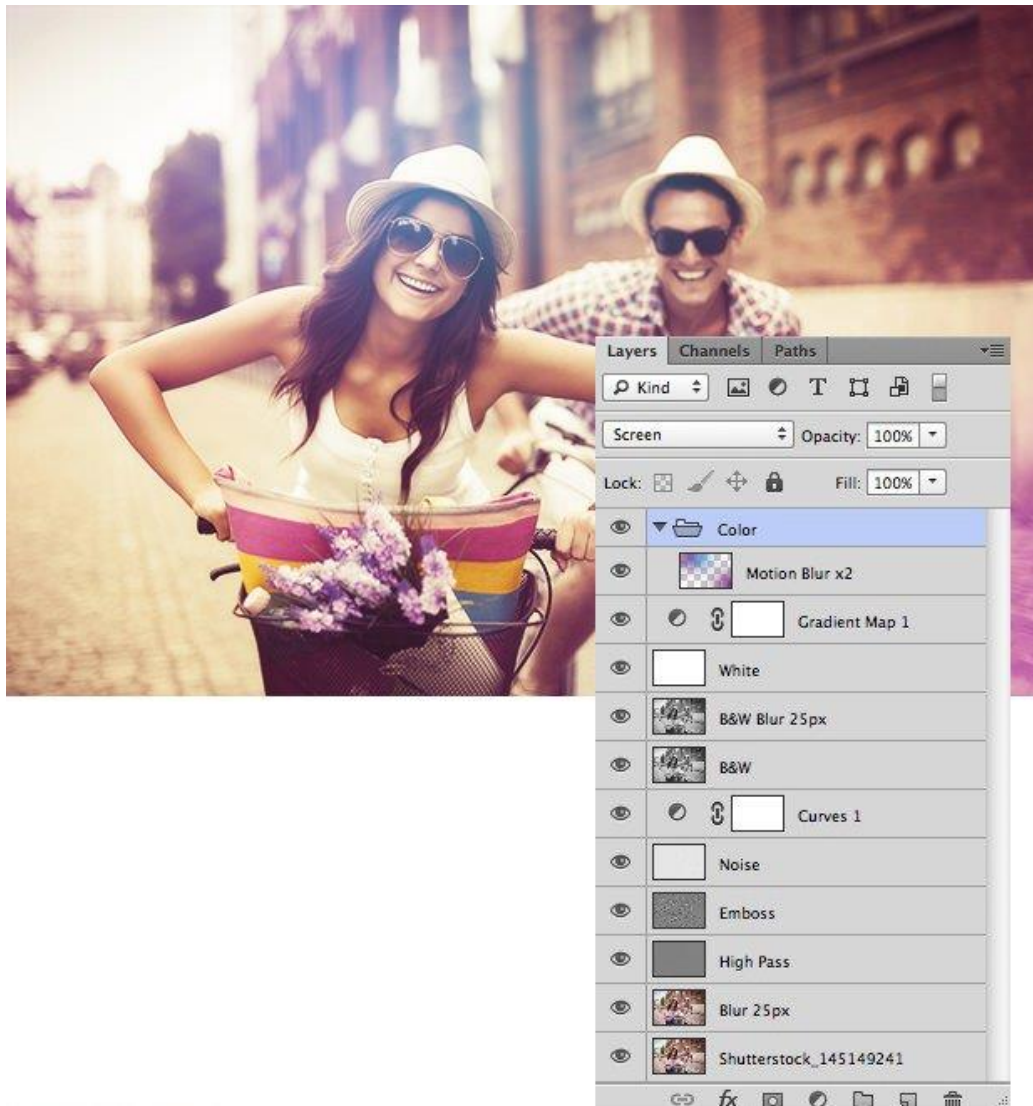
**Шаг 2.**

Примените к данному слою фильтр **Filter > Blur > Motion Blur** с настройками, как на скрине ниже.



**Шаг 3.**

Измените режим смешивания на **Screen**.



Вот и все...

Поздравляю вы выполнили задание

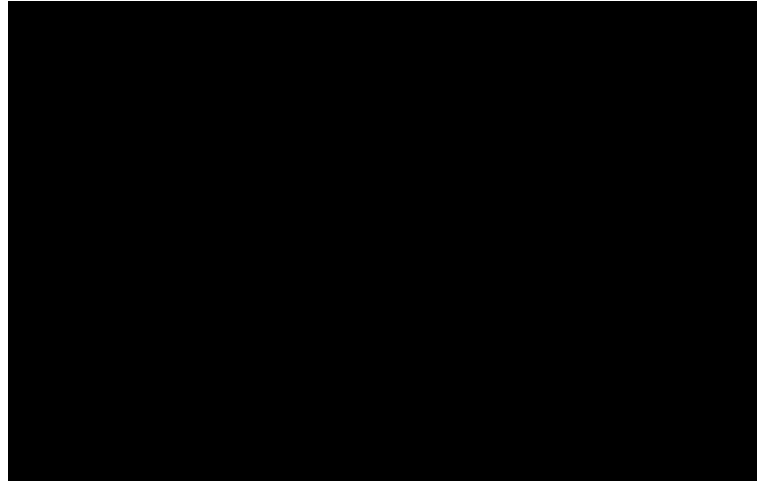


## Практическая работа №35 (ПР35)

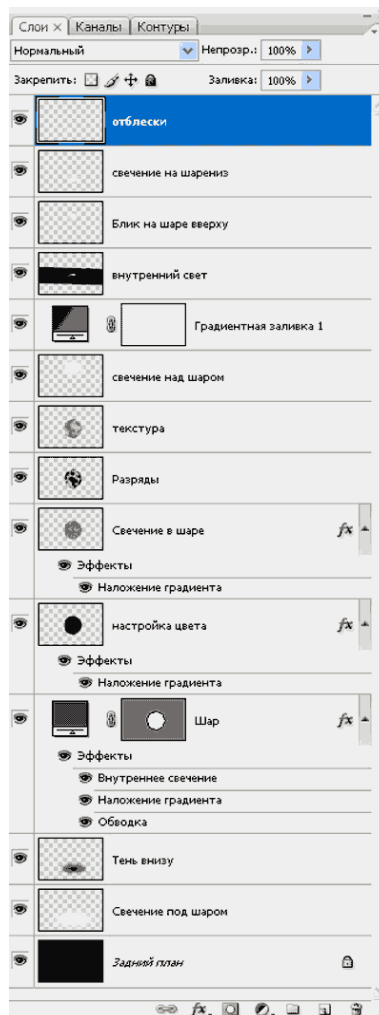
**Тема:** Растровый графический редактор Photoshop

**Цель:** Создание объёмного светящегося шара в Adobe Photoshop

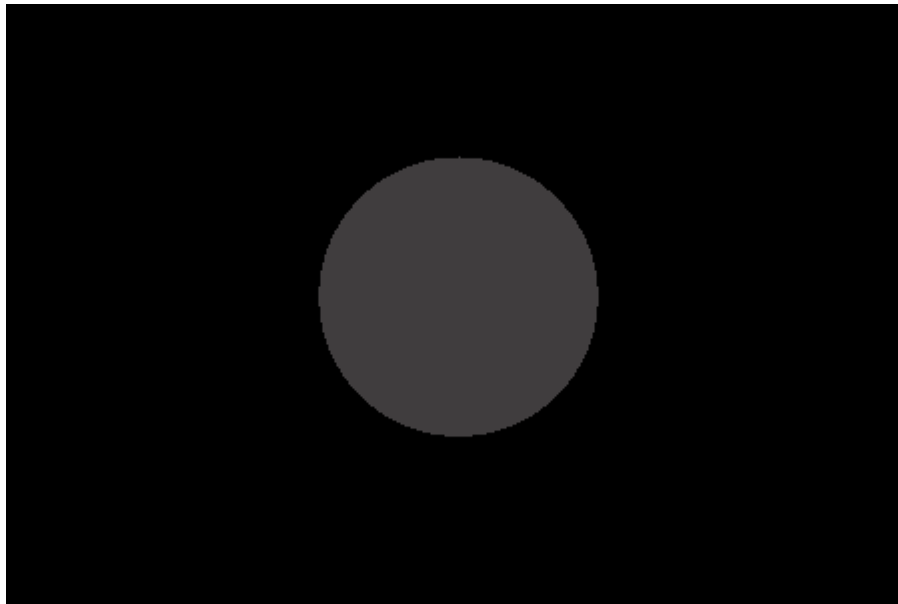
**Шаг 1.** Создайте новый документ размером **1900x1200 пикс.** Это можно сделать, нажав **CTRL+N**, либо через меню **Файл – Новый (File – New)**. Залейте документ черным цветом. Для этого сбросим цвета, нажав **D**, затем нажмем **ALT+BACKSPACE**.



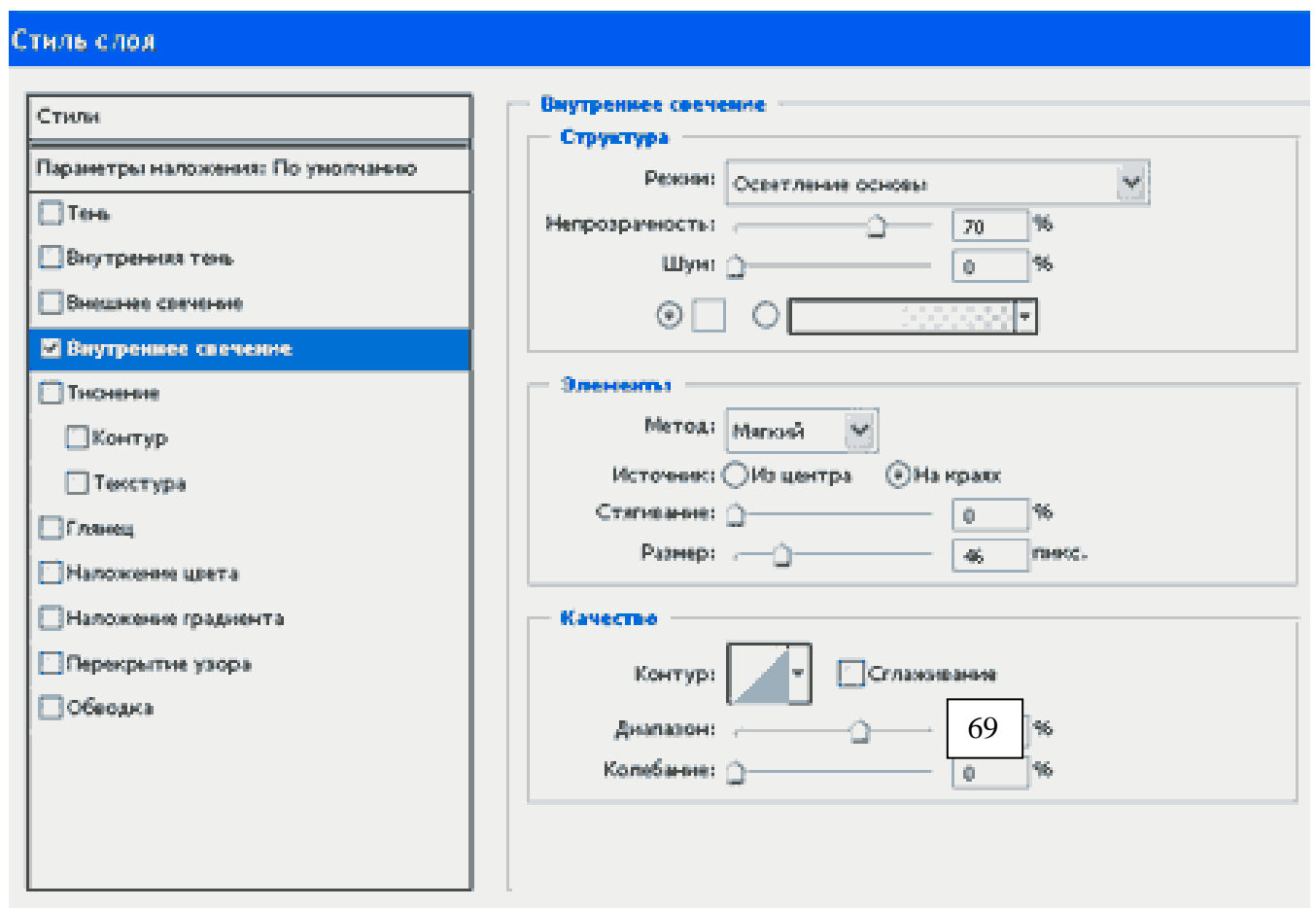
На следующем рисунке я привожу порядок и названия всех слоев, чтобы вы могли сверяться по нему. Поэтому я не буду подробно останавливаться на том, где какой слой размещать.



**Шаг 2. Создайте новый слой.** Выберем инструмент **Эллипс (Ellipse Tool)**, затем нарисуем правильный круг в центре, для этого нужно удерживать клавишу **SHIFT**. Залейте выделенный круг черным цветом.

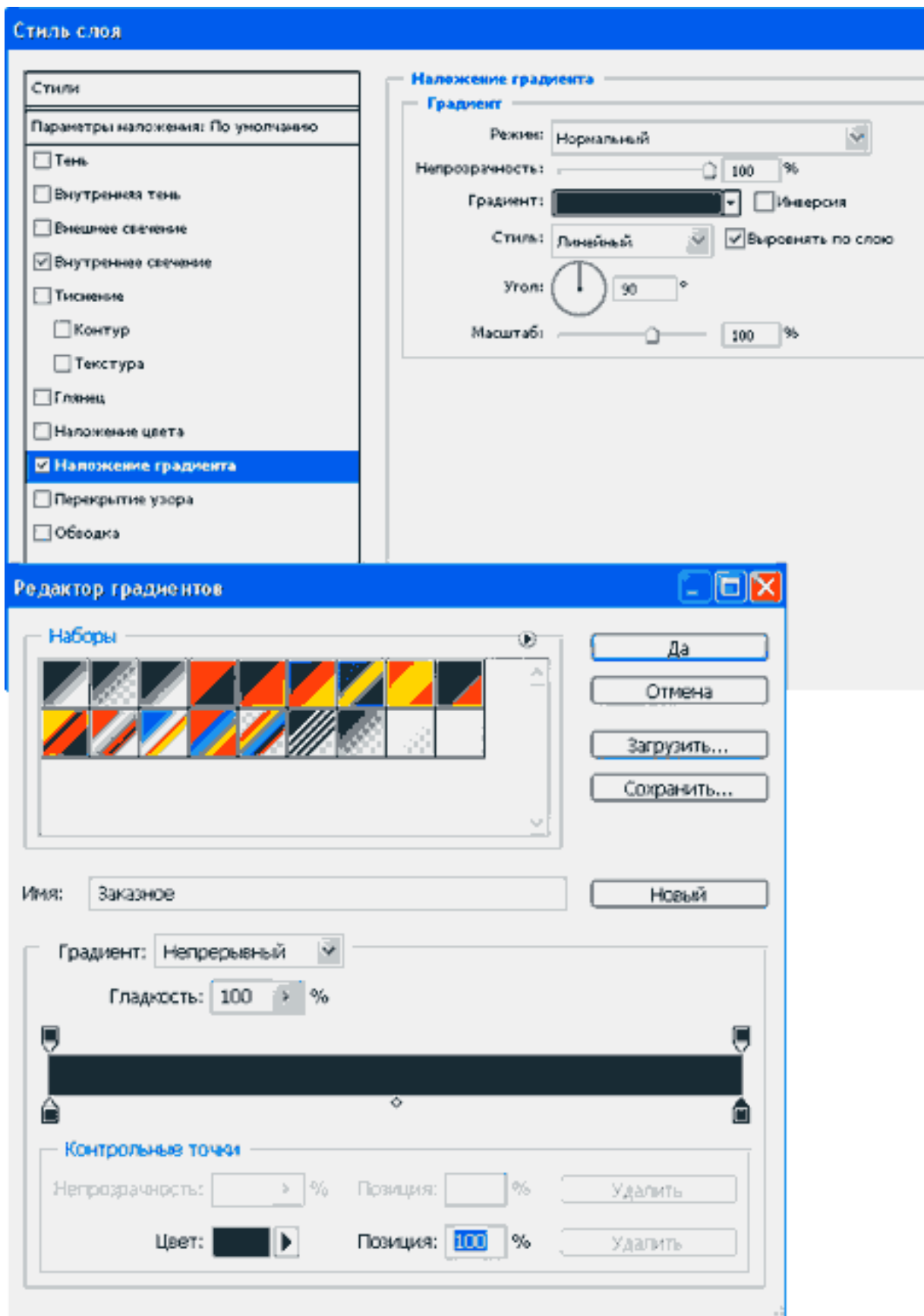


Применим к слою стили. **Слой – Стиль слоя – Параметры наложения (Layer – Layer Style – Blending Options)**. Сначала выберем **Внутреннее свечение (Inner glow)**. Используйте белый цвет, режим наложения - **Осветление основы (Color Dodge)**, непрозрачность - 70%. Мягкий (Softer)

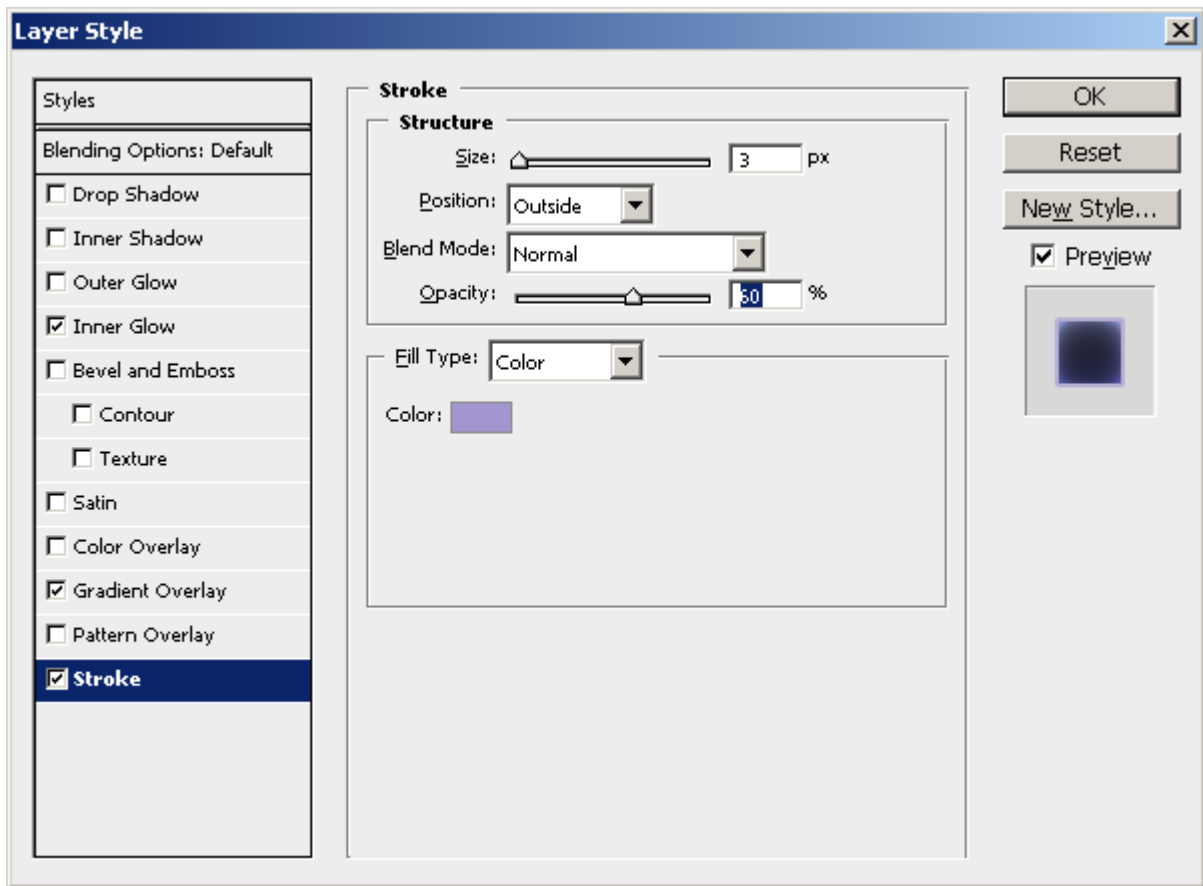


Далее переходим в подменю **Наложение градиента (Gradient Overlay)**. Используем линейный стиль, режим смешивания **Нормальный**. Цвета для градиента выбираем **18 19 44** и **49 52 68**.

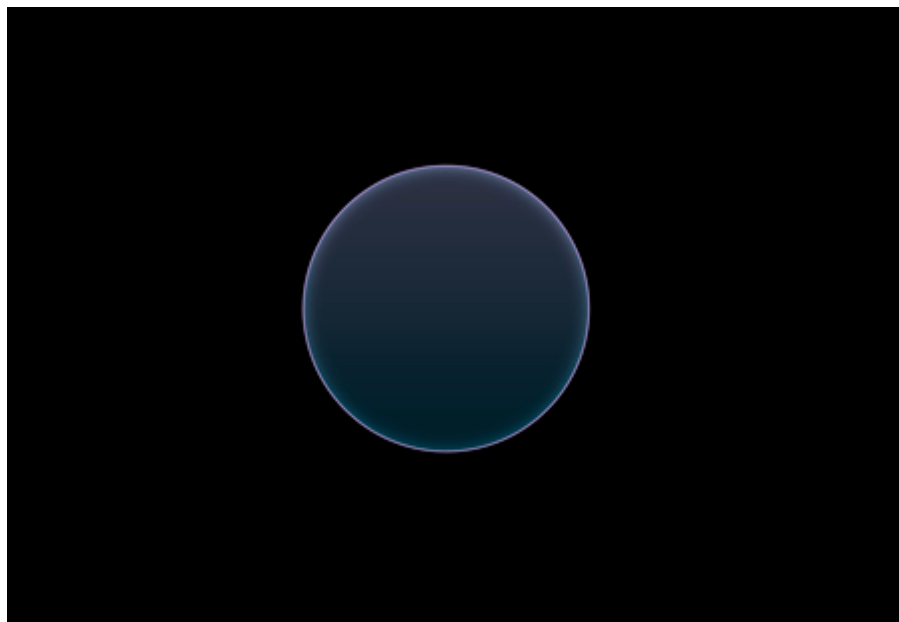




Перейдем в подменю **Обводка** (Stroke). Выберем цвет **163 149 207**, непрозрачность 60%, положение – внутри.



У вас должно получиться вот так.



**Шаг 3.** Выберем инструмент **Эллипс**. Теперь мы будем использовать его в режиме создания контура, а не фигуры. Нарисуйте эллипс в верхней части круга. При необходимости отрегулируйте его положение перемещая.

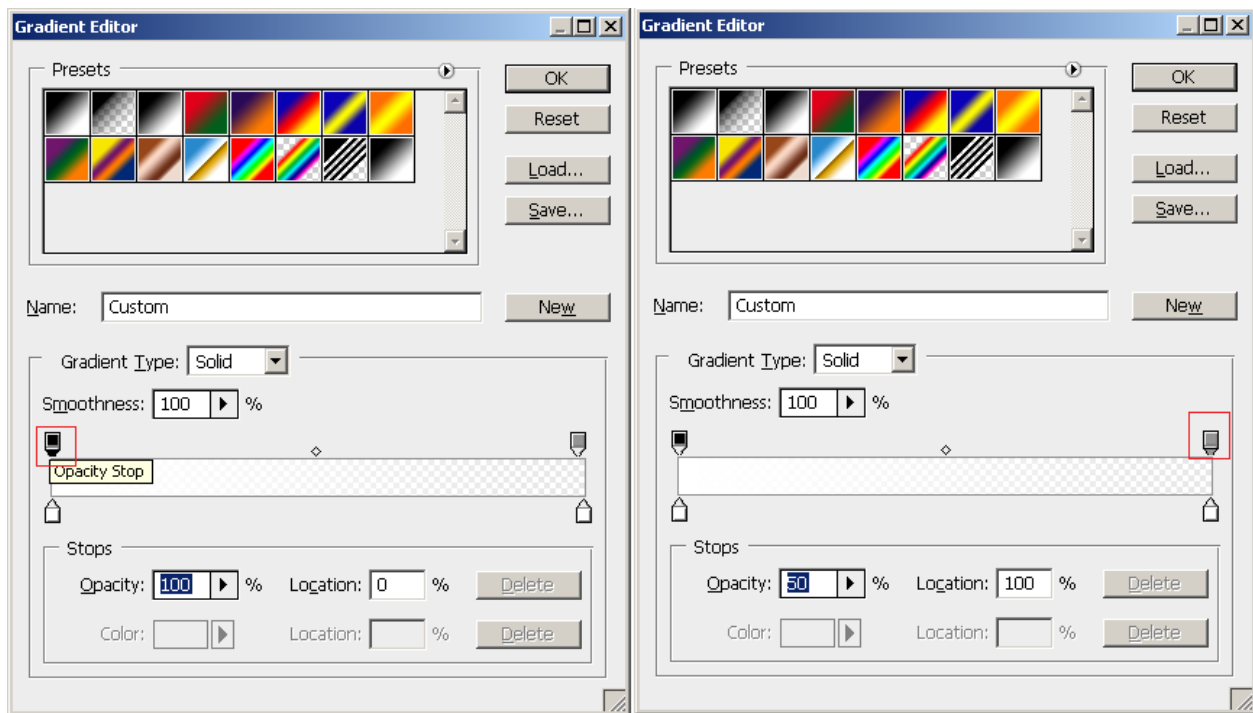




Создадим новый слой и назовем его «**Бликверху**». Всегда давайте слоям имена, ведь когда их образуется три – четыре десятка в процессе работы (бывает и три-четыре сотни), очень легко запутаться.

Выберем инструмент **Градиент** (Gradient Tool). Используем **белый** (255 255 255) цвет на обоих концах градиента.

Непрозрачность установите соответственно **100** и **50%**.

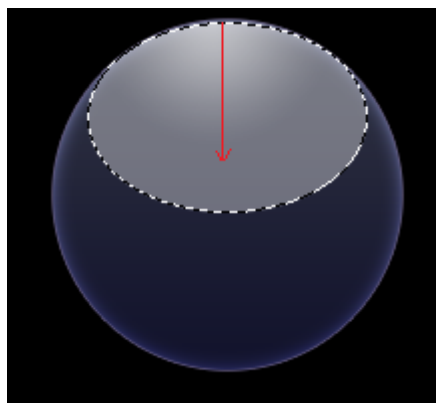


Чтобы установить цвета, нужно кликнуть по квадратикам внизу полосы редактирования, а непрозрачность – по квадратикам сверху полосы.

**Стиль градиента** используем **Radial**, непрозрачность **70%**.



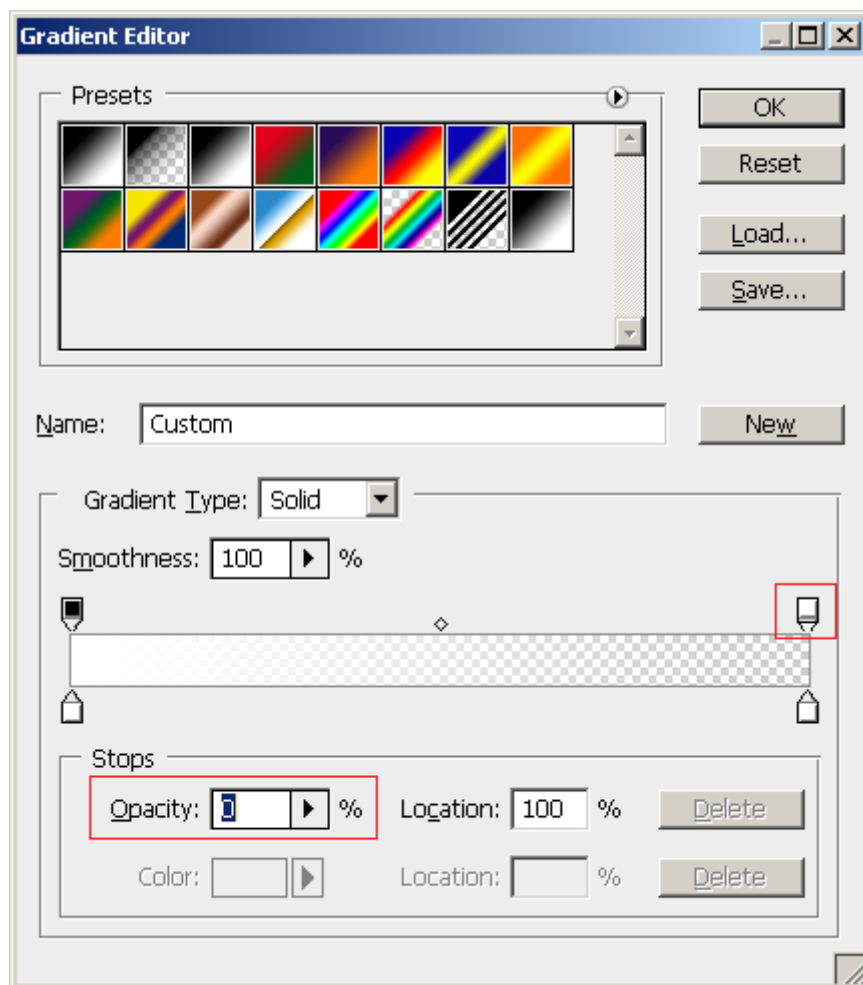
Залейте выделенную область как на рисунке ниже с помощью полученного градиента. Для этого проведите градиент с верхнего до нижнего края выделенной круглой области захватив  $\frac{3}{4}$  части окружности



**Шаг 5.** Теперь создадим свечение внизу нашего шарика.

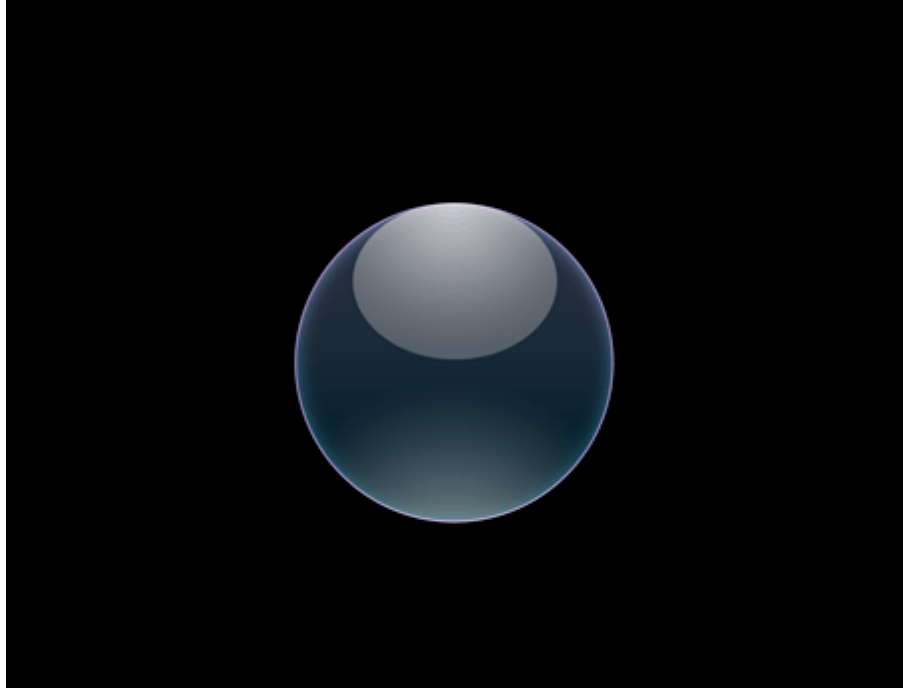
Для этого выделим контур фигуры, щелкнув по иконке главного слоя мышкой с нажатой клавишей **CTRL**. Создадим новый слой и назовем «Свечение внизу».

Выберем инструмент **Градиент**, изменим значение непрозрачности на правом конце с **50** на **0%**.



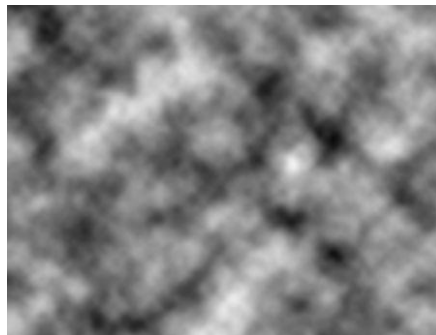
**Непрозрачность слоя (Opacity)** установим около **50%**.

Теперь создадим свечение внизу шара, проведя снизу окружности вертикально вверх до середины окружности.

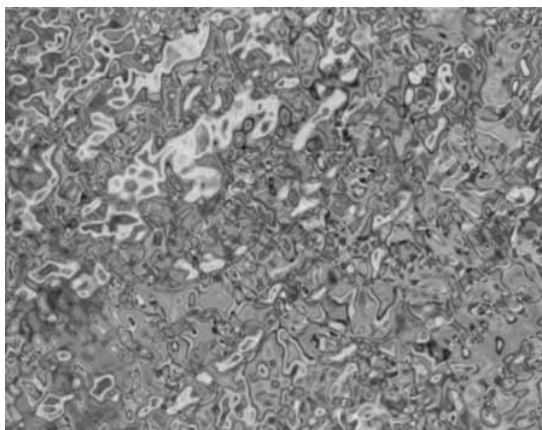


**Шаг 6.** Снимите выделение Ctrl + D

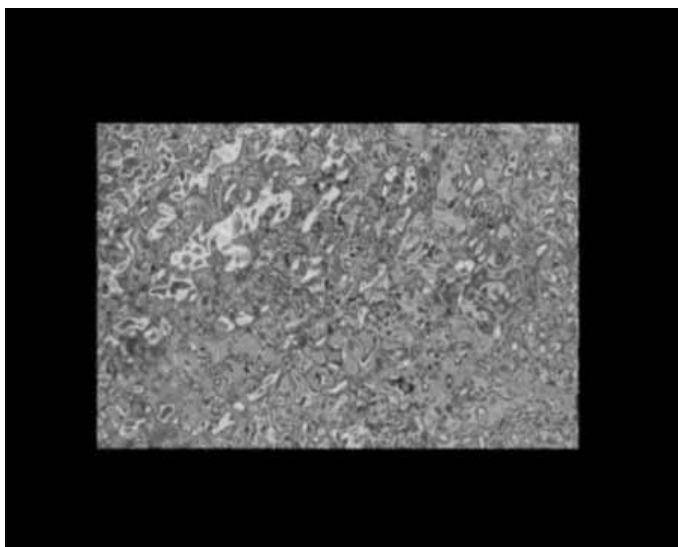
Создадим новый слой и назовем его «**Свечение в шаре**». Убедитесь, что основной и фоновый цвет белый и черный. Применим следующие фильтры: **Фильтр – Рендеринг – Облака** (Filter – Render – Clouds)



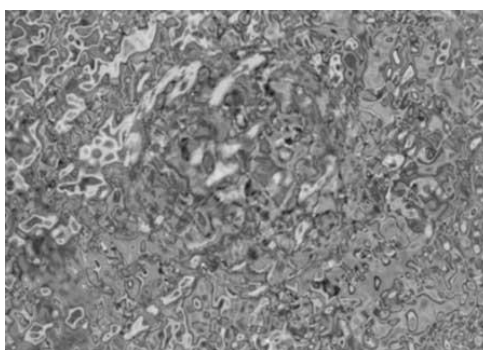
**Фильтр – Эскиз – Хром** (Filter – Sketch – Chrome). Детализация 3, Смягчение 10. ОК



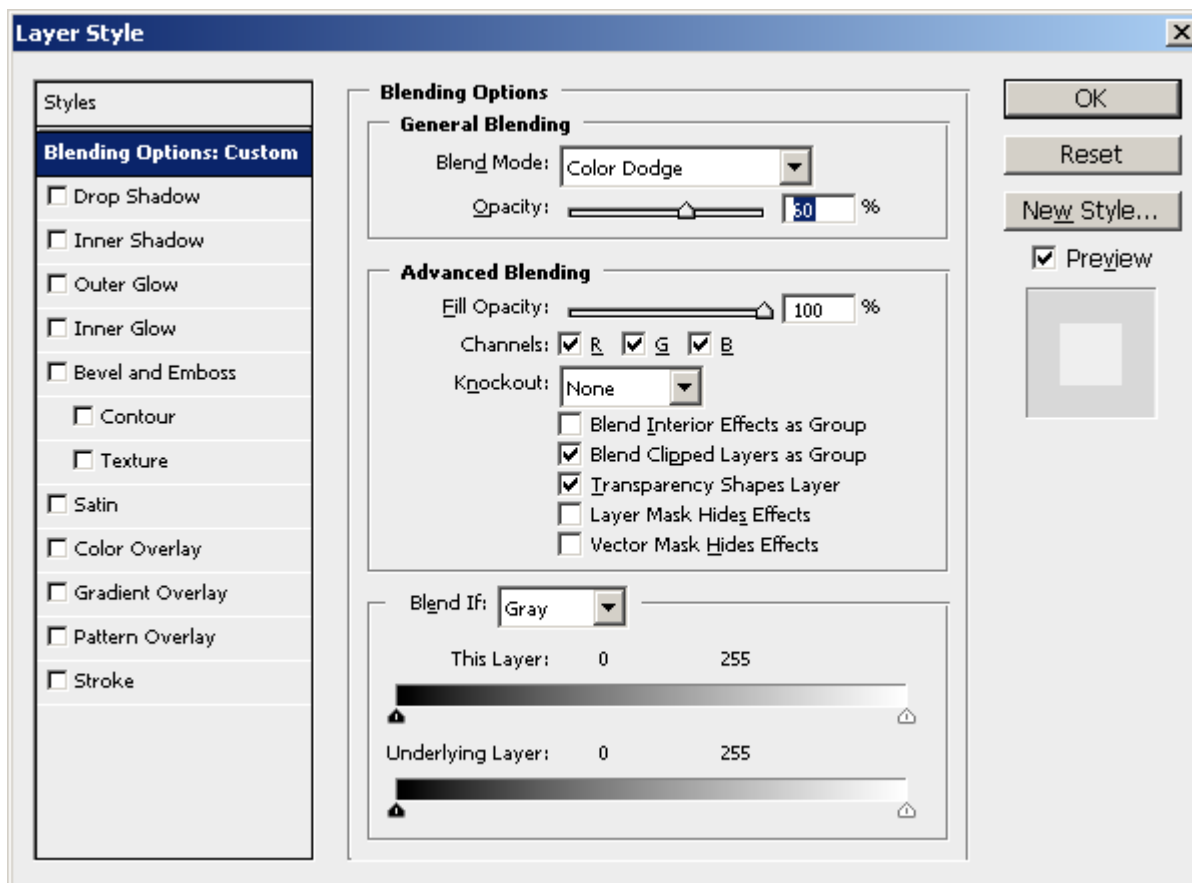
Затем применим свободное трансформирование, для этого нажмем **CTRL+T**, и изменим размер слоя так, чтобы он немного перекрывал шар. Примените трансформацию нажав галочку в верхнем правом краю окна



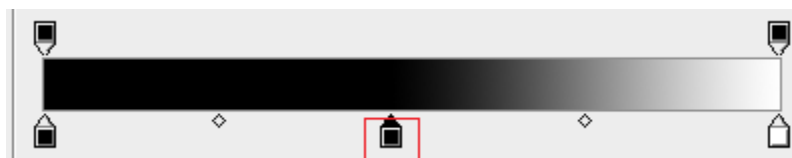
**Шаг 7.** Загрузим фигуру шара как выделение. Применим **Фильтр – Искажение – Сферизация** (Filter – Distort – Spherize).  
Затем **Фильтр – Штрихи – Аэрограф** (Filter - Brush Strokes - Sprayed Strokes).  
Длина штриха 17, радиус разбрызгивания 7.



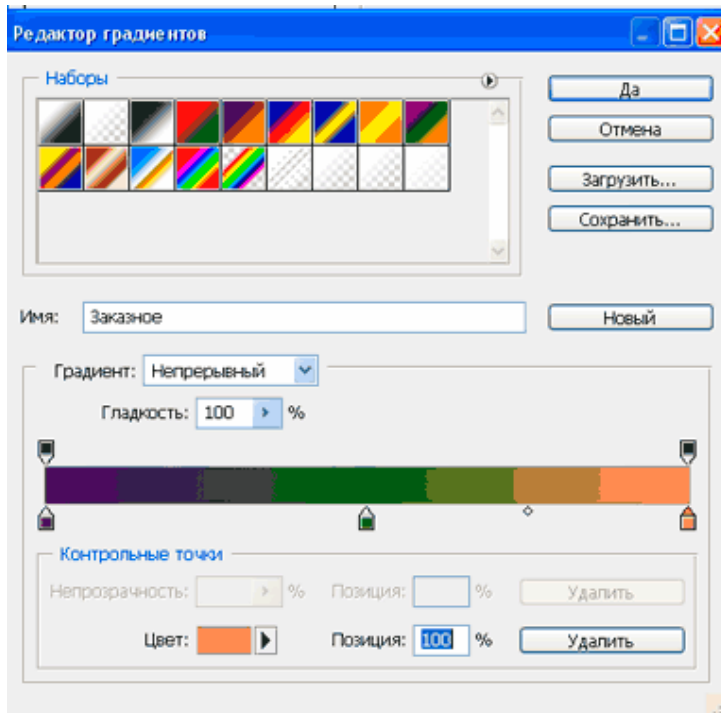
**Шаг 8.** Теперь добавим цвет нашему шару.  
Идем в меню **Слой – Стиль слоя – Параметры наложения**. (**Layer > Layer Style > Blending Options**)  
Изменим режим наложения (**Blend Mode**) на **Осветление основы** (**Color dodge**),  
непрозрачность (**Opacity**) заливки поставим **60%**.



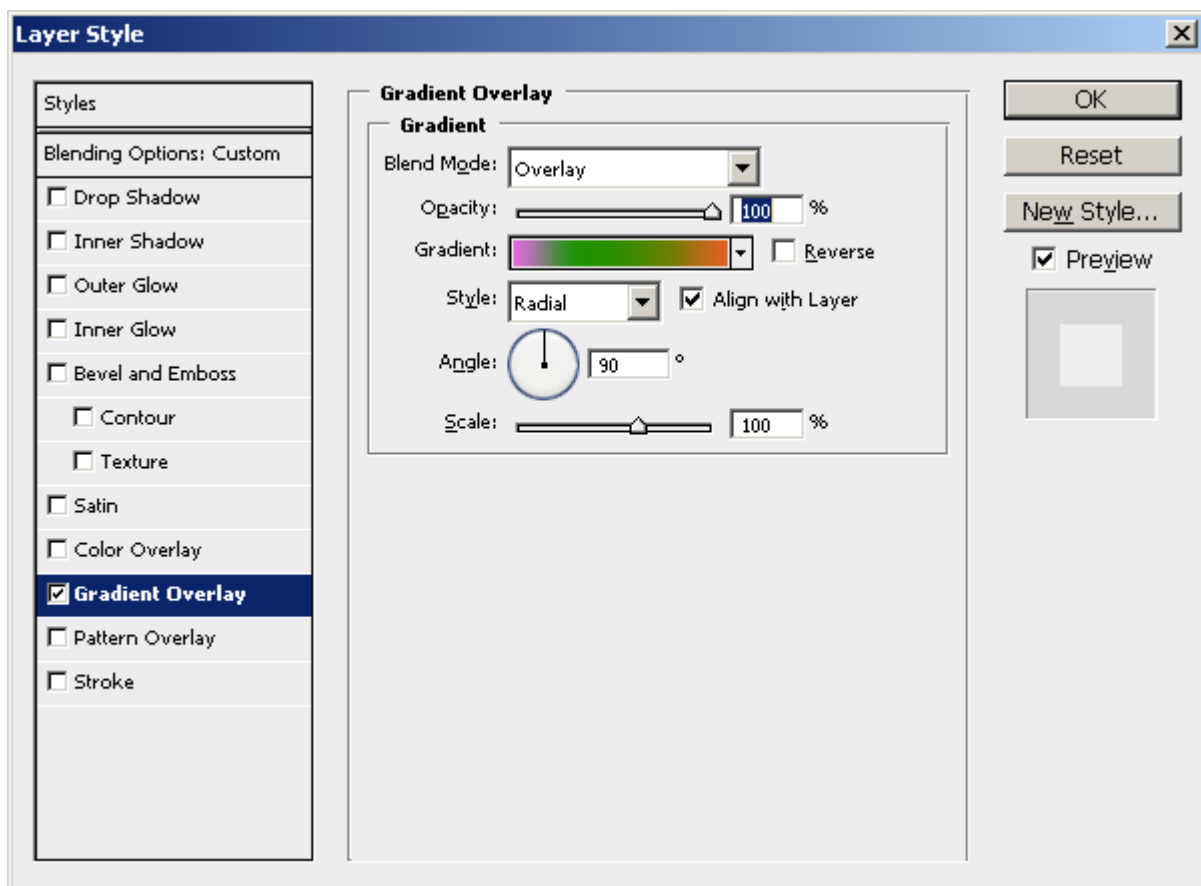
Затем переходим в подменю **Наложение градиента (Gradient overlay)**, и задаем новый градиент. Добавьте третий цвет, для этого нажмите снизу по-центру полосы цвета как на рисунке ниже)



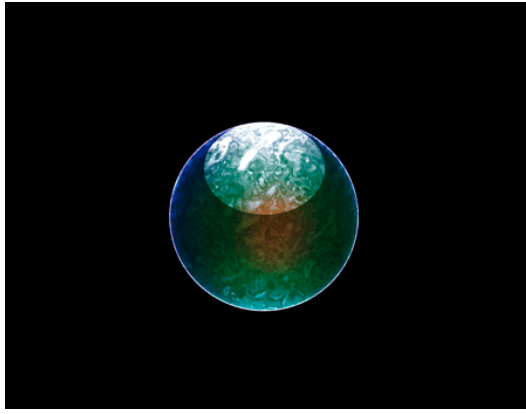
Затем введите в каждый нижний ползунок цвета: фиолетовый, зеленый, оранжевый.



Стиль градиента радиальный, режим смешивания **Overlay**

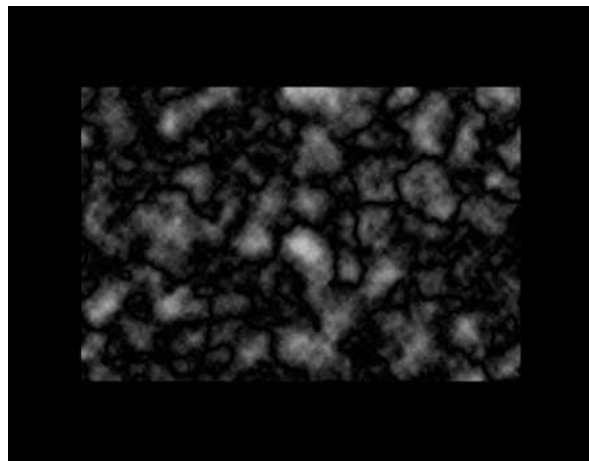


Выделим шар, нажав по иконке главного слоя. Затем инвертируем выделение, нажав **SHIFT+CTRL+I**, и удалим ненужные области вокруг шара (**Delete**)

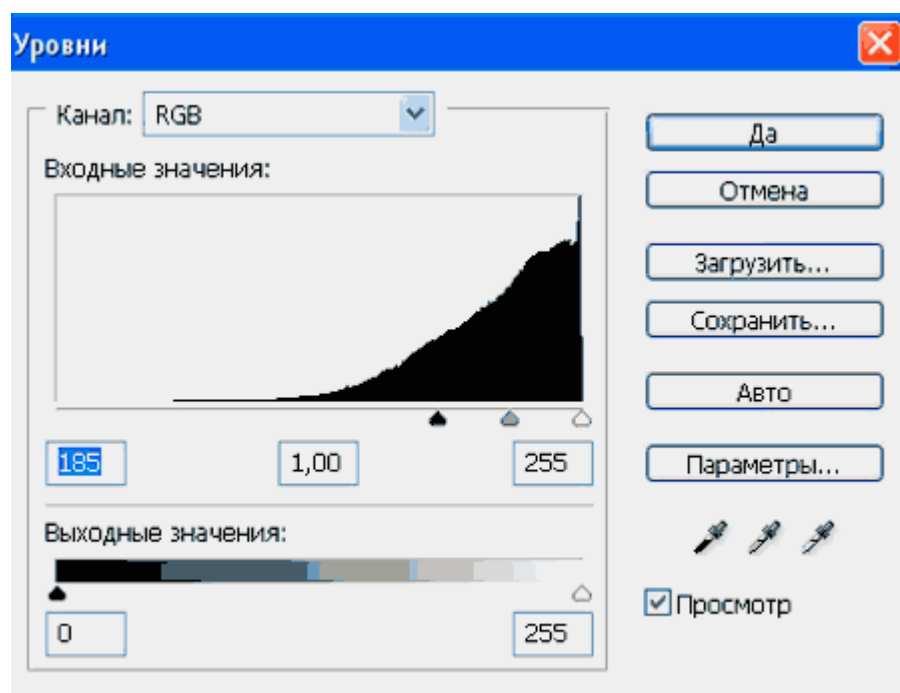


**Шаг 9.** Теперь создадим разряды внутри шара.

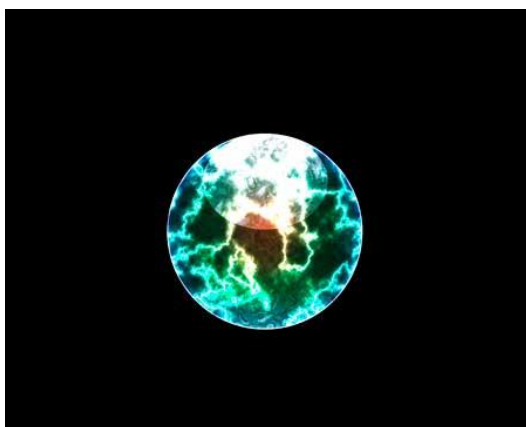
Создадим новый слой и назовем его «Разряды». Применим **Фильтр – Рендеринг – Облака** (Filter – Rendering – Clouds), затем **Фильтр – Рендеринг – Облака с наложением** (Filter – Rendering – Difference Clouds). Затем изменим размер слоя с помощью свободной трансформации, нажав **CTRL+T**. Нужно чтобы слой немного перекрывал шар.



**Шаг 10.** Инвертируем слой, нажав **CTRL+I**. Затем применим команду **Уровни** (Levels). Для этого нажмем **CTRL+L**. Настройки выставьте как на рисунке.



После применения команды измените режим наложения слоя на **Color Dodge**

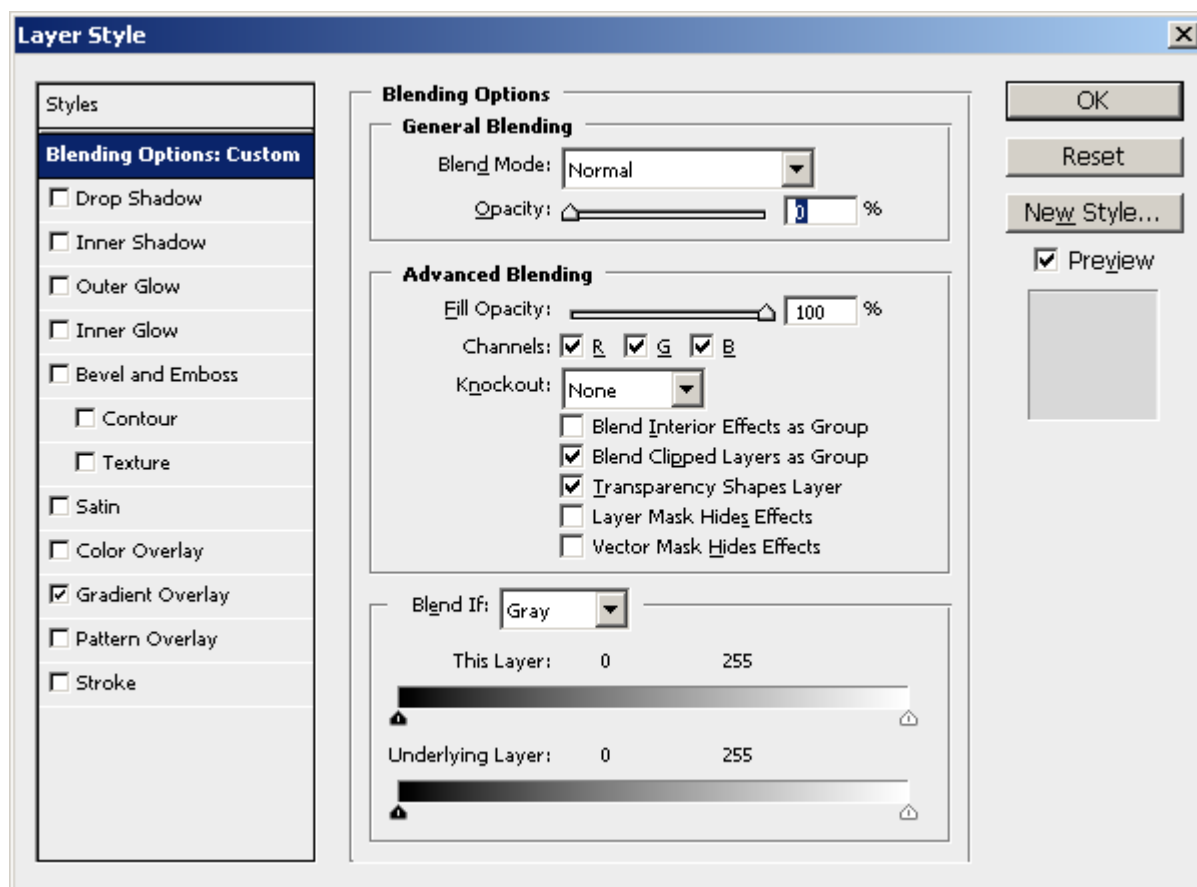


**Шаг 11.** Загрузим фигуру шара как выделение. Применим **Фильтр – Искажение – Сферизация**. (**Filter > Distort > Spherize**)

Инвертируем выделение, нажав **SHIFT+CTRL+I**, и удалим ненужные области на слое (**Delete**). Снова инвертируем выделение.

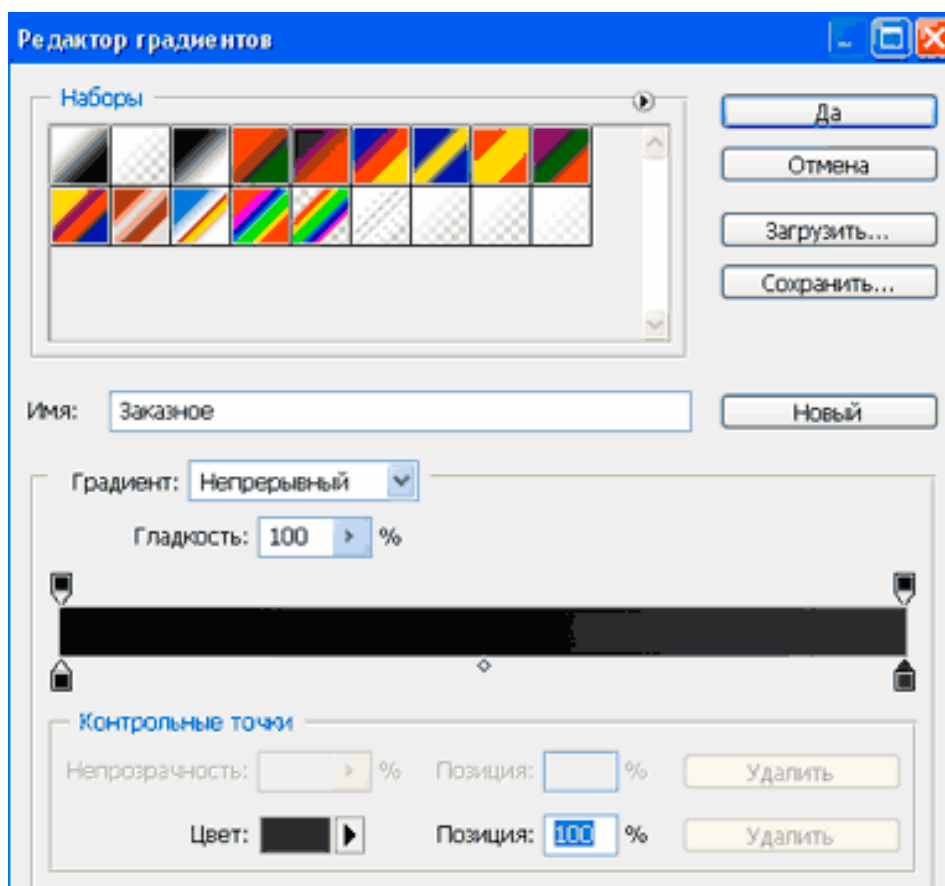
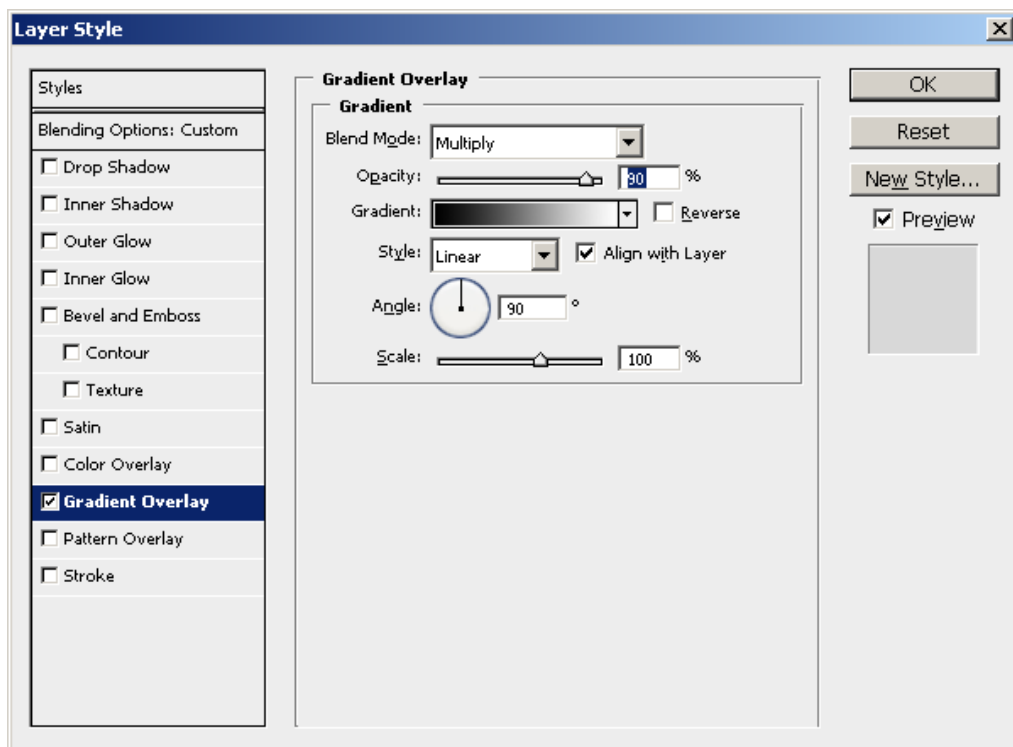
Теперь создадим новый слой выше слоя «**Свечение внизу**» и назовем его «**Настройка цвета**». Выполним заливку выделения черным цветом, нажав **ALT+BACKSPACE**.

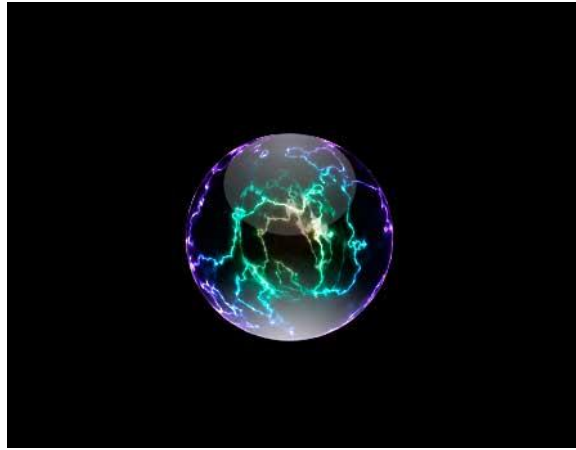
Идем в меню **Стиль слоя (Layer>Layer Style> Blending Options)**, уменьшим непрозрачность заливки до **0%**



Выберем подменю **Gradient Overlay**. Используем режим наложения **Умножение (Multiply)**, непрозрачность **90%**, стиль градиента линейный, цвета от черного (0 0 0) к темно-серому (50 30 30)



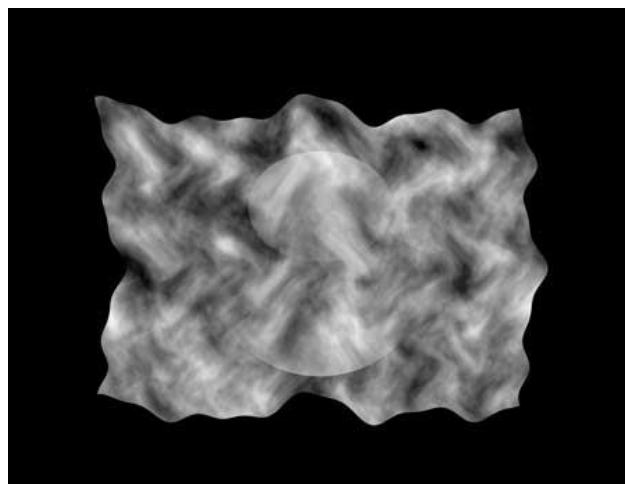
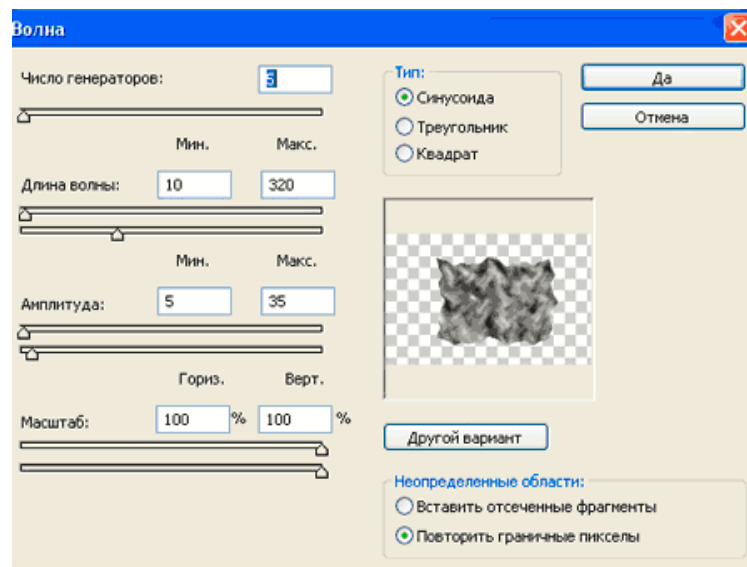




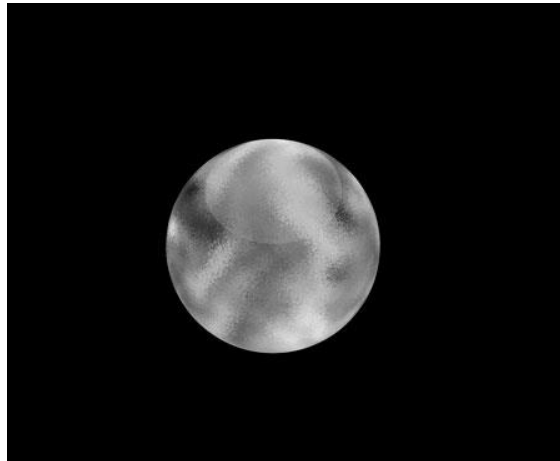
**Шаг 12.** Теперь добавим текстуру на шар.

Создадим новый слой над слоем «**Разряды**» и назовем его «**Текстура**». Не забывайте сверяться с рисунком, где указано расположение слоев. Применим **Фильтр – Рендеринг – Облака** (Filter – Rendering – Clouds).

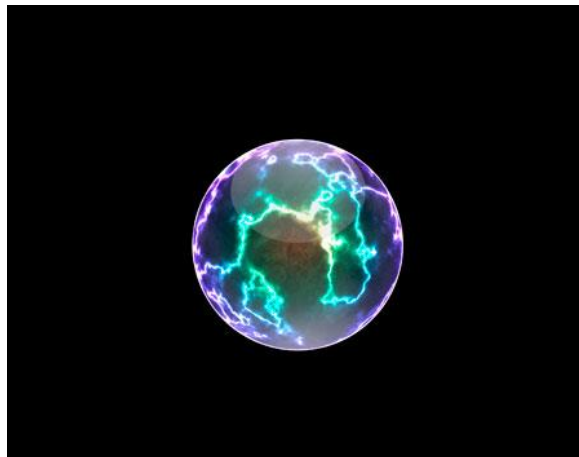
Затем с помощью свободной трансформации (**Ctrl+T**) уменьшим размер слоя, чтобы он слегка перекрывал шар. Применим **Фильтр – Искажение - Волна** (Filter - Distort - Wave) Настройки поставим как на рисунке.



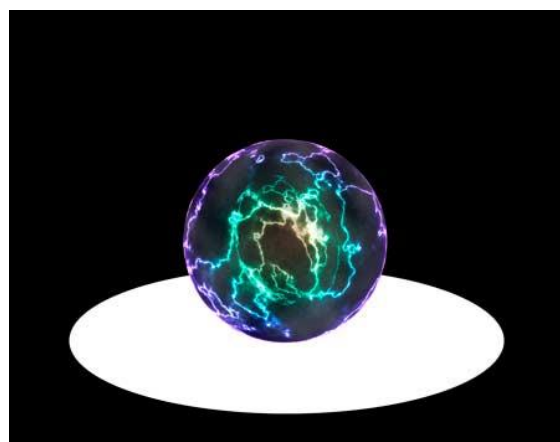
**Шаг 13.** Применим **Фильтр – Штрихи – Разбрызгивание (Filter - Brush Strokes - Spatter)**. Радиус разбрызгивания **15**, смягчение **2**. Затем выделим фигуру шара и применим **Фильтр – Искажение – Сферизация (Filter> Distort> Spherize)** Далее инвертируем выделение (**Ctrl+Shift+T**) и удалим (**Delete**) ненужные области вокруг шара.



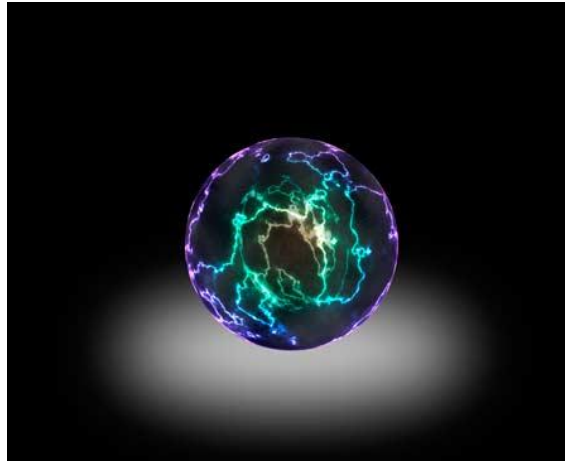
**Шаг 14.** Изменим режим наложения на **Освещение (Screen)**, непрозрачность слоя (**Opacity**) **50%**, непрозрачность заливки (**Fill**) **50%**.



**Шаг 15.** Создадим новый слой над **background**, назовем его «**Свечение под шаром**», сделаем на нем овальное выделение и зальем белым цветом.



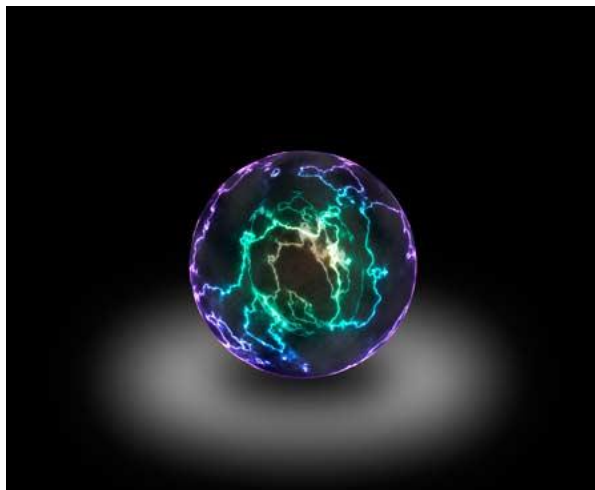
Применим к данному слою **Фильтр – Размытие – Размытие по Гауссу (Filter – Blur – Gaussian Blur)** на **160 px**



**Шаг 16.** Создадим новый слой над предыдущим, назовем его «Тень внизу», сделаем на нем овальное выделение и зальем его черным цветом.

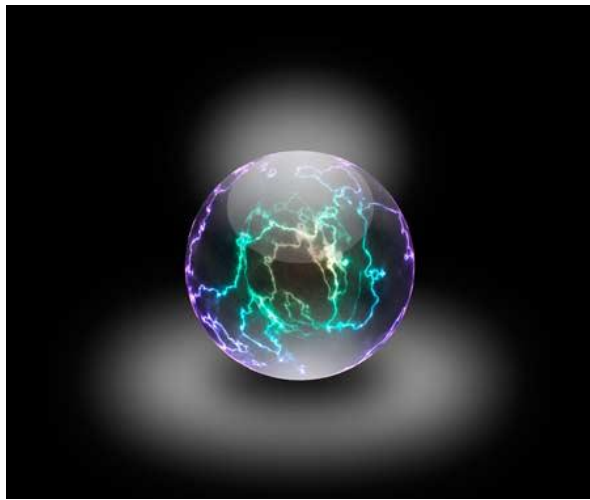


Применим к данному слою **Фильтр – Размытие – Размытие по Гауссу** (Filter – Blur – Gaussian Blur) на **50 px**

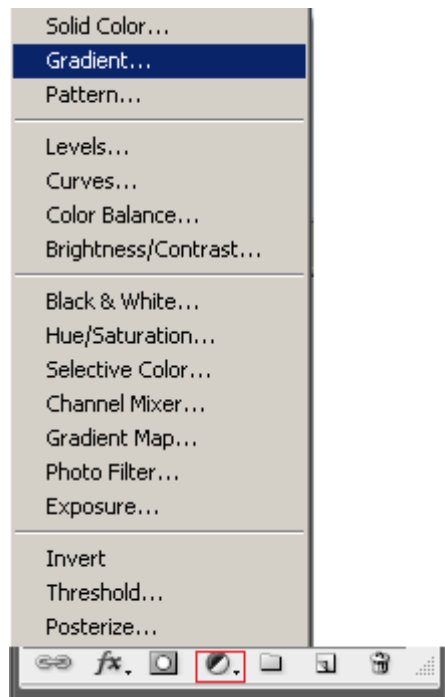


**Шаг 17.** Создадим новый слой над слоем «Текстура». Назовем его «Свечение над шаром». Аналогично создадим овальное выделение, зальем белым цветом. Снимим выделение **Ctrl+D**.

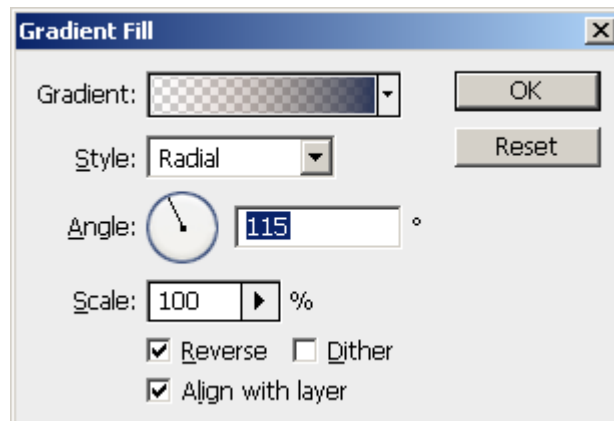
Выполним размытие по гауссу на 125 p. Уменьшите непрозрачность (**Opacity**) слоя до 90 %.

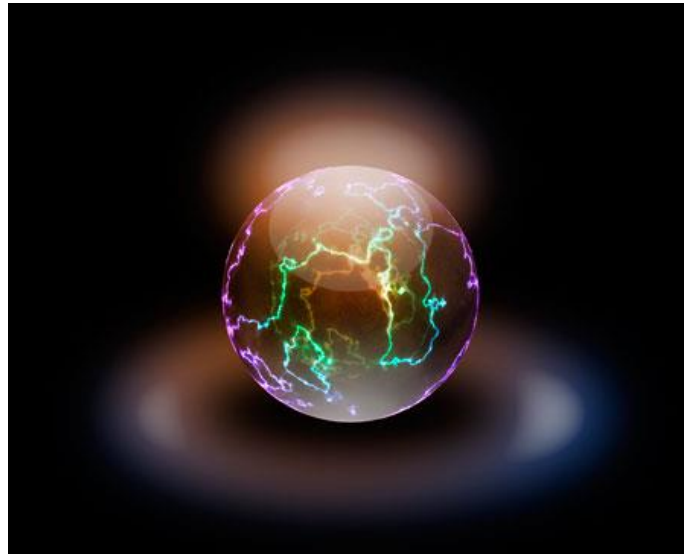
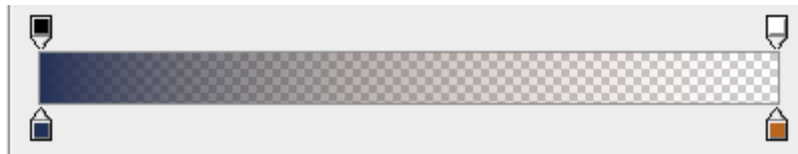


**Шаг 18.** Создадим новый корректирующий слой «Gradient».



Для градиента используем цвета **34 49 88** и **184 100 28**. Стиль градиента радиальный, угол в пределах **115-120°**. Режим наложения изменим на **Перекрытие (Overlay)**.

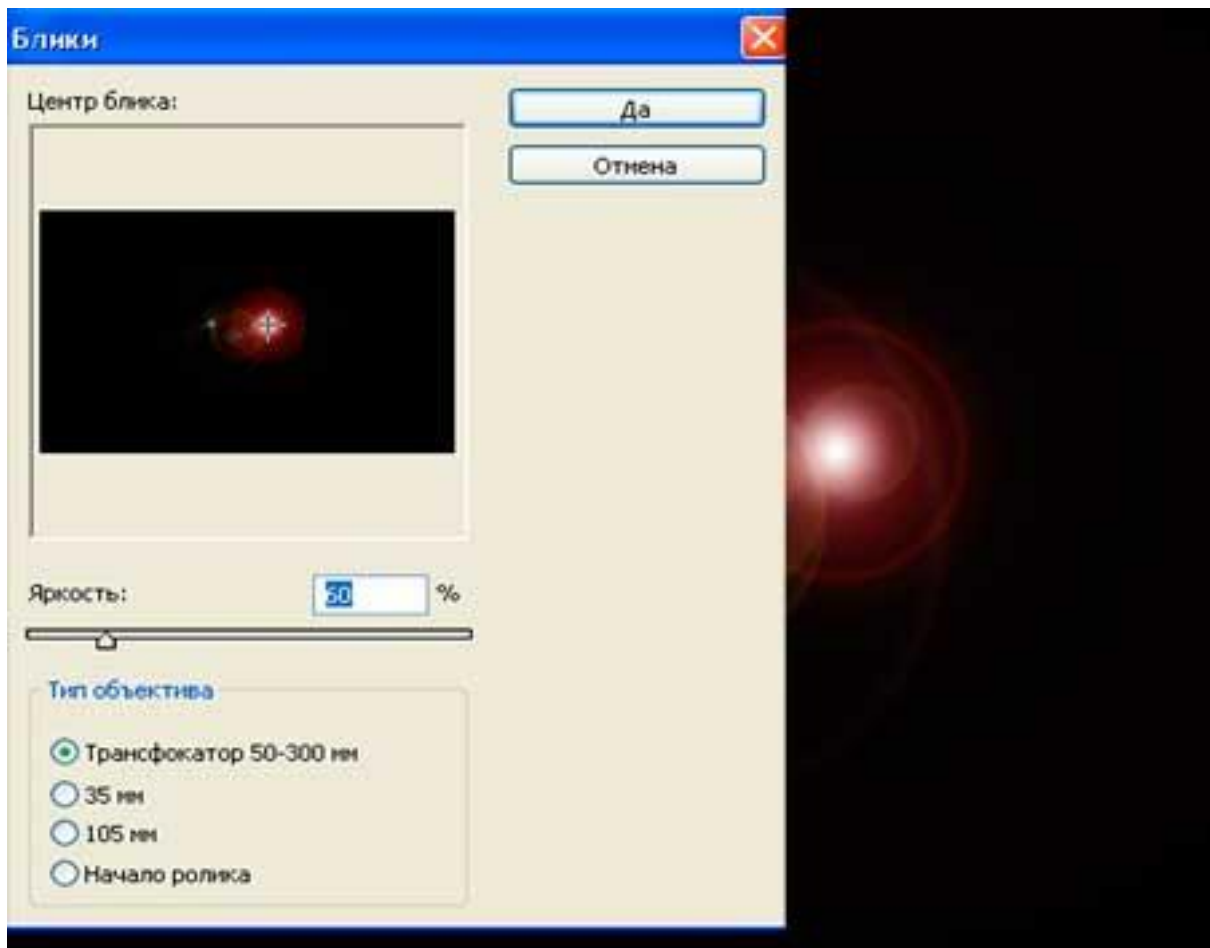




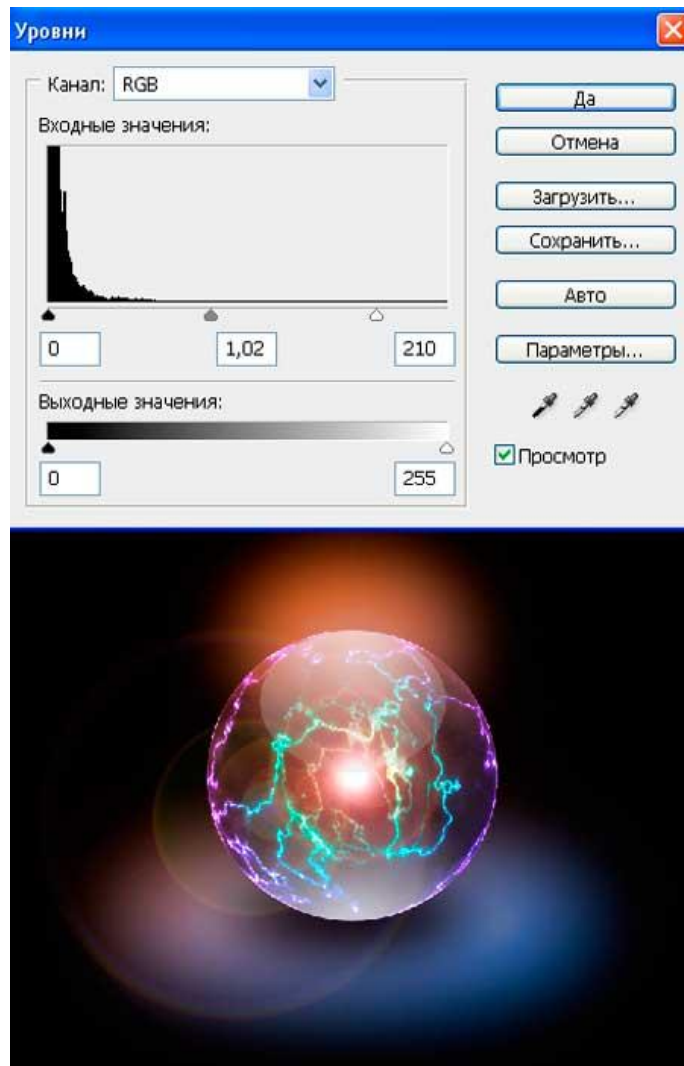
**Шаг 19.** Создадим новый слой выше остальных и назовем его «**Внутренний свет**». Зальем слой черным. (**Выделить весь (Ctrl+A)** и **залить выделение черным**)

Изменим режим наложения на **Lighten**.

Применим **Фильтр – Рендеринг – Блик** (Filter -Render - Lens Flare).



Теперь применим команду **Уровни (Ctrl+L)**, чтобы увеличить свечение.



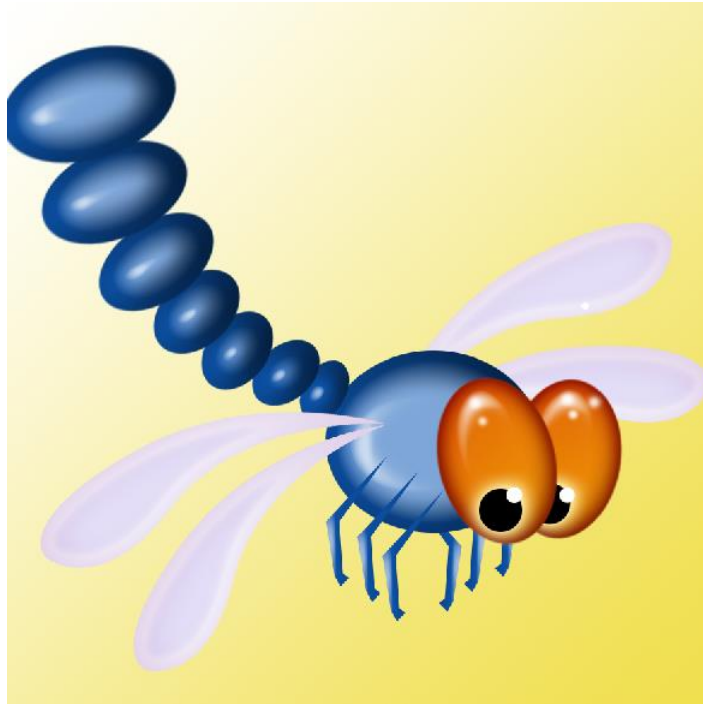
## Практическая работа №36 (ПР36)

**Тема:** Растровый графический редактор Photoshop

**Цель:** Создание стрекозы с помощью графических инструментов в Adobe Photoshop

### Стрекоза

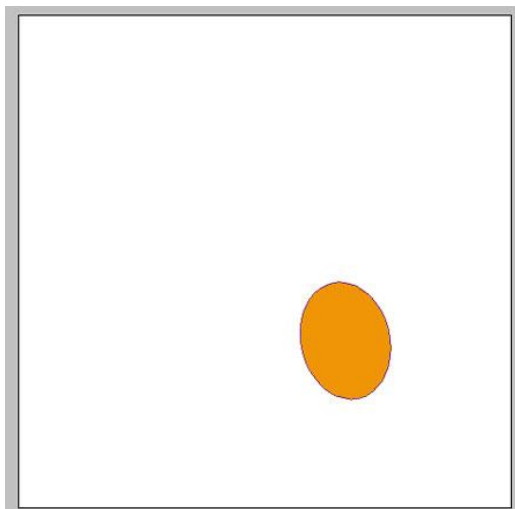
В этом уроке при помощи инструментов Перо (Pen Tool), Эллипс (Ellipse Tool) и Стилей Слоя (Layer Style) необходимо нарисовать такую стрекозу!



1) Необходимо создать Новый документ (File-New) 900x900px, RGB , 72dpi.

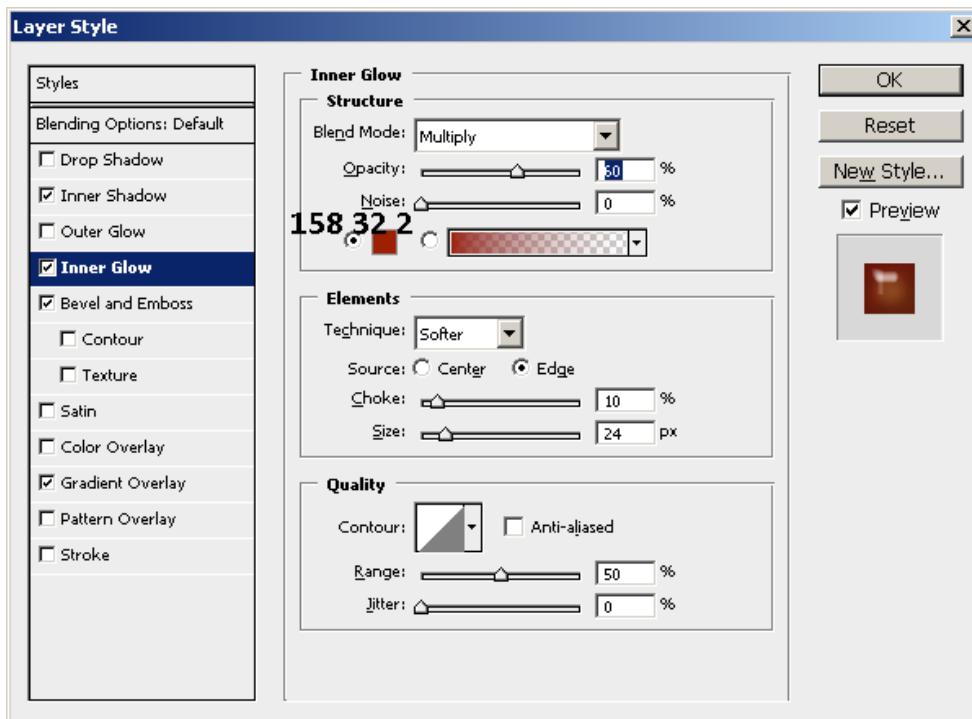
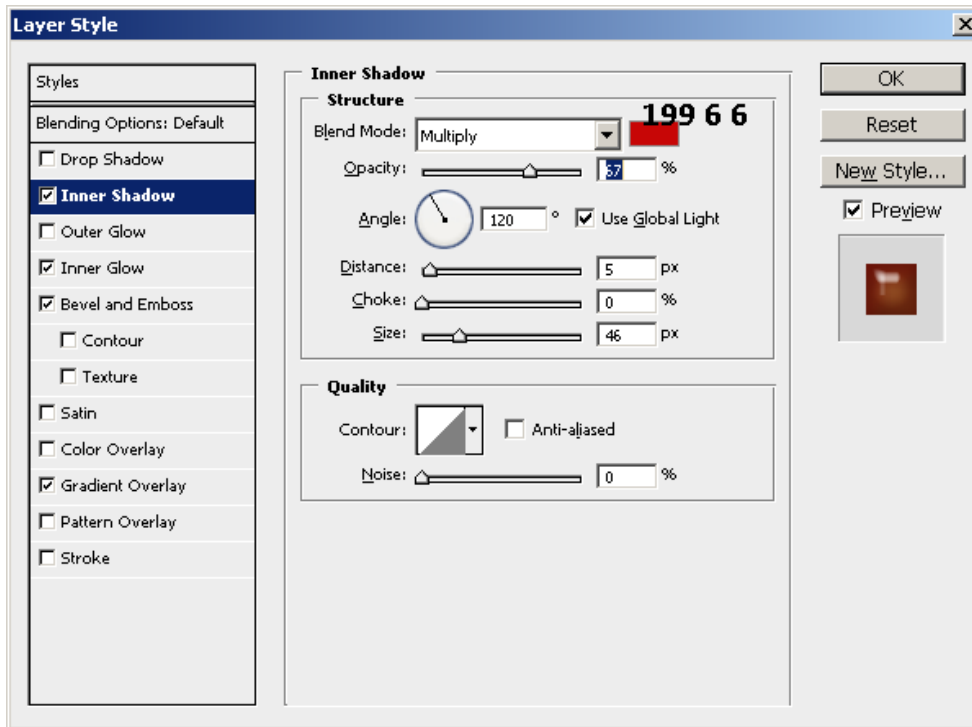
2) Затем выбираем инструмент выделения эллипс (Elliptical Marquee Tool). Рисуем овал. Затем используя свободное трансформирование (Ctrl+T) поворачиваем выделение левее за левый верхний угол.

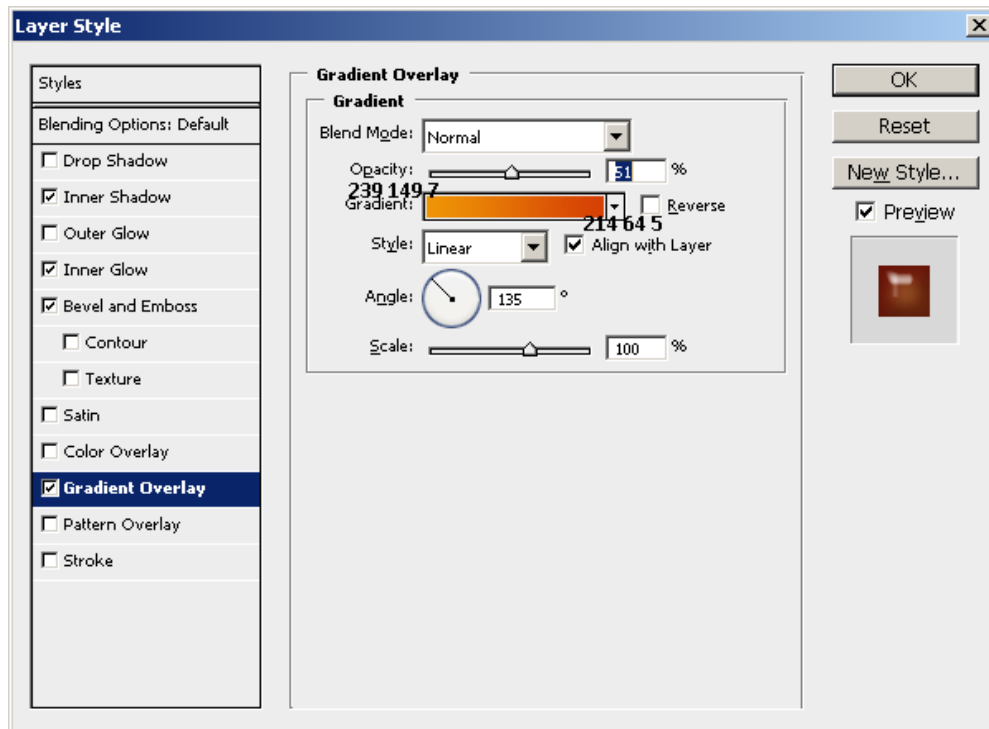
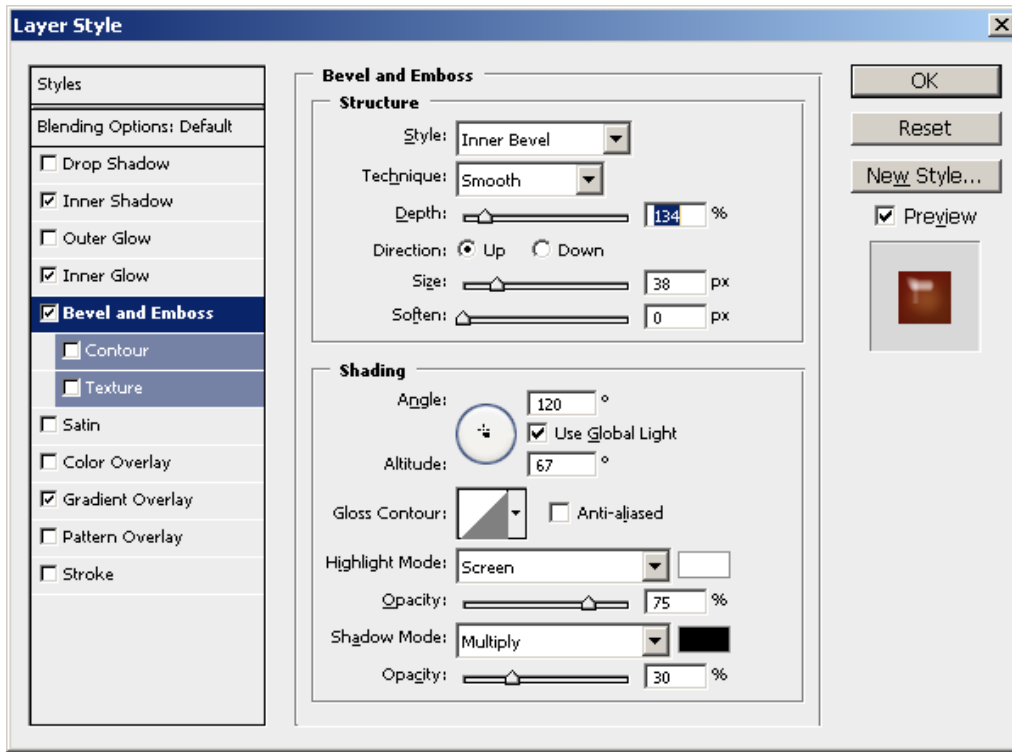
Заливаем овал цветом 239 149 7.



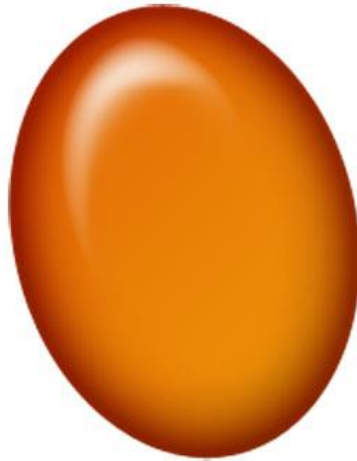
Настраиваем стили слоя (**Layer> Layer Style> Blending Options**):





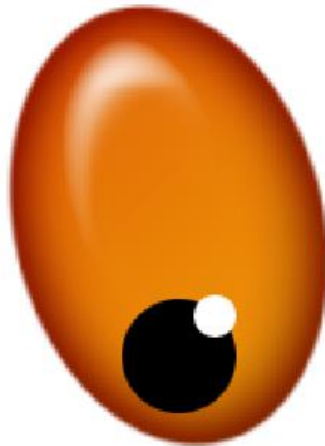


Получается такой глазик.

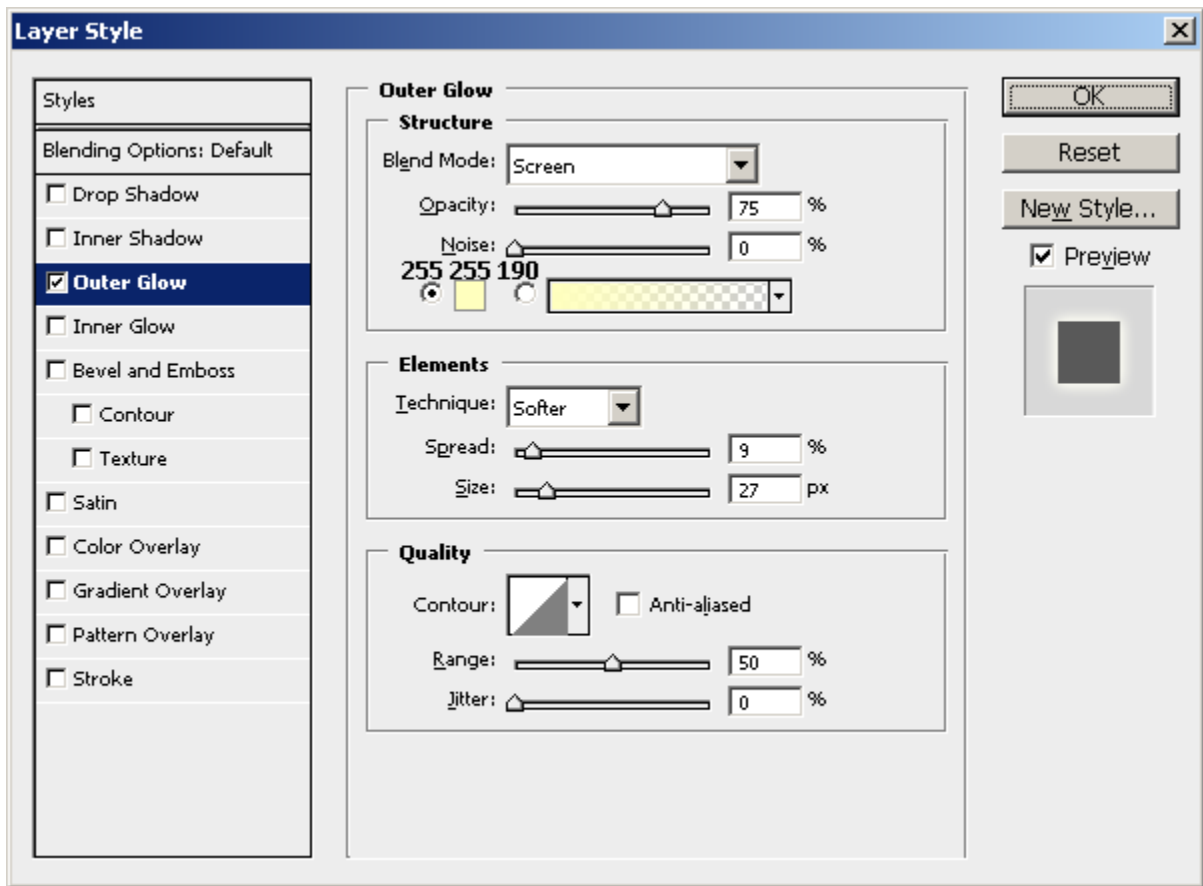


3) Создаем новый слой. Этот слой будет называться – Зрачок.

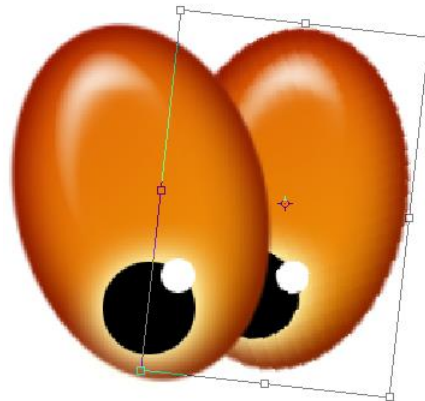
При помощи того же инструмента выделения - эллипс (Elliptical Marquee Tool) рисуем круг поменьше и заливаем черным. Затем рисуем блик зрачка еще меньшим радиусом



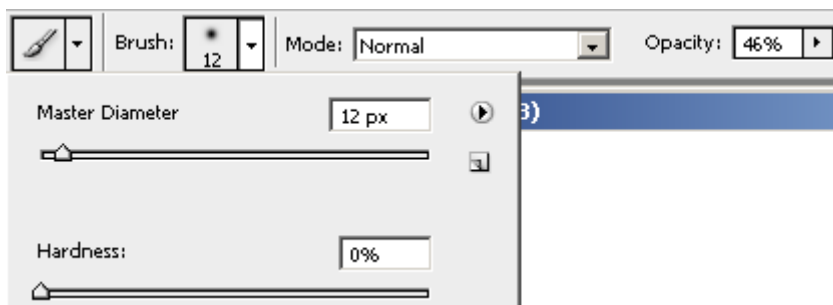
Применяем Стили слоя (**Layer > Layer Style > Blending Options**):



4) Сливаем слои с глазом и зрачком (Выделяем оба и Ctrl+E). Дублируем полученный слой. И помещаем дубликат под оригинальный слот, немного уменьшив как на рисунке ниже:



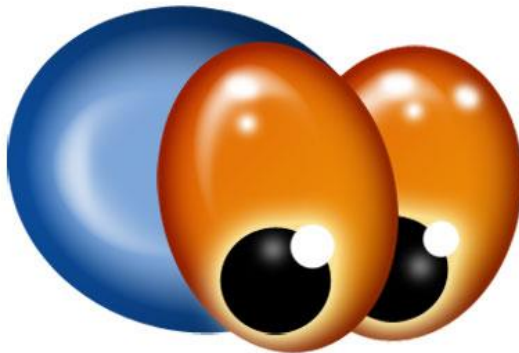
На отдельном слое рисуем, мягкой белой кистью, блики (размером 12 и 19 px) (можете подобрать по-своему)



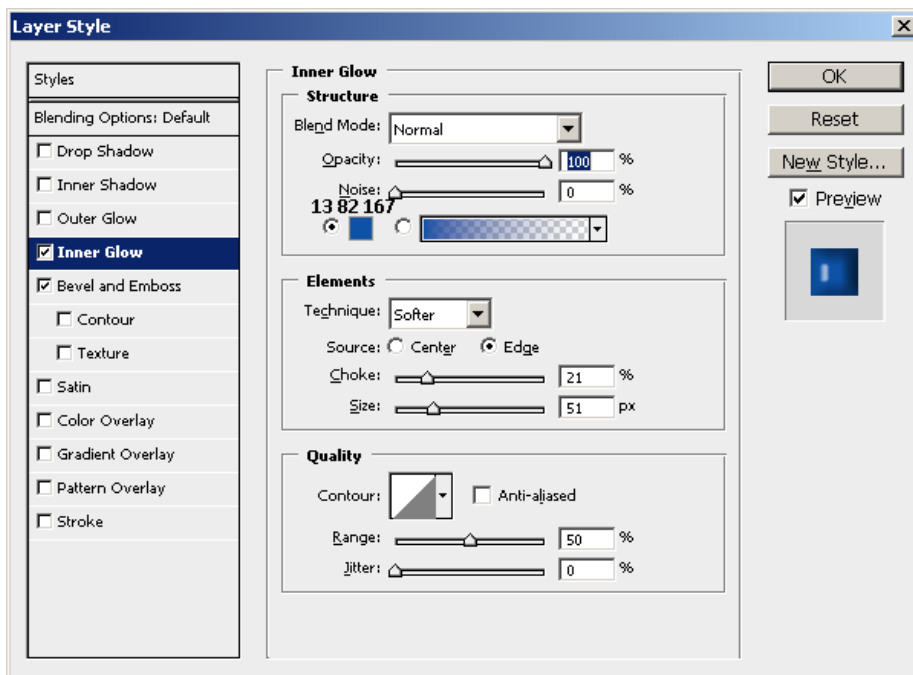


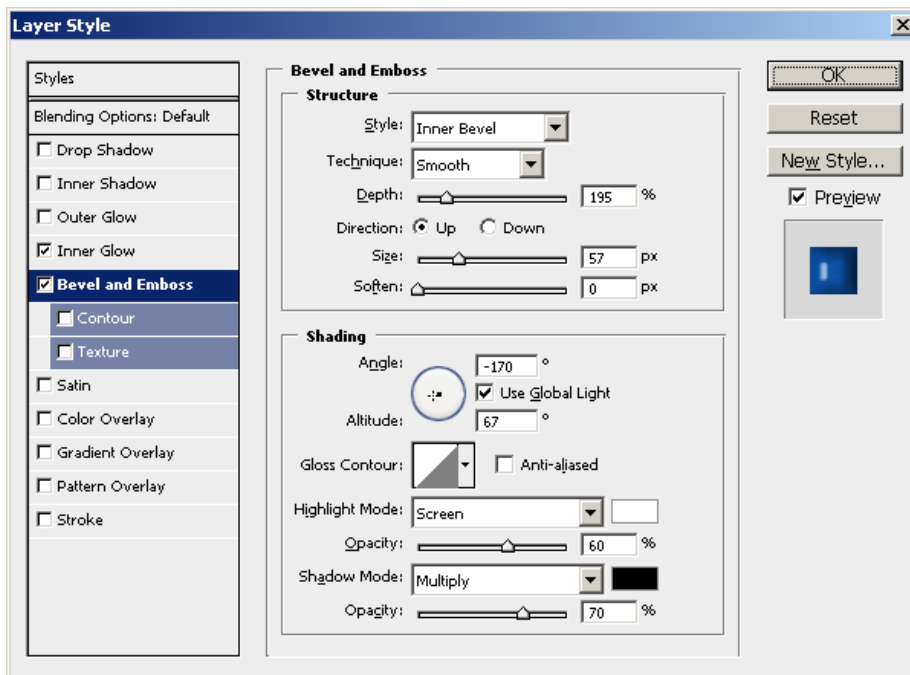
Сливаем все слои с глазками.

5) Создаем новый слой под слоем с глазами. Опять выбираем Эллипс (Elliptical Marquee Tool) и рисуем круг цветом **126 166 216**, располагая его под Глазами.

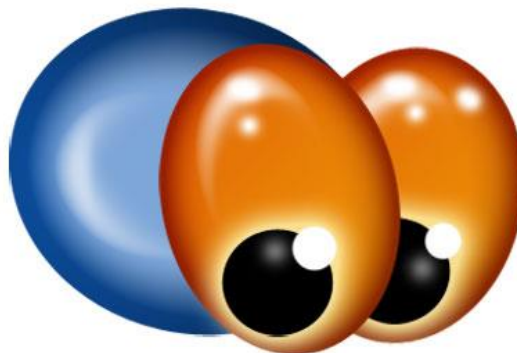


Применяем данные настройки Стилей Слоя (**Layer > Layer Style > Blending Options**):



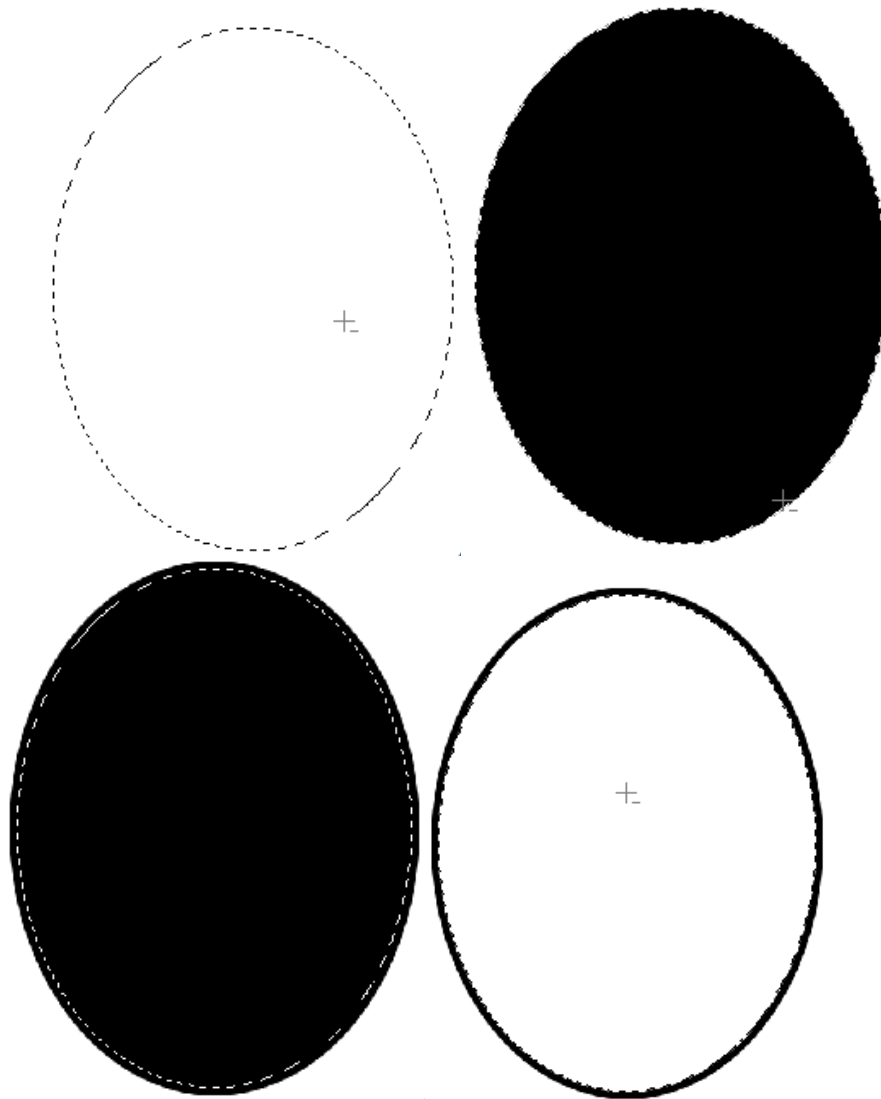


Получается тельце стрекозы.



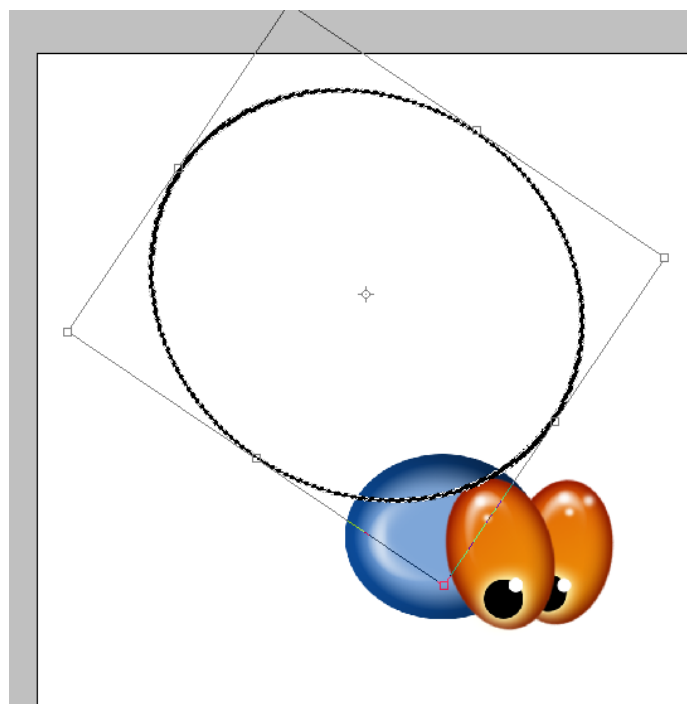
6) Необходимо нарисовать хвост стрекозы, для начала нужно построить дугу.

Для этого с помощью круглого выделения (**Elliptical Marquee Tool**) строим большой овал и заливаем его черным цветом (**правая кнопка мыши > Fill**). Затем жмем **Select > Modify > Contract** и устанавливаем 5px. Выделение сожмется внутрь. Нажмите **Delete** чтоб очистить лишнее.

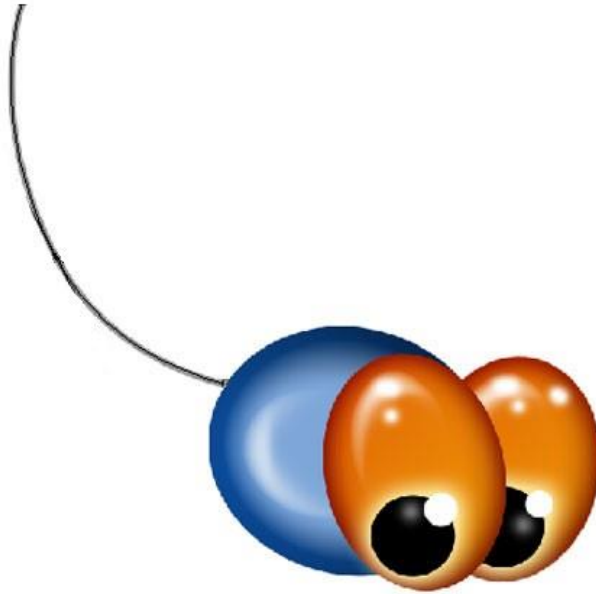


Кликните по иконке текущего слоя, чтобы выделить полностью контур окружности.

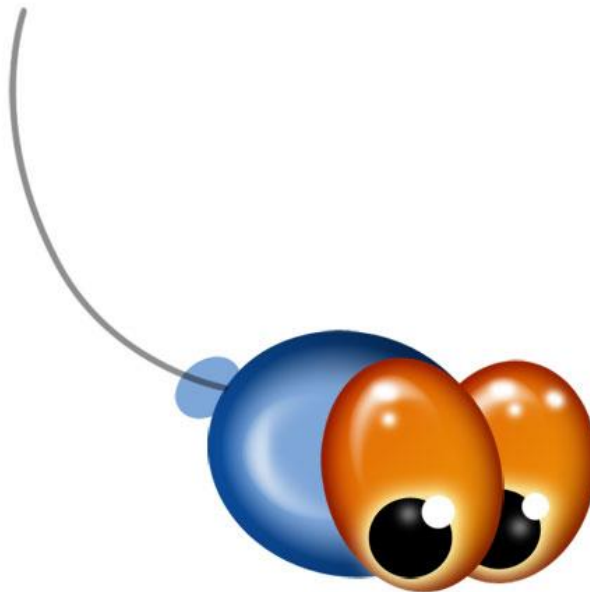
Используя «Свободное трансформирование» преобразуйте дугу (**Ctrl + T**) как показано на рисунке ниже



Примените трансформацию, нажав галочку в верхнем правом углу. Затем с помощью стиралки (**Eraser Tool (E)**) сотрите лишнее как на рисунке ниже, либо для этого можете использовать прямоугольное выделение (**Rectangle Marquee Tool (M)**) и кнопку (**Delete**).

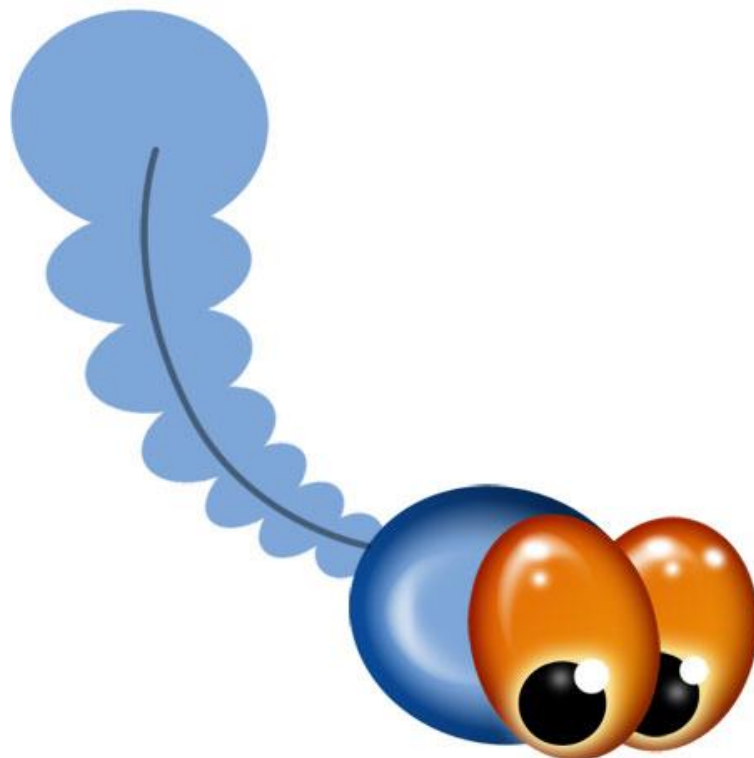


**7) Создаем новый слой** выше всех остальных. Выбираем инструмент Эллипс (Elliptical Marquee Tool) и создаём маленький овал того же цвета **126 166 216**. Ctrl+T – поворачиваем его. Этот слой располагаем под Телом стрекозы используя **Move Tool (V)**



Рисуем хвост у стрекозы следующим образом. Каждый овал должен быть на отдельном новом слое

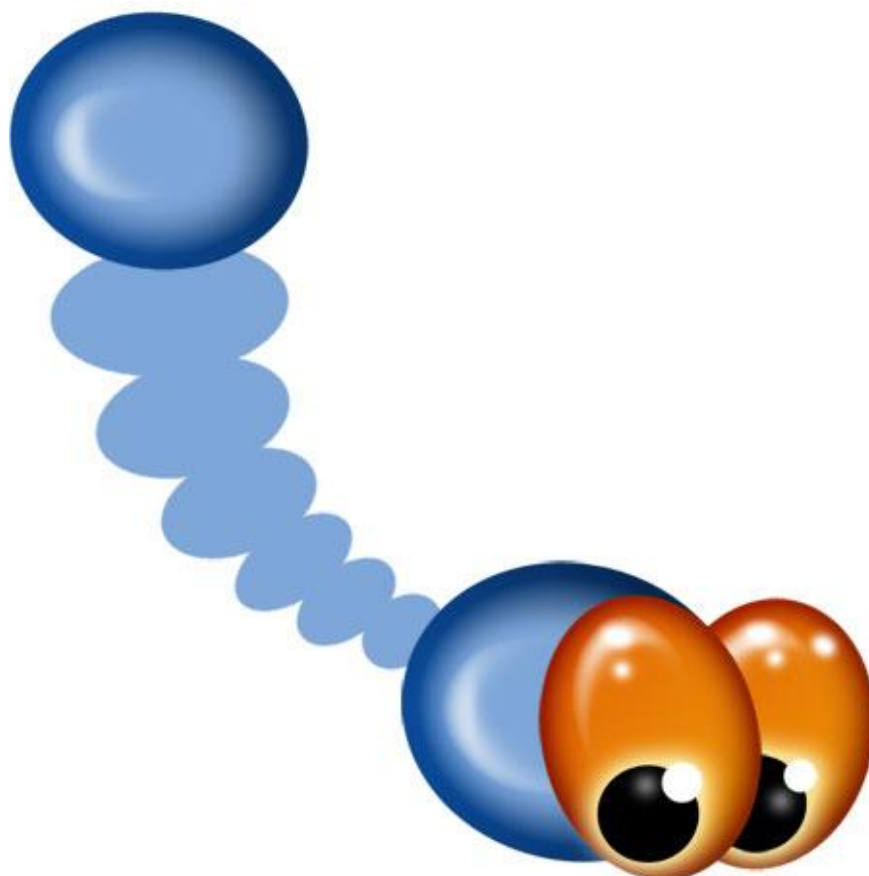




Слой с направляющей линией (дугой) можно скрыть (убираем глазок около слоя).

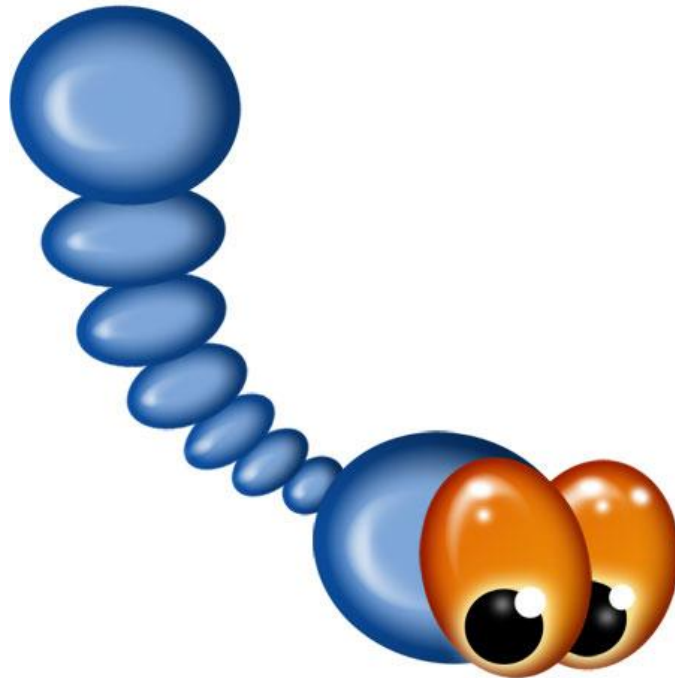
К самому последнему, большому, кругу хвоста применяем стили слоя Тельца стрекозы:

Находясь на слое **Тельце**, кликаем ПКМ и в выпадающем меню выбираем Копировать Стиль слоя (**Copy Layer Style**). Переходим на слой с самым большим кружком, опять кликаем ПКМ и в выпадающем меню выбираем Вставить Стили слоя (**Paste Layer Style**).

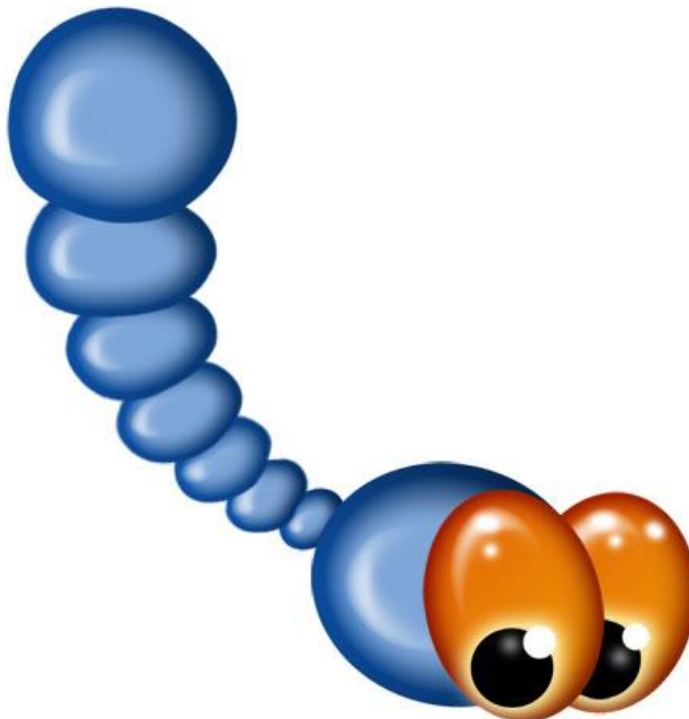


Далее применяем эти стили к каждому членику хвоста.

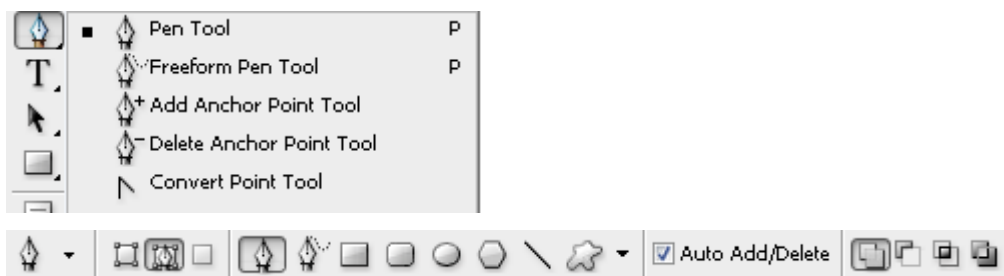
Заметно, что на маленьких деталях тени смотрятся грубовато. Можете для исправления ситуации, зайти в Стили на каждом слое (ПКМ на слое> **Blending Options**) и уменьшить постепенно размер эффектов. Я уменьшал значения (**Inner Glow**): **Size** и **Choke**, а для (**Bevel and Emboss**): **Size** и **Depth**. Экспериментируйте!



Но такие правильные овалчики мне не нравятся, и я немного поменял каждый из них, с помощью Деформации (Edit-Transform-Warp).



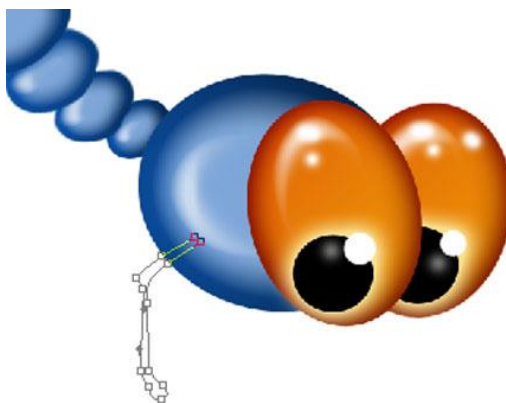
8) **Создайте новый слой поверх всех.** Теперь, используя инструмент Перо (Pen) рисуем маленькую стрекозину ножку.



По очереди ставьте точки нажатием левой кнопки мыши, таким образом образуя путь (не удерживайте левую кнопку мыши).

Если ошиблись сделайте шаг назад (Ctrl+Alt+Z)

Полностью закончите путь в начальной точке, таким образом будет сформированная фигура по контуру.

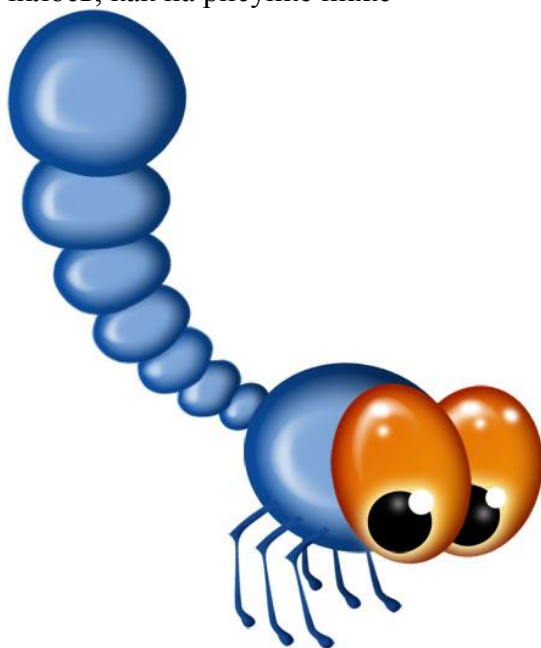


Нажмите на полученный контур правой кнопкой мыши и используя пункт Fill Path залейте ножку цветом 21 75 142. После заливки потребуется удалить контур пути ножки, нажав **ПКМ по контуру > Delete Path**

Дублируем слой два раза и передвигаем ножки на их место – MoveTool(V). Чтобы образовать 3 ножки рядом как на рисунке ниже. Можно их немного деформировать по своему вкусу (Ctrl+T).

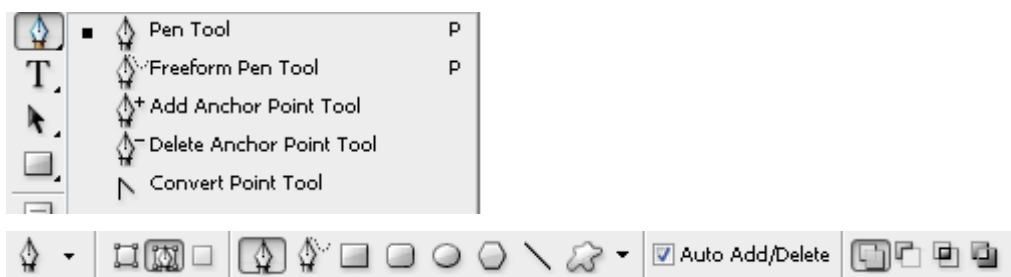
Эти три ножки (слоя) сливаем в один слой (**выделяем все три слоя и Ctrl+E**). Копируем слой от главного тельца (**ПКМ > Copy Layer Style**) и вставляем стиль в слитый слой с ножками (**ПКМ > Paste Layer Style**)

Дублируем слой с 3-мя ножками. Сдублированный слой с ножками перемещаем под слой с Тельцем, чтобы получилось, как на рисунке ниже

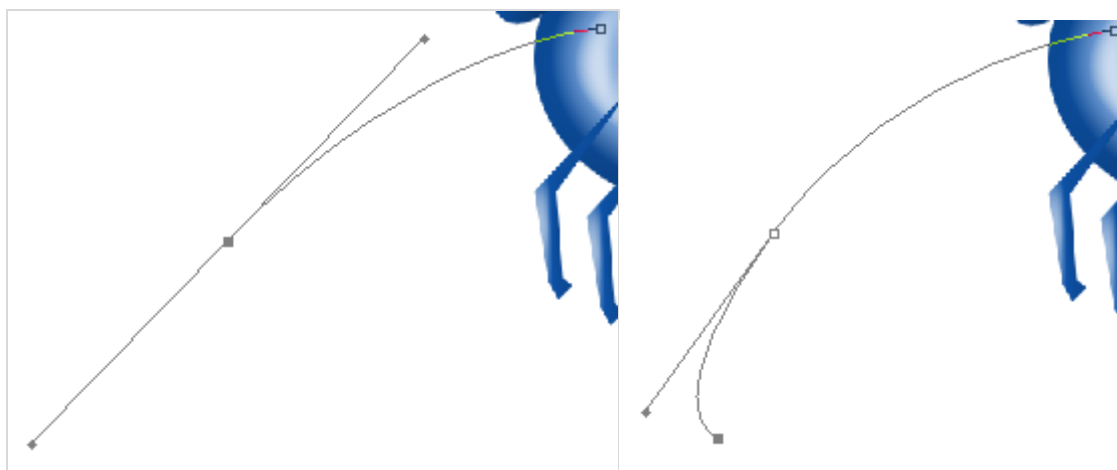


### 9) Создайте новый слой.

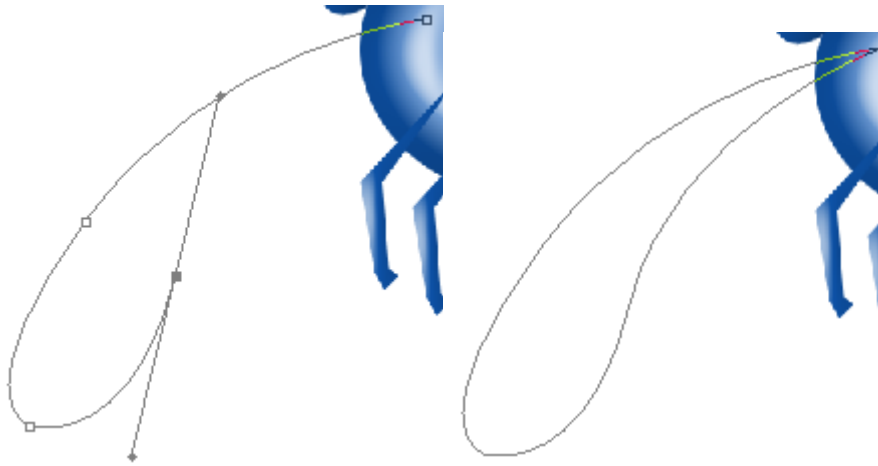
Далее, инструментом **Перо (Pen)** рисуем крыло стрекозы.



Поставьте первую точку, на тельце. Затем поставьте вторую точку и не отпуская кнопку мыши протягиваем мышку дальше, делая отрезок плавно-кривым.



Продолжаем аналогично. Если ошиблись сделайте шаг назад (Ctrl+Alt+Z)

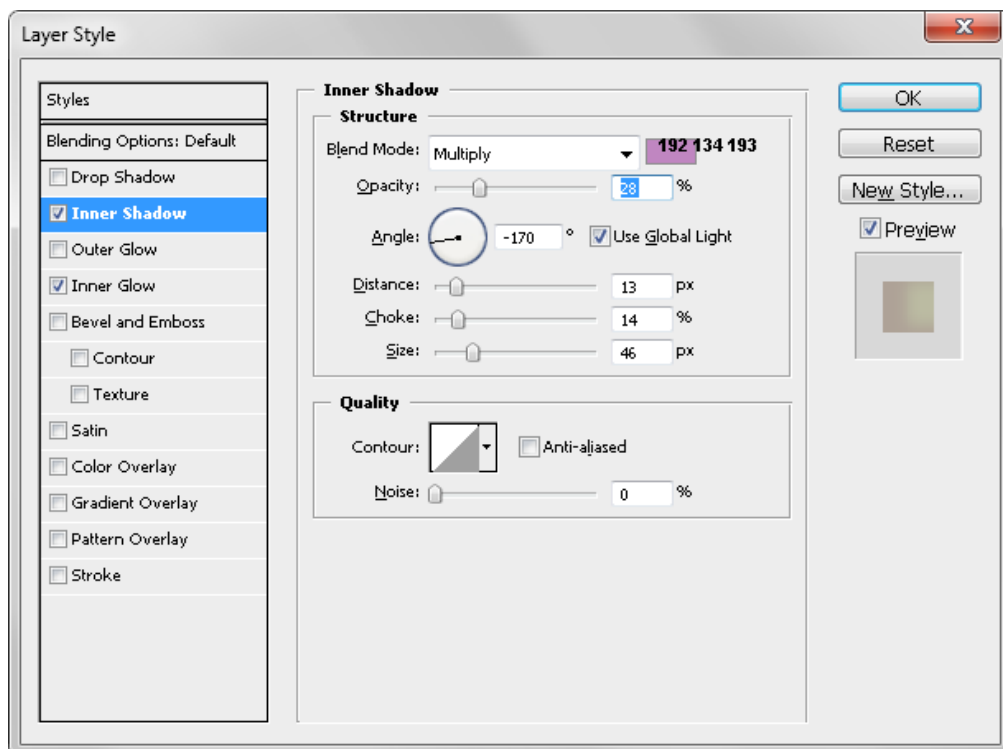


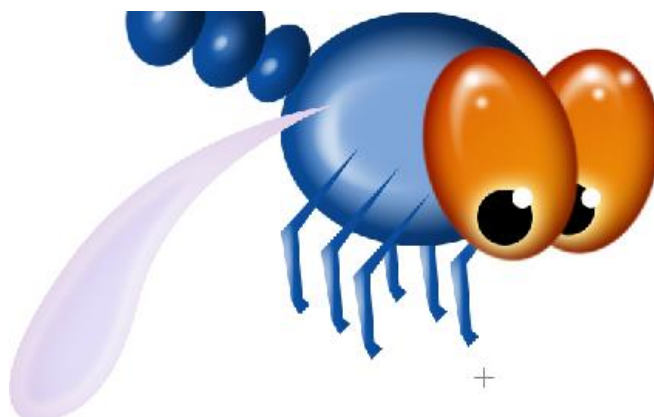
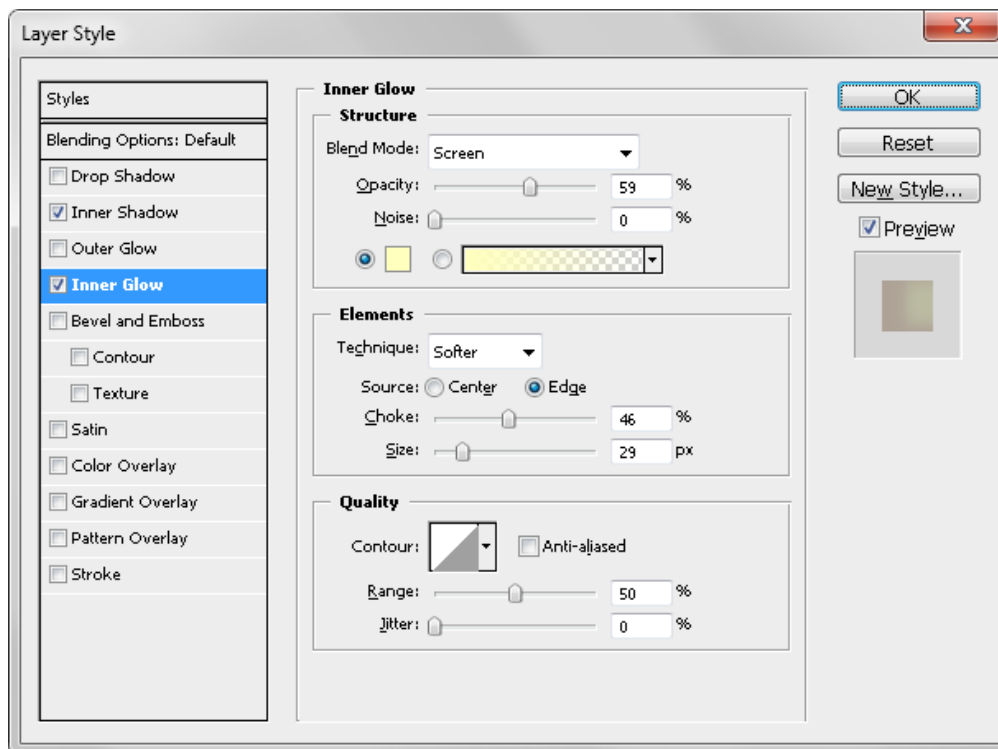
Залейте крыло светло-фиолетовым цветом (227 221 249) (ПКМ по контуру > Fill Path).

После заливки потребуется удалить контур пути ножки, нажав ПКМ по контуру >

### Delete Path

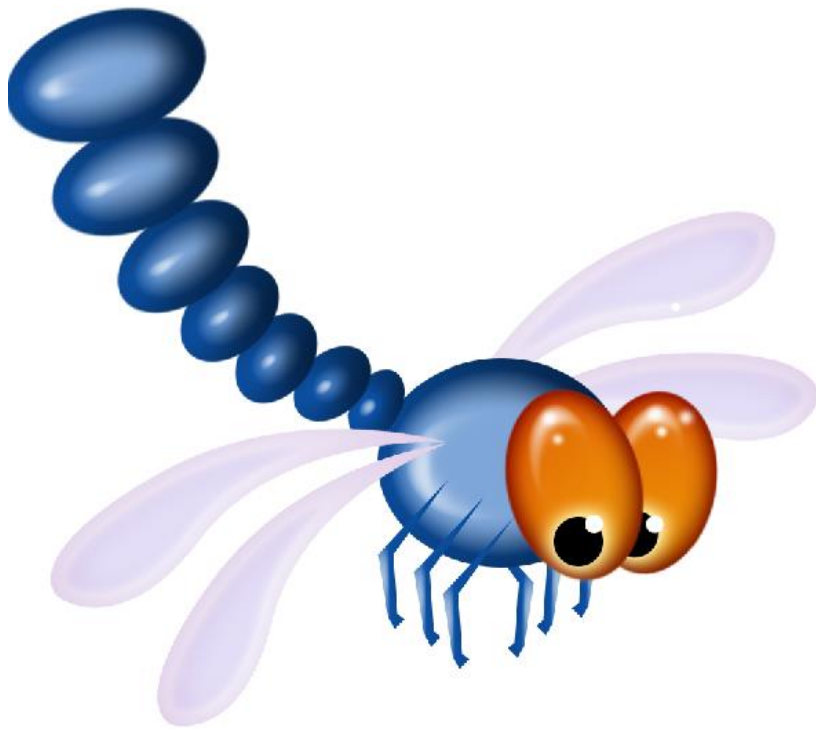
Применяем к слою с крылом стиль (Layer > Layer Style > Blending Options):





Дублируем слой с крылом и помещаем его под слой Тельца (под 3-ий слой снизу).  
Отразите слой по горизонтали (**Edit > Transform > Flip Horizontal**)/

С помощью свободной трансформации (**Ctrl+T**) приведите вторые крылья к виду как на рисунке ниже:



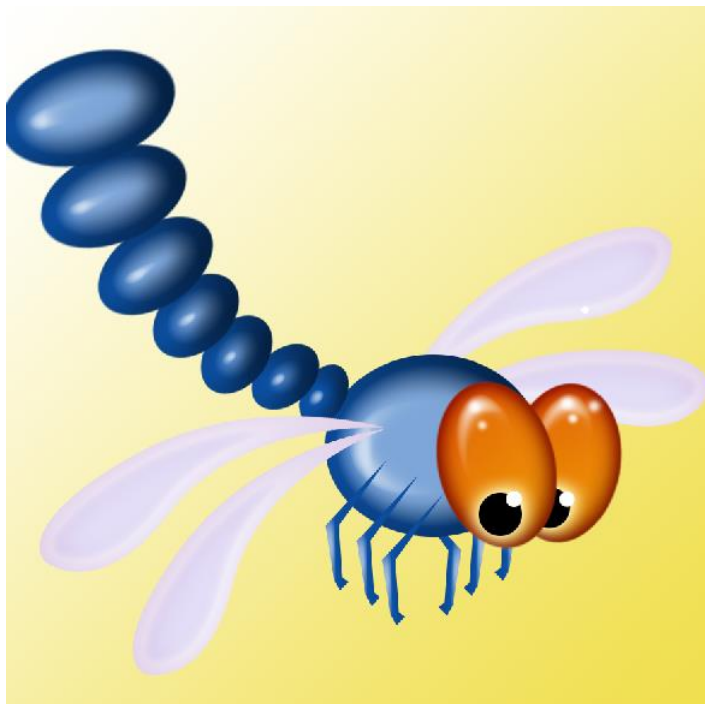
10) Выделите все слои удерживая **Ctrl** кроме **Background** и с помощью инструмента **Move Tool (V)** переместите стрекозу по центру.

11) Осталось добавить цвет фона. Выделите слой **Background**. Задайте градиент (**Gradient Tool**):

**Первый цвет:** 240 223 76

**Второй цвет:** 255 255 255

Проведите градиент из нижнего правого края листа к верхнему левому. **Вот и все.**



Можете что-то изменить или добавить на ваше усмотрение...

## Практическая работа №37 (ПР37)

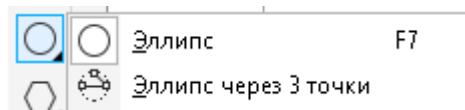
Тема: Векторный графический редактор CorelDraw

Цель: Создание разноцветного надувного шара в CorelDraw



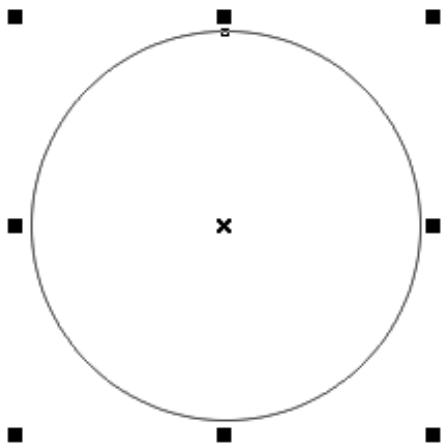
1) Создайте чистый лист с любыми параметрами (**Файл > Создать**)

Вначале просто создадим окружность: для этого выбираем инструмент **Эллипс (F7)**



**Внимание:** чтобы отобразить дополнительные возможности инструмента, необходимо удерживать на нем левую кнопку мыши несколько секунд.

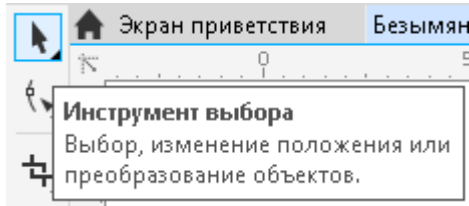
Рисуем эллипс, удерживая клавишу **Ctrl** (чтобы нарисовать **круг** с одинаковыми размерам сторон)



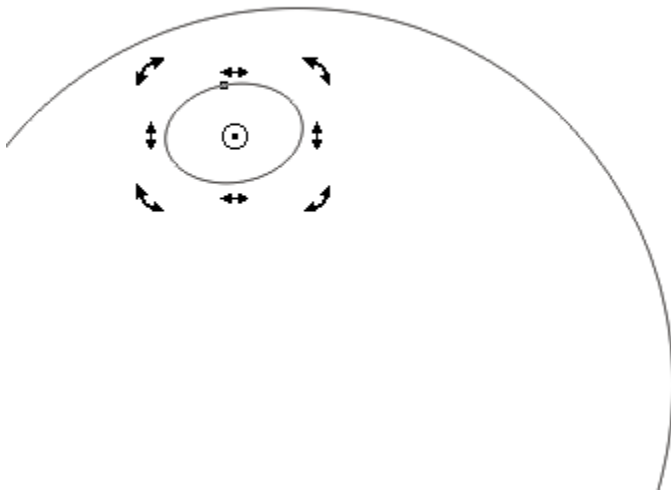


2) Затем на этой окружности создаём **эллипс**: вначале рисуем.

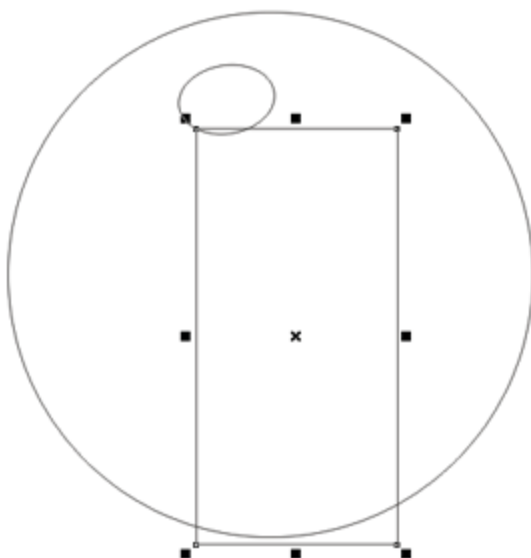
А затем используя **инструмент выбора** перетащим круг в нужное место



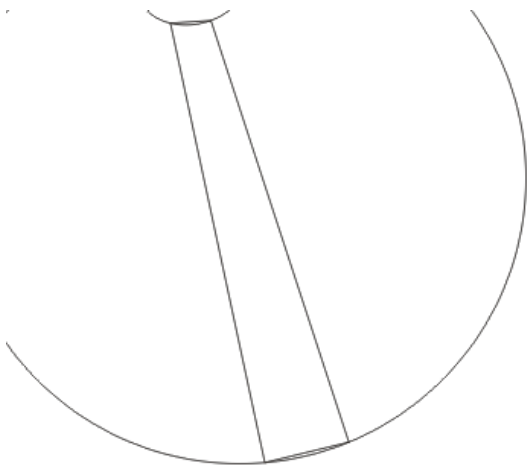
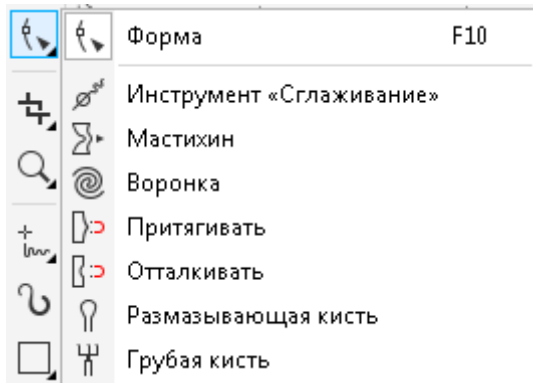
Нажав на центральный крестик, чтобы появились стрелки управления, поворачиваем:



3) Для создания полосок на шаре будем использовать **Прямоугольники (F6)**. Рисуем прямоугольник в том месте, где мы хотим нарисовать полоску:

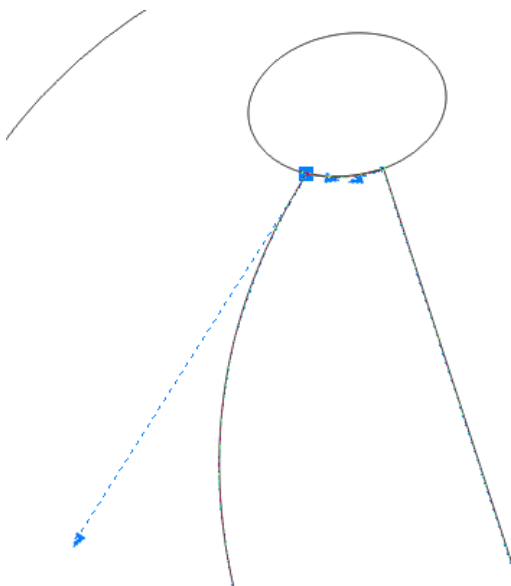


Теперь воспользуемся небольшим трюком: сначала переведём наш прямоугольник в линии **Преобразовать в кривую** (Ctrl+Q), а затем с помощью инструмента **Форма** (F10) и подведём углы к границам окружностей:

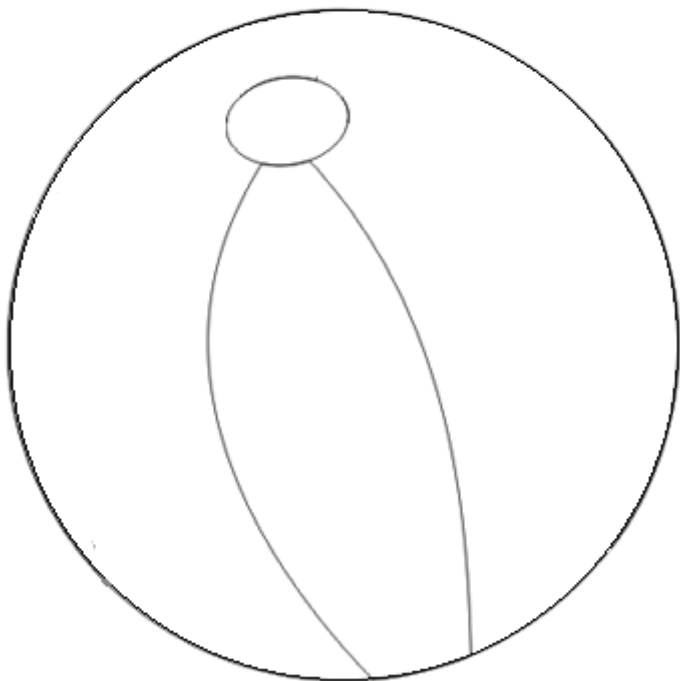


Для удобства можно использовать **средства масштабирования (Колесо вверх или вниз)**

Для того чтобы придать нормальную форму, сначала нажимаем правой кнопкой мыши на какую-нибудь сторону нашей фигуры и выбираем **Преобразовать в кривую**. Теперь с помощью стрелок изменять форму линии.

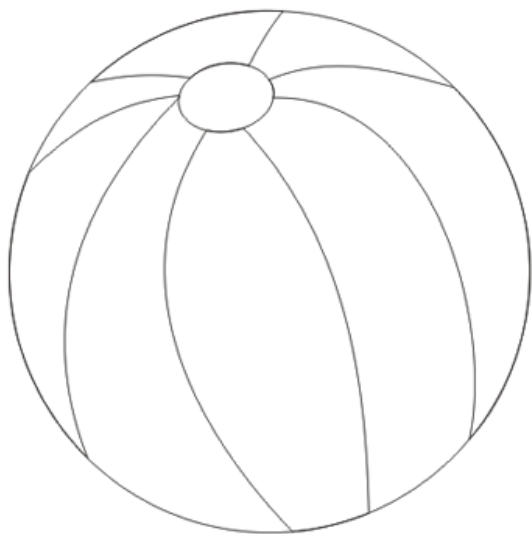


Так надо сделать с каждой из четырёх сторон, чтобы верхняя и нижняя сторона прилегала к окружностям, а боковые стороны были слегка изогнуты как на рисунке ниже.



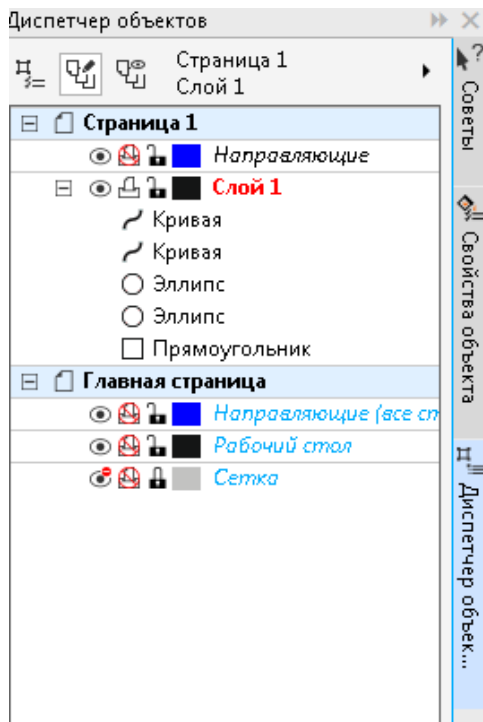
4) И так с небольшим удалением друг от друга создаём несколько полосок (**с помощью прямоугольника и его преобразования в кривую**)

В промежуточном итоге необходимо получить такой шар:

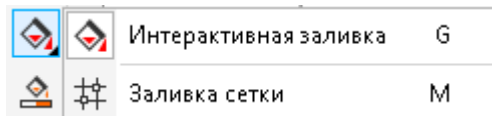


Все они должны находиться выше основной окружности. Это можно проверить на панели **Диспетчер объектов (с правой стороны окна)**.

На рисунке: *Страница 1* – текущая страница, *Прямоугольник* – фон страницы, два *эллипса* для мяча и две *кривых* (два измененных прямоугольника). *У вас кривых должно быть по итогу 8 шт.*



5) Теперь самое интересное: раскрашиваем шар с помощью **Интерактивной заливки (G)**.

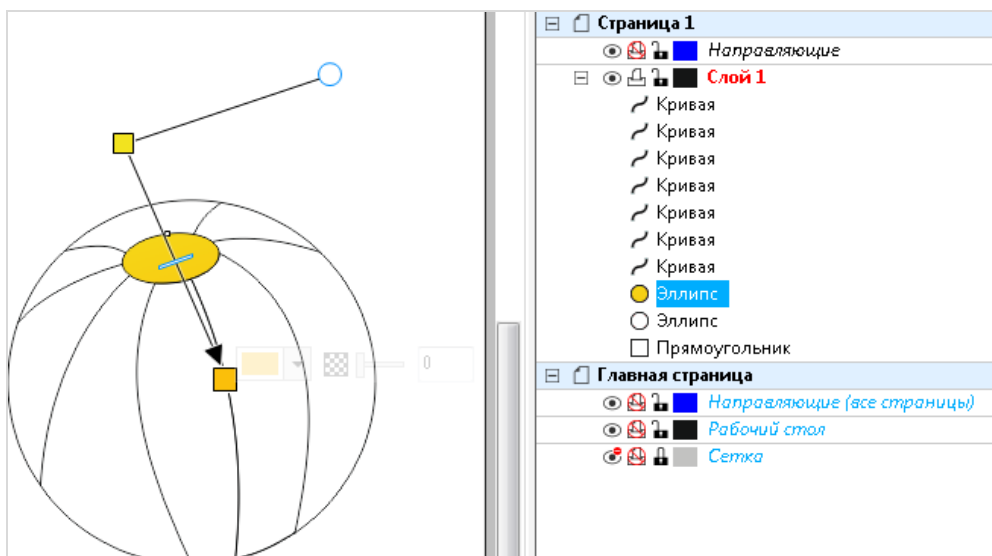


В верхней части меню выберите **Фонтанную заливку** (градиентная):



В диспетчере объектов выберем **маленький эллипс** (3 снизу).

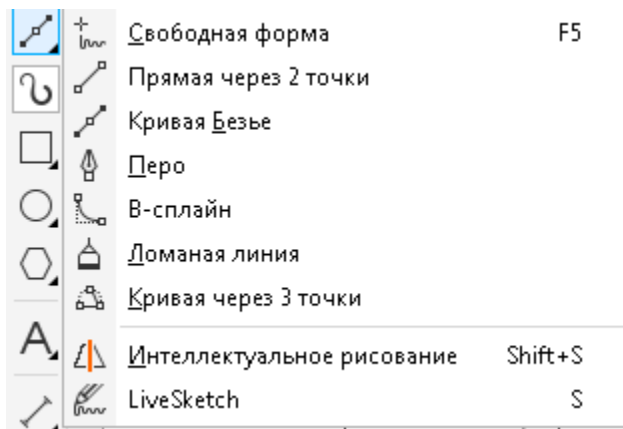
Затем попробуем протянуть инструмент заливки сверху вниз примерно, как на рисунке ниже (сверху цвет чуть светлее, чем снизу):



Покрасив все слои шара, у нас должен получился вполне симпатичный шар:

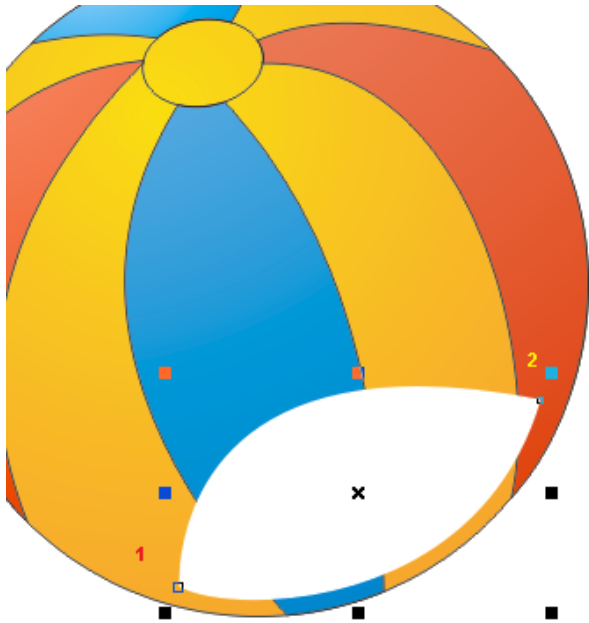


б) Но у него есть несколько недостатков: например, нет тени и бликов. Попробуем создать сначала блик. Для этого с помощью **Кривой безье** рисуем форму нашего блика.

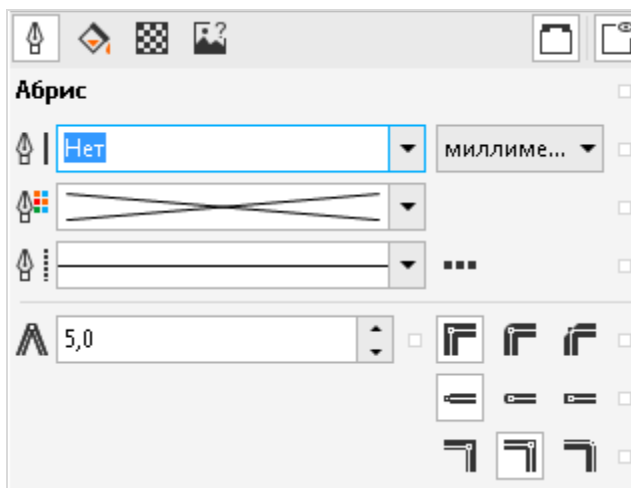


Устанавливаем начальную точку слева (**в позицию 1**), просто кликнув ЛКМ. Затем устанавливаем вторую точку (**в позицию 2**), но не отпускаем ЛКМ, а тянем так, чтобы получилась верхняя полуокружность. Затем снова ставим точку (**в позицию 1**), и не отпускаю, создаем нижнюю полуокружность.

**Правая сторона получится скругленной, не пугайтесь.** Чтобы это исправить выберите инструмент **Форма (F10)**, нажмите правой кнопкой мыши по узлу в точке 2 и выберите пункт **Перегиб**. Затем используя стрелочки у правого узла приведите кривые к виду как на рисунке снизу.



Также отрегулируйте общий вид фигуры как на рисунке, используя узлы и стрелки.

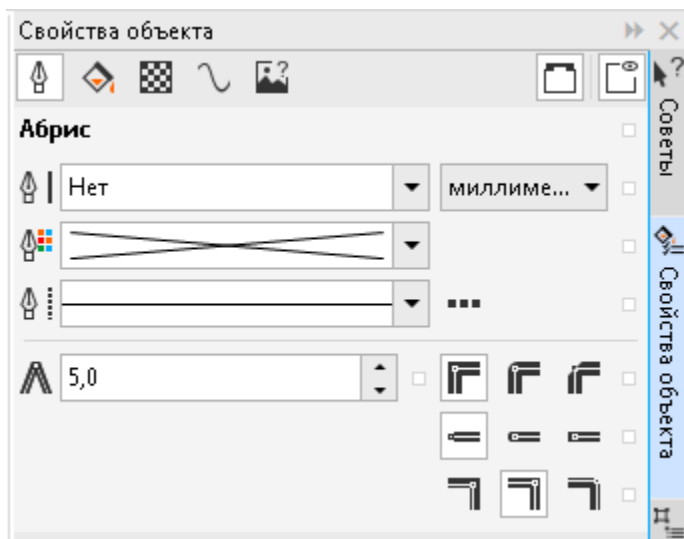


Выделите в Диспетчере объектов слой с этой кривой (должна быть самая первая) и залейте ее белым цветом с помощью однородной заливки (устанавливается в меню сверху слева)

Далее просто применяем к нашей белой форме инструмент **Прозрачность**.



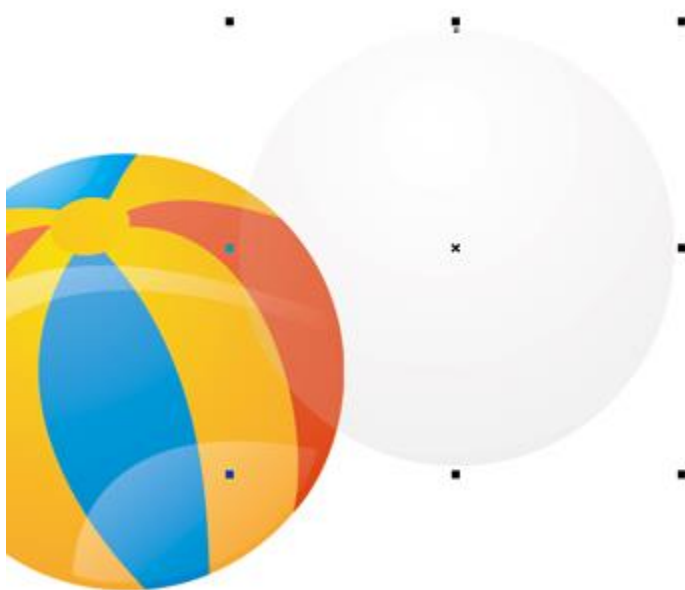
7) В Диспетчере объектов выделите все слои с Эллипсами и Кривыми через **Ctrl**. Перейдите на вкладку **Свойства объекта** и уберите у этих объектов контур.



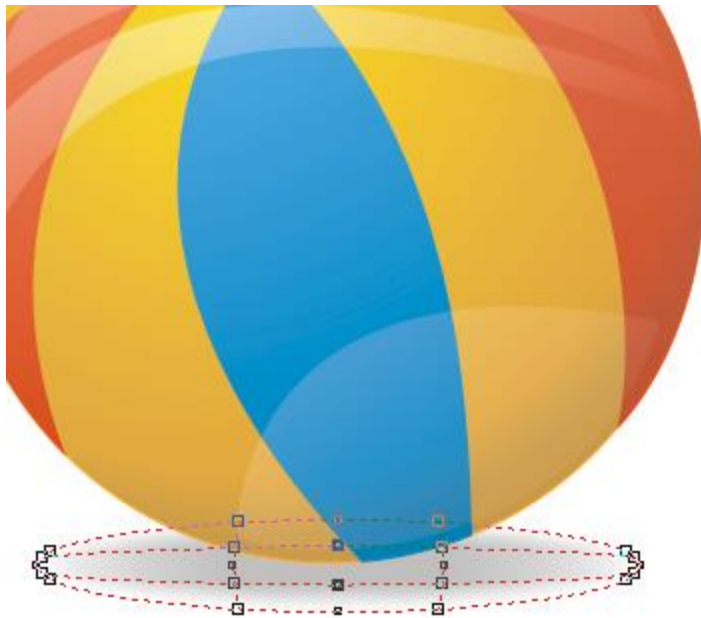
Затем перейдите в **Диспетчер объектов** и снимите выделение со всех объектов.

8) Аналогично создайте тень в виде полоски на шарике.

9) Для создания тени на самом шарике копируем основную окружность в **Диспетчере объектов** поверх остальных, затем заливаем его **серым** и применяем **Фонтанную заливку** инструмента **Прозрачность**:



10) Эту тень накладываем поверх всего шара, но перед бликами. Теперь осталось сделать только тень от самого шара. Для этого создаём эллипс **залить** его **серым цветом**. А затем использовать инструмент **Прозрачность**



11) В итоге у нас получился шар в векторе, который очень легко можно масштабировать для создания иконки:





## Практическая работа №38 (ПР38)

Тема: Векторный графический редактор CorelDraw

Цель: Рисуем гроздь винограда в CorelDraw



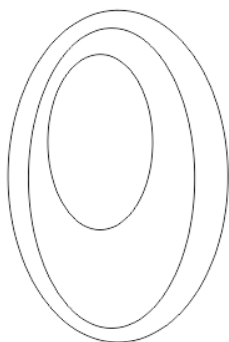
Вариант 1



Вариант 2

### 1) Рисуем виноградинки.

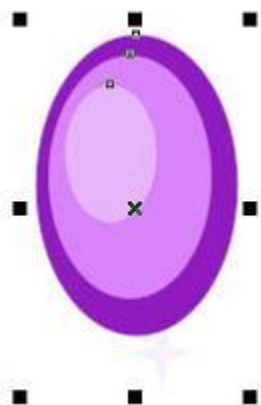
Рисуем **Эллипс (F7)**, копируем 2 раза. Располагаем как на рисунке ниже. Для удобного расположения **Эллипсов** используем **Инструмент выбора** и **Диспетчер объектов** справа окна.



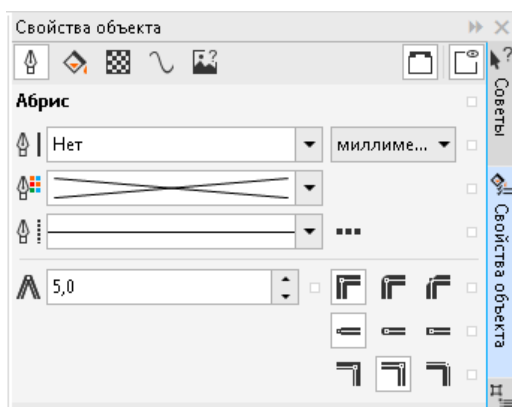
2) Используя **Диспетчер объектов** выберете основной слой виноградинки (**Эллипс снизу**). И с помощью инструмента **Интерактивная заливка (G)** залейте его **однородной заливкой** темно-фиолетового цвета как на рисунке снизу:



Залейте остальные эллипсы как на рисунке ниже:



3) Выделите все три эллипса в Диспетчере объектов, а затем во вкладке Свойства объекта скройте контур.



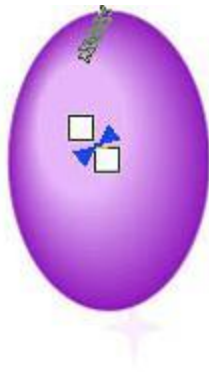
Затем перейдите в Диспетчер объектов и снимите выделение со всех объектов.

4) Применяем эффект перетекания со значением по желанию для этого:

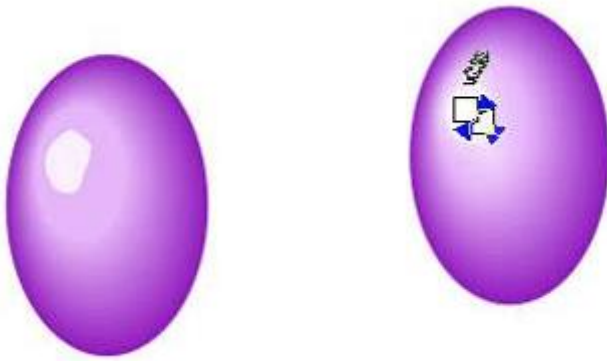
- в Диспетчере объектов выделите два нижних Эллипса
- отобразите вкладку Перетекание (Эффекты > Перетекание)
- во вкладке Перетекание установите параметры как на рисунке ниже и нажмите **Применить:**



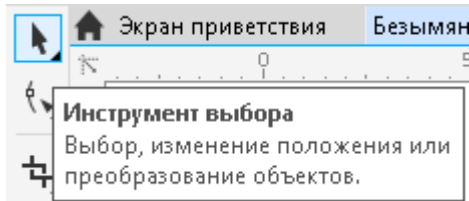
- аналогично сделайте перетекание между первыми двумя Эллипсами
- отрегулируйте Перетекание как на рисунке ниже:



5) Рисуем блик белого цвета и применяем к нему тот же эффект перетекания. В качестве блика можете использовать **Эллипс** или **Многоугольник**



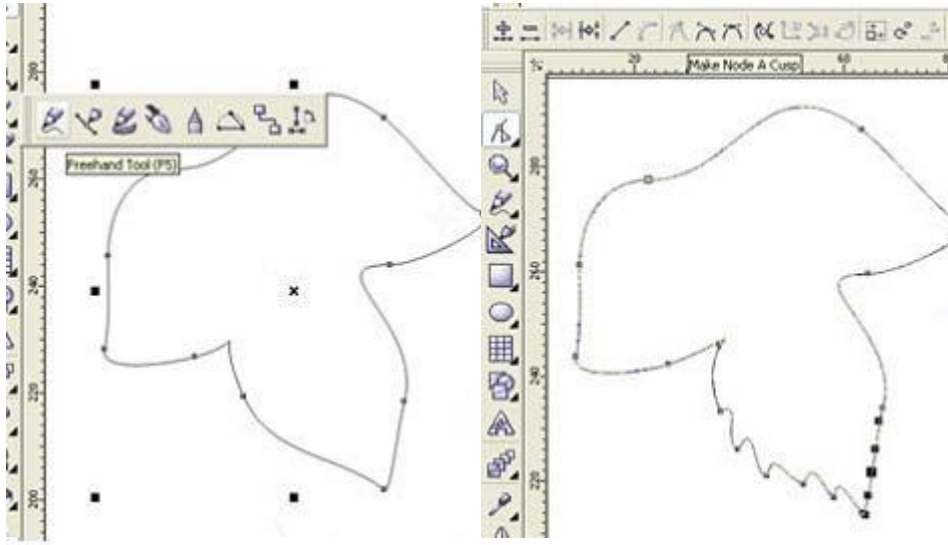
6) Многократно копируем нашу ягоду, меняем размеры, разворачиваем с помощью **Инструмента выбора**.



По желанию рисуем блики внизу. Под большим увеличением можно менять окраску для нижнего эллипса, создавая разные оттенки для виноградинок в грозди.



7) Рисуем листик с помощью инструмента **Свободная форма (F5)** или другого. Моделируем зазубринки.



Два раза дублируем, уменьшаем копии, придаём оттенки. И применяем Перетекание.



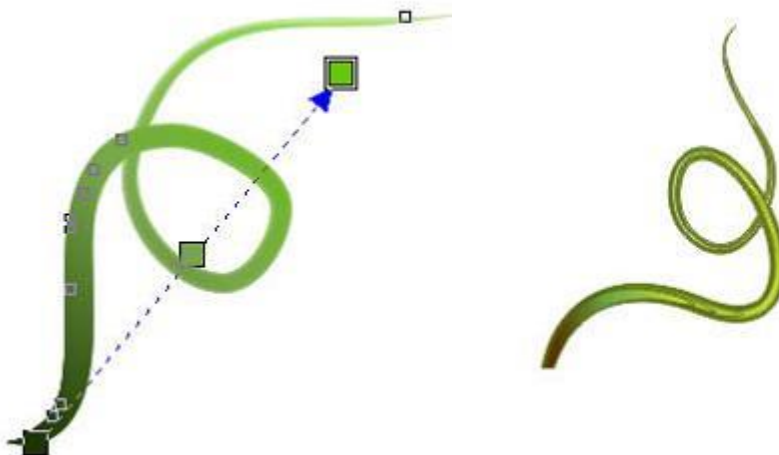
Поверх готового листика инструментом **Художественное оформление (I)** рисуем прожилочки и раскрашиваем их в светло-зелёный цвет.



8) Тем же инструментом создаём черенок и лозу.



9). Затем жмём Ctrl+K, удаляем стержень (черная полосочка внутри) и раскрашиваем наши закорючки градиентом. Ставим обводку более тёмного цвета.



10) Компонуем наши детали, дублируем листик и рисуем солнышко, но можно и без него.



## Практическая работа №39 (ПР39)

**Тема:** Изучение CSS

**Цель:** Освоить приемы создания web-страниц с использованием CSS (каскадных таблиц стилей)

Параметры внешнего вида документа можно задать с помощью каскадных таблиц стилей (Cascading Style Sheets, CSS). Таблица стилей, подобно шаблону форматирования текстов, может быть разработана отдельно от конкретного (X)HTML-документа, а затем применена к нему. Модификация содержимого таблицы стилей меняет внешний вид (X)HTML-документов, не затрагивая их структуры и информационного содержания. Одна и та же таблица стилей может применяться к нескольким документам, и, наоборот, к одному и тому же документу может быть применено несколько таблиц стилей. В последнем случае браузер учитывает приоритеты таблиц и по определенным правилам разрешает возникающие конфликты, в результате чего таблицы выстраиваются неким каскадом (отсюда и название-каскадные таблицы стилей).

С помощью CSS для любого элемента можно задать размеры и координаты расположения, а также другие параметры визуализации. Так что, применяя CSS, можно обойтись без тегов таблиц и фреймов, широко использующихся в качестве средства компоновки Web-страниц.

CSS содержит наборы стилевых параметров или правила форматирования.

### Задание 1.

Требуется определить для всех заголовков второго уровня шрифт "Comic Sans MS", 36 пунктов, пурпурного цвета. Соответствующее правило можно записать так:

## **h2 {font-family: Comic Sans MS; font-size: 36pt; color:Purple}.**

Код web-странички, содержащей афоризм, будет выглядеть таким образом:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset = "utf-8">
5 <title>Применение каскадной таблицы стилей</title>
6 <style type="text/css">
7     h2 {font-family: Comic Sans MS; font-size: 36pt; color:Purple}
8 </style>
9 </head>
10 <body>
11 <h2>Человек должен верить, что непонятное можно понять.</h2>
12 </body>
13 </html>
14
```

**Результат:**

Человек должен верить, что непонятное можно понять.

### **Задание 2.**

В фигурных скобках заключен список таких параметров, как начертание, цвет шрифта и размер шрифта.

Одно и то же правило можно применить к различным элементам, и, наоборот, для одного и того же элемента может быть задано несколько правил. Следующее правило определяет одинаковые стилевые параметры одновременно для заголовков второго и третьего уровней:

## **h2, h3 {font-family:Calibri; color:Teal;}.**

*Напишите страницу, которая будет выглядеть следующим образом:*

Слово "Завтра" придумано для людей нерешительных и для детей  
Иван Тургенев

Здесь шрифт Calibri и цвет Teal применяются как для тэгов h2, так и для тэгов h3.

### **Задание 3.**

Приведем пример каскадного задания стилевых параметров:

**h2, h3 {font-size: 36pt; color: Fuchsia}**

**h1 {font-family: Arial CYR}**

**h2 {font-family: Century Gothic}**

*Напишите страницу, которая будет выглядеть следующим образом:*



Уж сколько раз твердили миру

Об уважении к мундиру.

Он уважается! Пока...

В него не втиснут дурака.

Здесь сначала задаются размер и цвет шрифта для обоих заголовков, а затем для каждого из них определяется своя гарнитура шрифта (имя шрифта).

#### Задание 4.

Стилевые параметры, заключенные в фигурные скобки, можно назначить не только именам тегов, но и тегам по значениям их атрибута **id** (идентификатора элемента). Так можно изменить представление не всех, например, абзацев (тегов `<p>`), а только тех, которые имеют указанное значение атрибута **id**.

В предлагаемом ниже примере показаны стили для двух абзацев. Причем в одном случае стиль можно применять только к конкретному тэгу, имеющему **id="t1"**, во втором случае стиль можно применять для всех тэгов `<p>`:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset = "utf-8">
5 <title></title>
6 <style type="text/css">
7 #t1 {color:Purple; font-size:20pt; font-family:Comic Sans MS}
8 p{color:Blue;font-family:Calibri;font-size:16pt}
9 </style>
10 </head>
11 <body>
12 <p id="t1">От любой большой заботы Принцип лодыря спасет:</p>
13 <p align="center">Если хочется работать - ляг поспи и все пройдет.</p>
14 </body>
15 </html>
```

#### Результат:

От любой большой заботы Принцип лодыря спасет:

Если хочется работать - ляг поспи и все пройдет.

#### Задание 5.

Для одного и того же элемента можно назначить несколько разнообразных стилей, называемых классами. При этом селектор можно сформировать как имя тэга, за которым через разделительную точку следует имя класса, или же как просто имя класса, перед которым указывается точка. В первом случае класс применяется только с указанными в селекторах элементами. Во втором случае создается класс без предварительных указаний на то, к каким элементам он должен быть применен. Класс применяется к тем элементам, у которых значением атрибута `class` является имя этого класса.

В предлагаемом примере создаются два класса ( `red` и `blue` ) для заголовка второго уровня

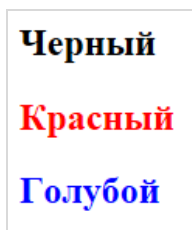


```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset = "utf-8">
5   <title></title>
6   <style type="text/css">
7     h2.red {color:Red}
8     h2.blue {color:Blue}
9   </style>
10 </head>
11 <body>
12 <h2>Черный</h2>
13 <h2 class="red">Красный</h2>
14 <h2 class="blue">Голубой</h2>
15 </body>
16 </html>

```

**Результат:**



### *Размещение каскадных таблиц стилей*

Каскадные таблицы стилей можно размещать или непосредственно в самом (X)HTML-документе, или во внешнем файле.

В первом случае код CSS заключается в контейнерный тэг `<style type="text/css">...</style>`. В наших предыдущих примерах мы использовали как раз такой подход.

Во втором случае CSS располагается во внешнем файле, а в вызывающем файле в разделе `<head>` размещают тэг `<link href="URL css-файла"/>` или в контейнер `<style>` - вставляют директиву `@import url ("url css-файла")`.

**Задание 6.** Создайте файл `mystyle.css`. Добавьте в него следующий код:

```

body {background-color: Aqua;}
h1{font-family:Verdana;color:Navy;font-size:36pt}

```

Затем создайте `html`-страничку со следующим содержимым:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset = "utf-8">
5   <title></title>
6   <style type="text/css">
7     @import url("mystyle.css");
8   </style>
9 </head>
10 <body>
11 <h1>Достойный человек не может не обладать широтой познаний и твердостью духа.
12   Его ноша тяжела, а путь его долог.</h1>
13 </body>
14 </html>

```

**Результат:**

# Достойный человек не может не обладать широтой познаний и твердостью духа. Его ноша тяжела, а путь его долг.

**Задание 7.** Связывание каскадной таблицы стилей с web-страницей с помощью ссылки  
Создайте новую таблицу стилей **StyleSheet2.css** со следующим содержимым:

```
body { background-color: Transparent}
h3 { font-family: Calibri; font-size:16pt; color: Navy}
h4{font-family: Courier New CYR; font-size:14pt; color: Orange}
```

Далее, создайте новый HTML-документ и введите в него следующий код:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset = "utf-8">
5   <title>Связывание каскадной таблицы стилей с web-страницей с помощью ссылки</title>
6   <link rel="stylesheet" type="text/css" href="StyleSheet2.css"/>
7 </head>
8 <body>
9   <h3>Поверь мне — счастье только там,</h3>
10  <h3>Где любят нас, где верят нам.</h3>
11  <br/>
12  <h4>/Михаил Юрьевич Лермонтов/</h4>
13 </body>
14 </html>
```

**Результат:**

```
Поверь мне — счастье только там,
Где любят нас, где верят нам.

/Михаил Юрьевич Лермонтов/
```

**Задание 8.** Встраивание стиля непосредственно в HTML-тэг

Создайте новый HTML-документ. В контейнер `<body>` вставьте следующий код

```
1 <body>
2   <h1 style="color:Fuchsia;font-family:Arial Cyr; font-size:18pt">
3     Учись и читайте.
4     <br/>
5     Читайте книги серьезные.
6   </h1>
7
8   <h2 style="color:Red; font-family: Calibri; font-size:14pt">Жизнь сделает остальное.</h2>
9   <h3>/Федор Михайлович Достоевский/</h3>
10 </body>
11
```

**Результат:**

```
Учись и читайте.
Читайте книги серьезные.

Жизнь сделает остальное.

/Федор Михайлович Достоевский/
```

## Задание 9. Применение правила форматирования по атрибуту id

Создайте новый HTML-документ и введите в него следующий код:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset = "utf-8">
5     <title>Применение правила форматирования по атрибуту id</title>
6     <style type="text/css">
7         #code_1 {color:Blue;font-family:Comic Sans MS}
8         h3{color:Red; font-family:Verdana}
9     </style>
10 </head>
11 <body>
12     <h3 id="code_1">Если тебе мешают люди, то тебе жить незначем.
13         Уходить от людей – это самоубийство.</h3>
14     <h3>Лев Николаевич Толстой</h3>
15 </body>
16 </html>
```

Результат:

Если тебе мешают люди, то тебе жить незначем. Уходить от людей – это самоубийство.

**Лев Николаевич Толстой**

## Задание 10. Использование классов в CSS

Создайте новый HTML-документ и введите в него следующий код:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset = "utf-8">
5     <title>Использование классов в CSS</title>
6     <style type="text/css">
7         p.a {font-family: Tahoma; text-align: center; font-size:12pt}
8         p.b {font-family:Comic Sans MS; text-align:justify; font-size:14pt}
9     </style>
10 </head>
11 <body>
12 <p class="a">Симонов Константин (Кирилл) Михайлович (1915 - 1979), поэт, прозаик, драматург.
13 Родился 15 ноября (28 н.с.) в Петрограде, воспитан был отчимом, преподавателем военного училища.
14 <p class="b">Первый роман "Товарищи по оружию" увидел свет в 1952,
15 затем большая книга - "Живые и мертвые" (1959). В 1961 Театр "Современник" поставил пьесу
16 Симонова "Четвертый". В 1963 - 64 пишет роман "Солдатами не рождаются".
17 (В 1970 - 71 будет написано продолжение - "Последнее лето".)</p>
18 </body>
19 </html>
20
21
```

Результат:

Симонов Константин (Кирилл) Михайлович (1915 - 1979), поэт, прозаик, драматург. Родился 15 ноября (28 н.с.) в Петрограде, воспитан был отчимом, преподавателем военного училища. Детские годы прошли в Рязани и Саратове.

Первый роман "Товарищи по оружию" увидел свет в 1952, затем большая книга - "Живые и мертвые" (1959). В 1961 Театр "Современник" поставил пьесу Симонова "Четвертый". В 1963 - 64 пишет роман "Солдатами не рождаются". (В 1970 - 71 будет написано продолжение - "Последнее лето".)

## Практическая работа №40 (ПР40)

**Тема:** Векторный графический редактор CorelDraw

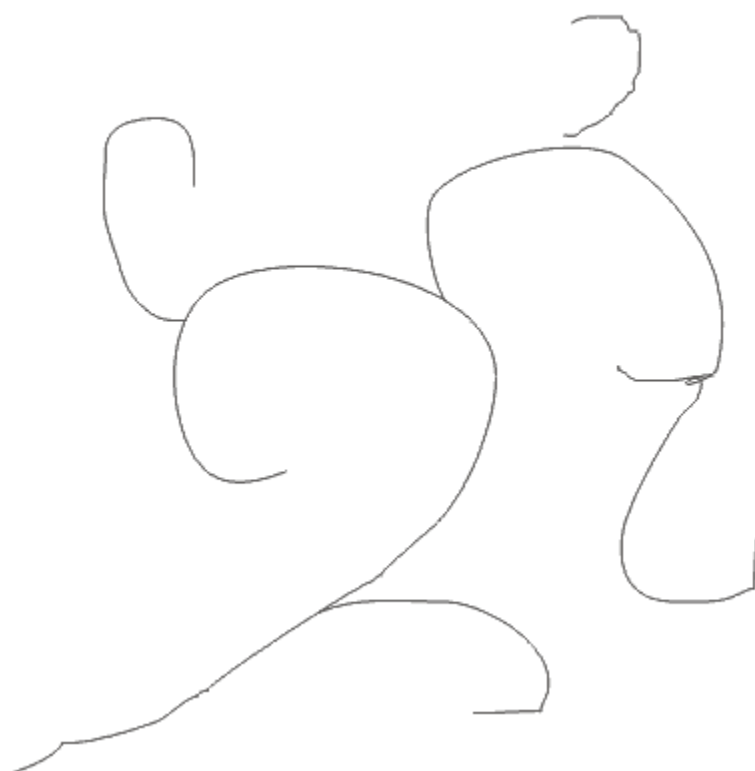
**Цель:** Рисуем модные растительные орнаменты.

---

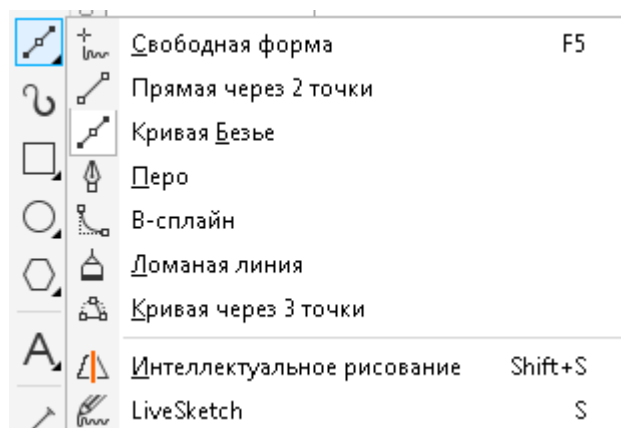
Урок по созданию векторной иллюстрации с растительными орнаментами в графическом редакторе CorelDRAW, от наброска до завершённой композиции, с подробным пошаговым описанием приемов работы с необходимыми инструментами и эффектами.

### 1. Выбор и построение формы.

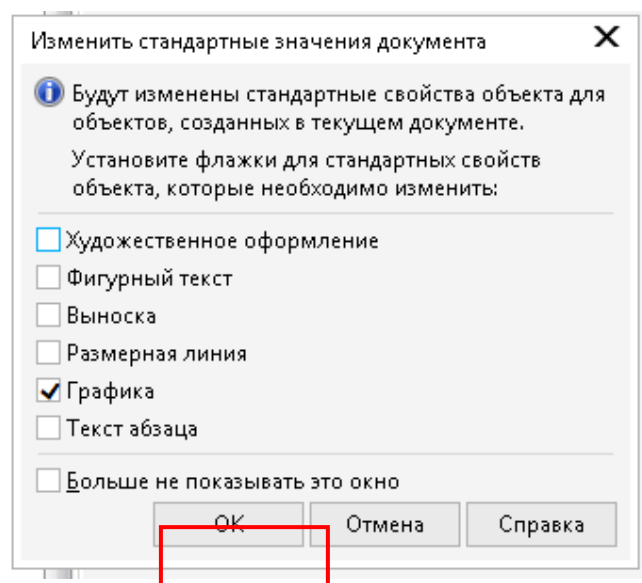
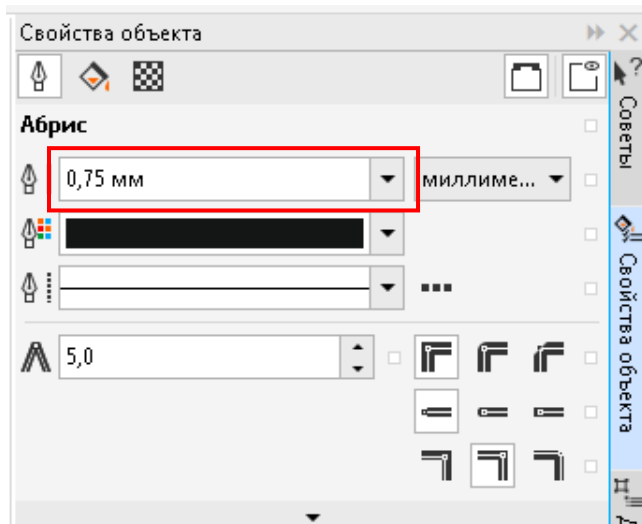
Определившись с будущей формой и композицией, сделаем набросок инструментом **Свободная форма (F5)**



Теперь поверх наброска аккуратно **отрисует линии** с помощью **Кривых Безье**.



Во вкладке **свойства объекта** установите **толщину кривой 0,75 мм**.



Чтобы **прервать** рисование линий нажмите **Пробел**

Линии должны быть именно **незамкнутыми линиями**



Орнамент представляет собой некий конструктор, к разным формам можно применять разные кисти и оформлять различными элементами, формы можно редактировать, модифицировать, собирать простые в сложные и т.д. тем самым создавая множество вариаций.

## 2. Детализация.

Приступим к самой интересной части.

Придадим форму основе орнамента, применив различные кисти к различным частям формы.

Для этого отобразите вкладку **Эффекты > Художественное оформление**. Выбрав в **Диспетчере объектов** одну из **Кривых** измените ее орнамент во вкладке **Художественное оформление**.

Не забудьте изменить **толщину линий** на примерно **1,5 -2мм**



**Аналогично проделайте с остальными кривыми.**

В диспетчере объектов удалите объекты путей, нарисованные **Свободным пером**.

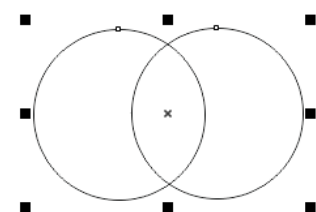
С помощью **Интерактивной заливки (G)** окрасьте кривую в любой понравившийся вам цвет

С помощью инструмента **Форма (F10)** исправьте неровности, должно получиться примерно так:

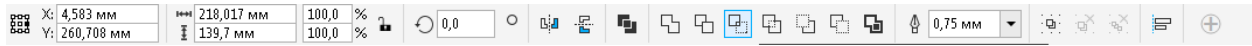


## 3. Создание листиков и цветочков с Искажением:

1) Создаем простую форму из пересечения двух эллипсов.

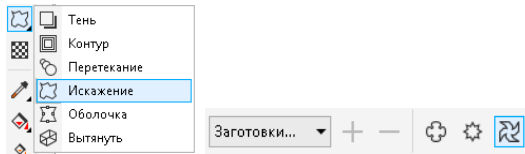


Выделяем их пересечение

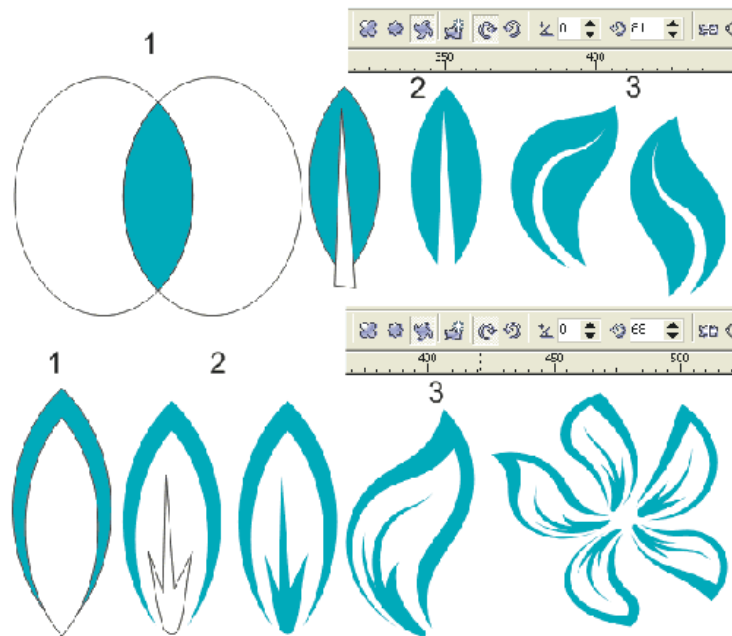


2) Делаем прожилку для листика обрезав простой кривой, и контур для лепестка обрезав ее уменьшенной копией, также рисуем простенькую серединку в лепестке.

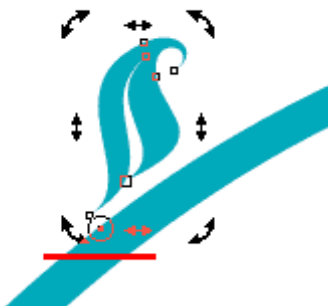
3) Копируем заготовку несколько раз и применяем эффект, скручивающий **Искажения** к каждой копии с разными настройками, тем самым создавая разнообразные элементы оформления. Из нескольких лепестков собираем цветок.

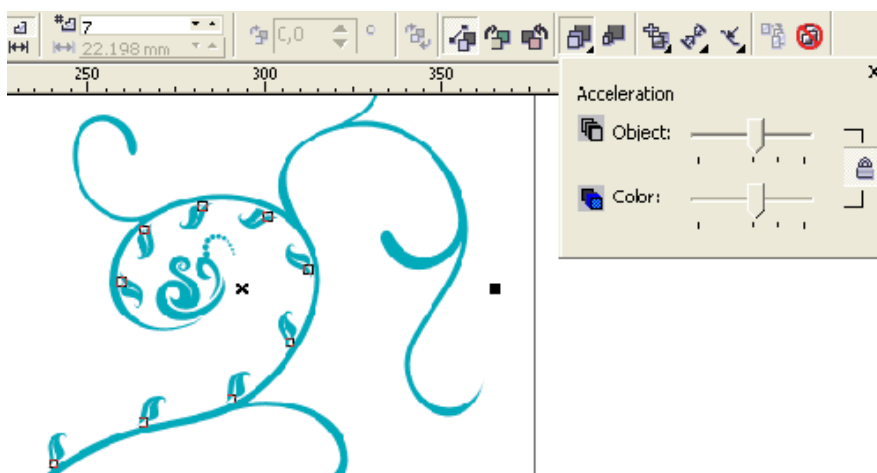
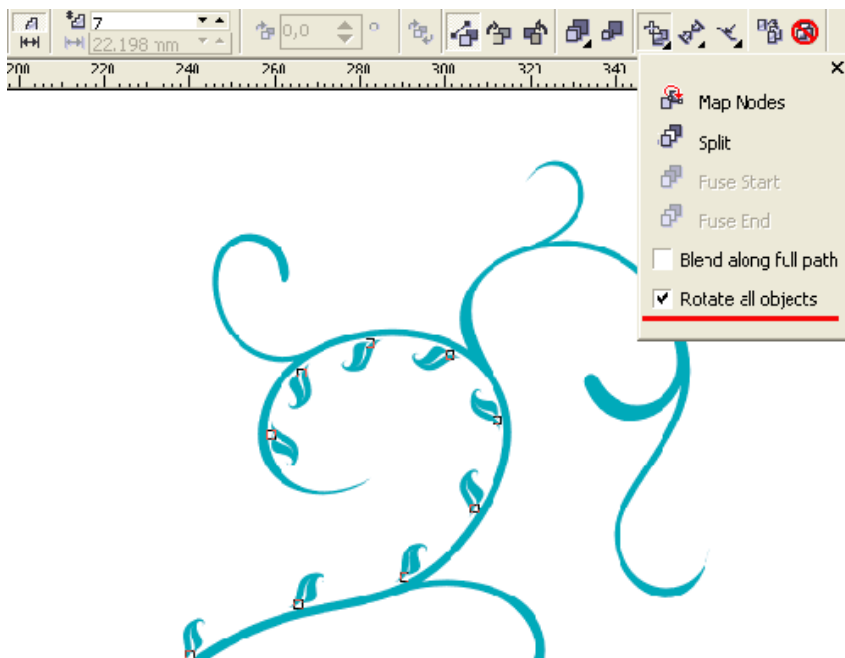


**Сделайте листок и цветочек как на рисунке ниже**



4) Теперь расставим вдоль основных линий орнамента листики. Регулируйте листики с помощью инструментов **Выбора** и **Формы (F7)**





Расставляем цветы и более крупные элементы, а чтобы небыло ощущения пустоты, добавим тонких завитушек - просто набросаем линии и переведем в объекты.





Все почти готово, осталось раскрасить, на фон можно кинуть градиент или продублированный зеркальный орнамент, покрашенный в оттенок фона или и то и другое, опять же это дело вашего вкуса и фантазии.



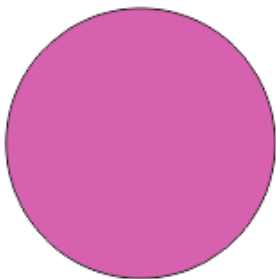
## Практическая работа №41 (ПР41)


**Тема:** Векторный графический редактор CorelDraw

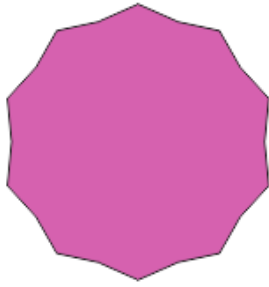
**Цель:** Рисуем различные элементы для букетов цветов

### Задание 1.

1. Нарисуйте правильную **окружность** диаметром около **80 мм (смотрите в нижней части рабочего окна)** с помощью **Эллипса (F7)** (удерживая нажатым **Ctrl**).
2. Окружность залейте фиолетовым цветом.



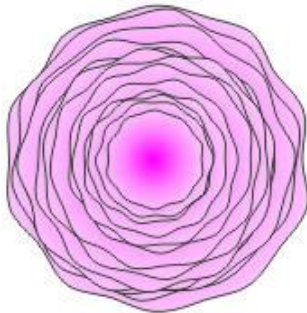
3. Выделяем фигуру и выбираем инструмент **Искажение** , используем верхнюю панель настроек, чтобы выбрать тип искажения (**Искажение в виде застёжки-молнии**). Выставляем следующие параметры: амплитуда (Amplitude) - 17, частота (Frequency) - 4, выбираем Сглаженное искажение



4. Для создания лепестков необходимо создать несколько концентрических копий (13 шт примерно). Для создания копий можно воспользоваться окном Преобразование (Alt+F8). Используйте этот инструмент делая копию на переднем плане наиболее маленькой, остальные копии (на заднем плане) чуть больше.

Поверните каждую из копий объекта, чтобы они сместились относительно друг друга (как показано на рисунке ниже). Чтобы быстро повернуть объект с помощью **Инструмент выбора**, щелкните на любом выбранном объекте **по-центру**, чтобы включить режим поворота/сдвига, и за любой угол поверните объект.

5. Повторяйте пока не получите цветок как на рисунке ниже.

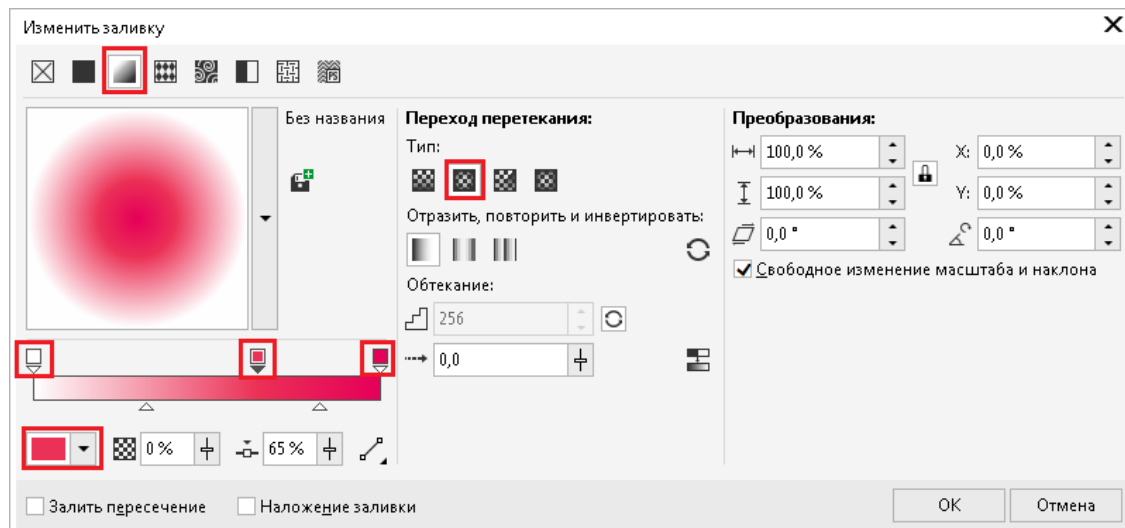


## Задание 2.

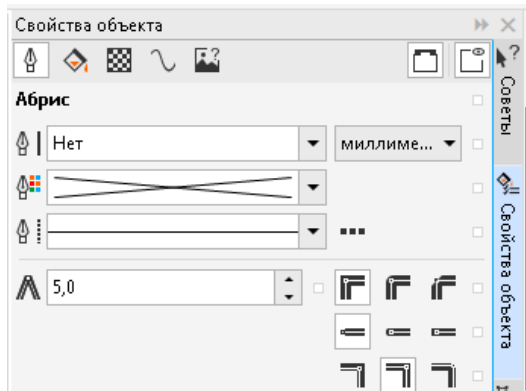
1. Нарисуйте **окружность** диаметром около **20 мм (смотрите в нижней части рабочего окна)** с помощью **Эллипса (F7)** (удерживая нажатым **Ctrl**).

2. Назначьте радиальную заливку (**Эллиптическая фонтанная заливка**) первой окружности. Можете использовать инструмент **Интерактивной заливки (G)**, но для быстроты и точности лучше нажать **F11** на выделенной окружности, чтобы вызвать диалоговое окно заливки. Выбираем тип заливки: **Эллиптическая фонтанная заливка**

Устанавливаем маркер цвета с белой заливкой на позицию "0", а маркер с пурпурным цветом на позицию "100". Добавляем новый маркер на позицию "65", назначаем ему заливку пурпурным цветом.



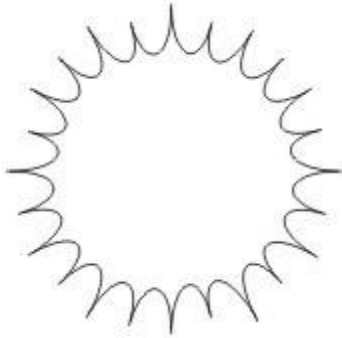
Убираем контур у окружности в **Свойствах объектов**



2. Нарисуйте еще одну **окружность** диаметром около **80 мм** (**смотрите в нижней части рабочего окна**) с помощью **Эллипса (F7)** (удерживая нажатым **Ctrl**).

3. Не заливайте ее.Используем верхнюю панель настроек, чтобы выбрать тип искажения (**Искажение в виде застёжки-молнии**). Выставляем следующие параметры: амплитуда (Amplitude) - 30, частота (Frequency) - 5, нажмите Enter.

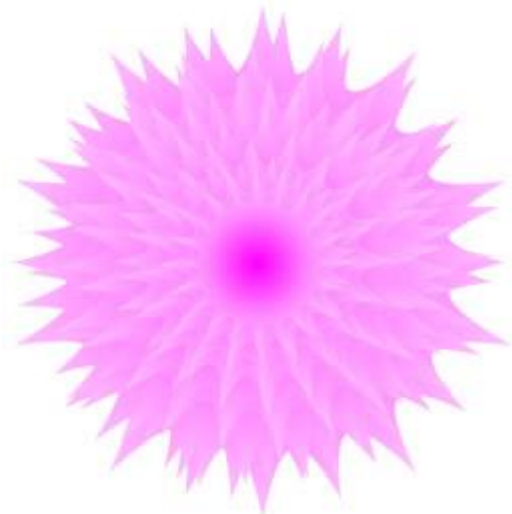
Далее выберите тип искажения (**Искажение при сжатии и растяжении**), выставьте Amplitude - 20, нажмите Enter.



4. Окрасьте данную фигуру в фиолетовый. Скопируйте ее некоторое кол-во раз используя окно **Преобразование**. По центру расположите ранее созданную фигуру с **Эллиптической фонтанной заливкой**.

Для этого можете использовать **Диспетчер объектов** и **Инструмент выбора**

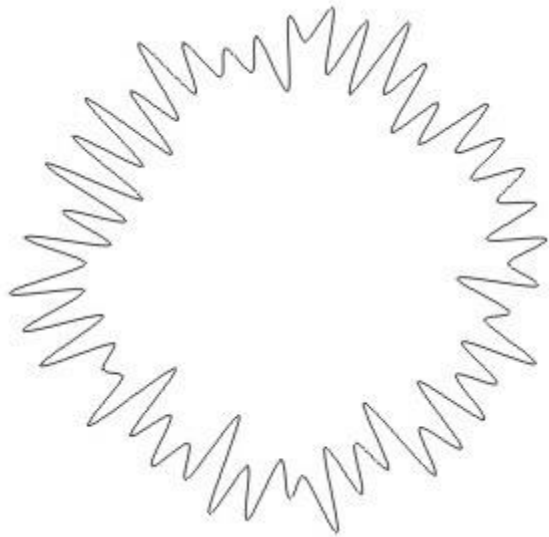
5. Повторяйте пока не получите цветок как на рисунке ниже.



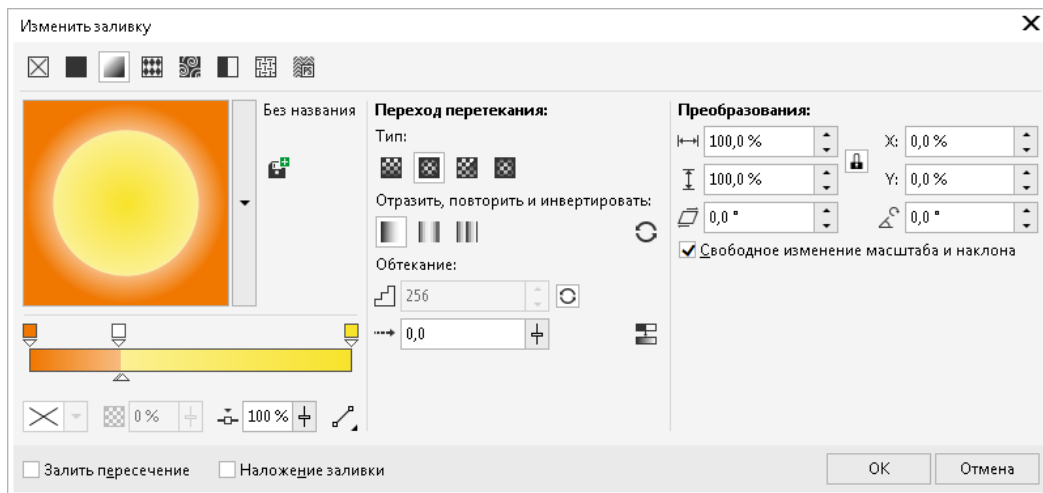
6. Повторите **Искажение** с другими настройками для другой **Окружности** (не забывайте преобразовывать окружность в кривую Ctrl+Q):

**При сжатии и растяжении** -> Амплитуда 5 -> Enter.

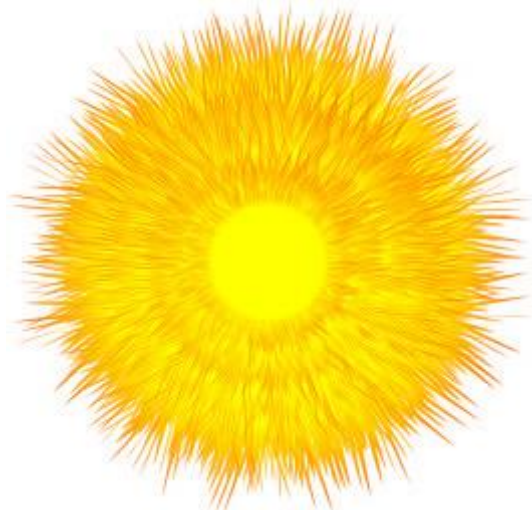
**В виде застёжки-молнии** -> Случайное искажение -> Амплитуда 100, Частота 20 -> Enter.



Не снимая выделения с цветка, меняем заливку:



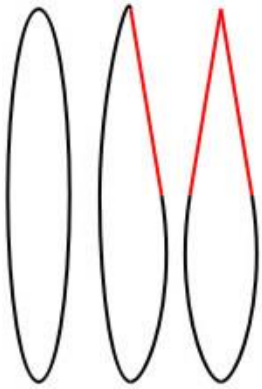
Копируя цветок и наложивая друг на друга используя окно **Преобразование** (не более 8-9 раз) и поворачивая объекты попробуйте получить подобный результат:



### Задание 3. Рисуем листья

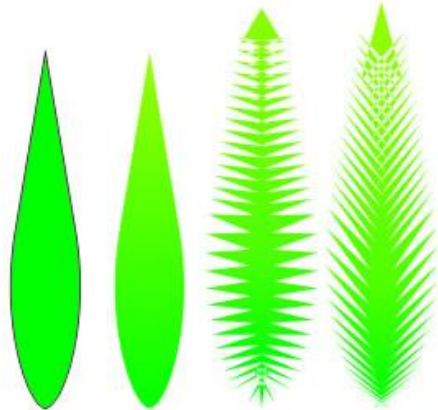
1. С помощью **Эллипса (F7)** нарисуйте вытянутый эллипс размером примерно 25x90 мм и преобразуйте его в кривую (**Ctrl+Q**).

Используя инструмент **Форма (F10)** сделайте две верхних ветки кривых прямыми линиями. Для этого выделите верхнюю точку эллипса, нажмите ПКМ > Перегиб. Используя стрелочку приведите фигуры к итоговому виду внизу.



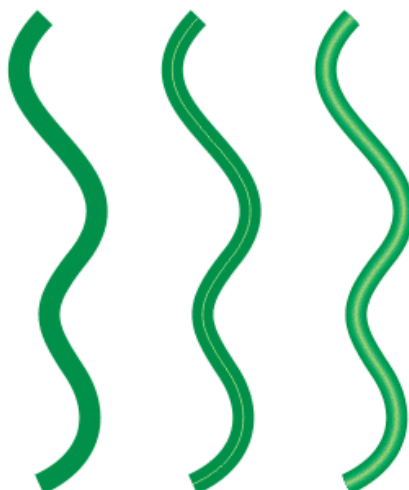
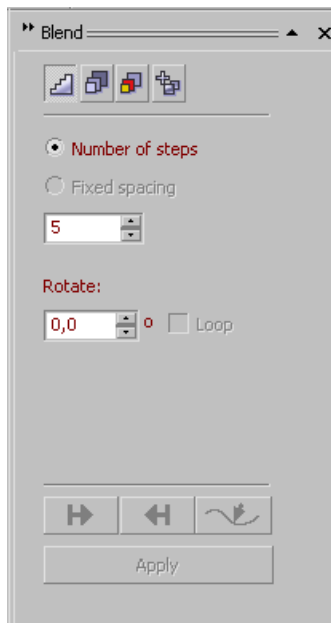
В итоге получается каплеобразная кривая.

2. Назначьте объекту градиентную заливку от темно-зеленого к светло-зеленому, скройте контур.
9. Выберите **Искажение**, тип искажения **В виде застёжки-молнии**. Выставьте настройки для начального искажения: Амплитуда - 50, Частота - 30. Чтобы откорректировать направление зубчатых краев листа перетащите ромбообразный маркет в верхнюю часть объекта. Теперь зубчики смотрят вверх.



#### Задание 4. Рисуем стебли

1. Создайте кривую с помощью **Кривой Безье**. Чтобы остановить рисование нажмите **пробел**. Выделив нарисованный объект в верхней части окна назначьте толщину около 8 мм. В Свойствах объекта залейте фигуру темно-зеленым цветом.
2. Скопируйте объект. Назначьте копии кривой толщину 1 мм и светло-зеленую заливку.
3. Поместите тонкую кривую на более толстую кривую
4. Выделите обе кривых в **Диспетчере объектов**, откройте окно свойств перетекания (Эффекты > Перетекание), выставьте число шагов 5.



Примените эффект.

**Задание 5.**

Используя полученный опыт и готовые фигуры сформируйте подобные букетики цветов. Можете нарисовать букет на свое усмотрение.





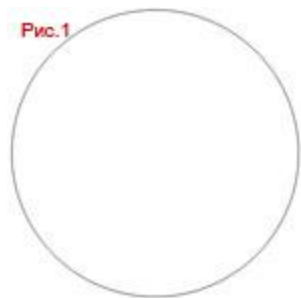
## Практическая работа №42 (ПР42)

**Тема:** Векторный графический редактор CorelDraw

**Цель:** Рисуем простые часы с циферблатом

1. Инструментом **Эллипс (F7)** рисуем круг для того чтобы он действительно был круглым удерживаем клавишу **Ctrl**.

Нажимаем **Ctrl+Q**, так мы получим кривую. Рис.1

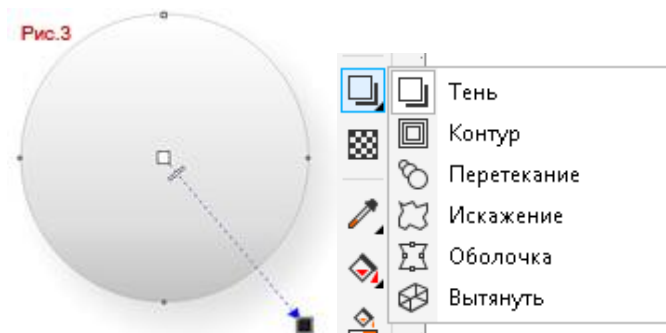


2. Закрашиваем обводку в окне Свойства объекта цветом **C:0 M:0 Y:0 K:30**

Изливаем **Линейной фонтанной заливкой** с помощью инструмента **Интерактивная заливка (G)**



3. Добавим **Тень** Рис.3



4. Копируем эту окружность **Ctrl+C** и вставляем поверх исходной **Ctrl+V** (правда удалите с нее тень)

5. Меняем у внутреннего слоя **Линейной фонтанной заливку**, чтобы казалось что окружность впадает в другую. Убираем у нее контур



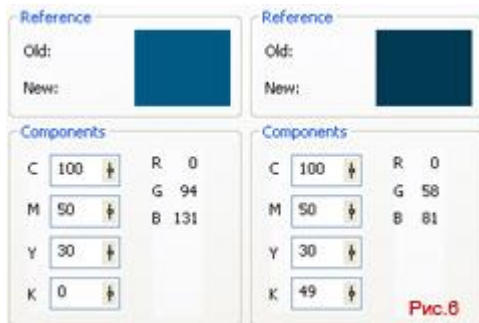
5.Теперь Копируем эту окружность **Ctrl+C** и вставляем поверх исходной (перевернутой) **Ctrl+V**  
 Контур красим в черный цвет, саму фигуру пока что красим в **белый**



6.Если все получилось тогда пора перекрасить белый цвет, и залить кривую двухцветной  
**Эллиптической фонтанной заливкой**, более светлый синий должен быть в центре, вот цвета Рис.6



Результат должен быть подобным Рис.6А



7.Нарисуем еще **два круга** и расставим их, **первый** тот что побольше поставим в **центр**, **второй** **повыше**. Мы собираемся сделать циферблат.

Выделяем оба объекта через **Shift** и группируем (**Ctrl+G**)



Рис.7



8. После выделяем нашу группу и смещаем точку центра вращения в низ как на Рис.8



9. Копируем группу и вставляем ее про верх исходной.

Удерживая клавишу **Ctrl** поворачиваем группу на 90 градусов в любом направлении Рис.9А и так еще 2 раза как на Рис.9В.

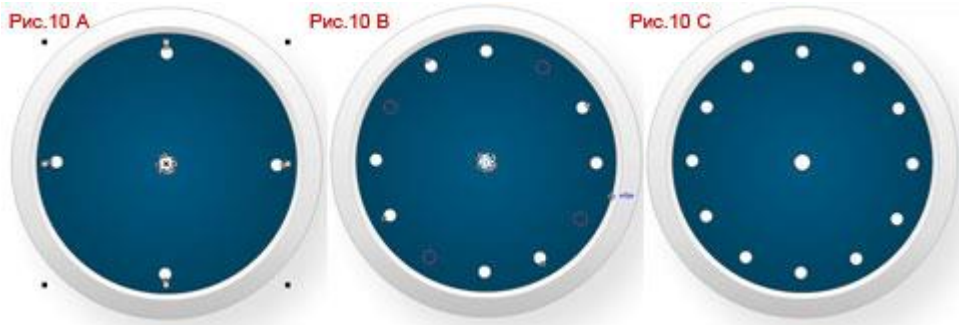
Рис.9 А



Рис.9 В

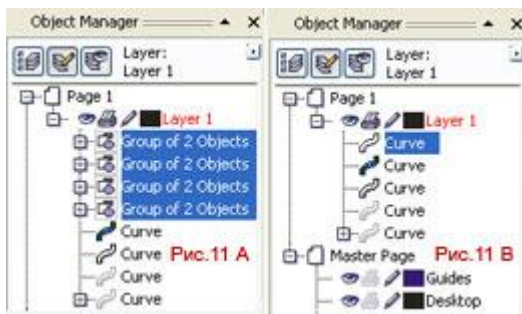


10. Снова выделяем группы циферблата через **Shift** и объединяем их **Ctrl+G**. после создаем еще одну копию и поворачиваем группу но на этот раз градусов где-то на 22 Рис.10В и еще раз



11. Теперь осталось немного упростить все это

Берем **инструмент выбора** и выделяем все группы Рис.11А и выбираем **Объединить** и у нас должно получиться как на Рис.11В



12. Центральной точке можно придать немного объёма при помощи градиентной заливки

После начинаем создавать стрелочки (Для этого можете воспользоваться любым удобным вам инструментом, например **Прямые через 2 точки**)

Конечно можно было бы ещё украсить циферблат цифрами но это по желанию.



13. Теперь еще бы стеклянного блеску и можно считать работу готовой

Рисуем новую полу-окружность (С помощью Кривой Безье) для блеска Рис.13А

Красим ее в **белый** и применяем **Прозрачность** Рис.13В

14. Тоже самое повторяем только внизу Рис.13В и Рис.13D



вот и результат

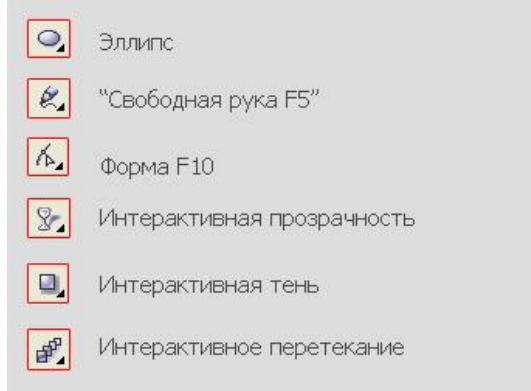


## Практическая работа №43 (ПР43)

**Тема:** Векторный графический редактор CorelDraw

**Цель:** Рисуем божью коровку в CorelDRAW.

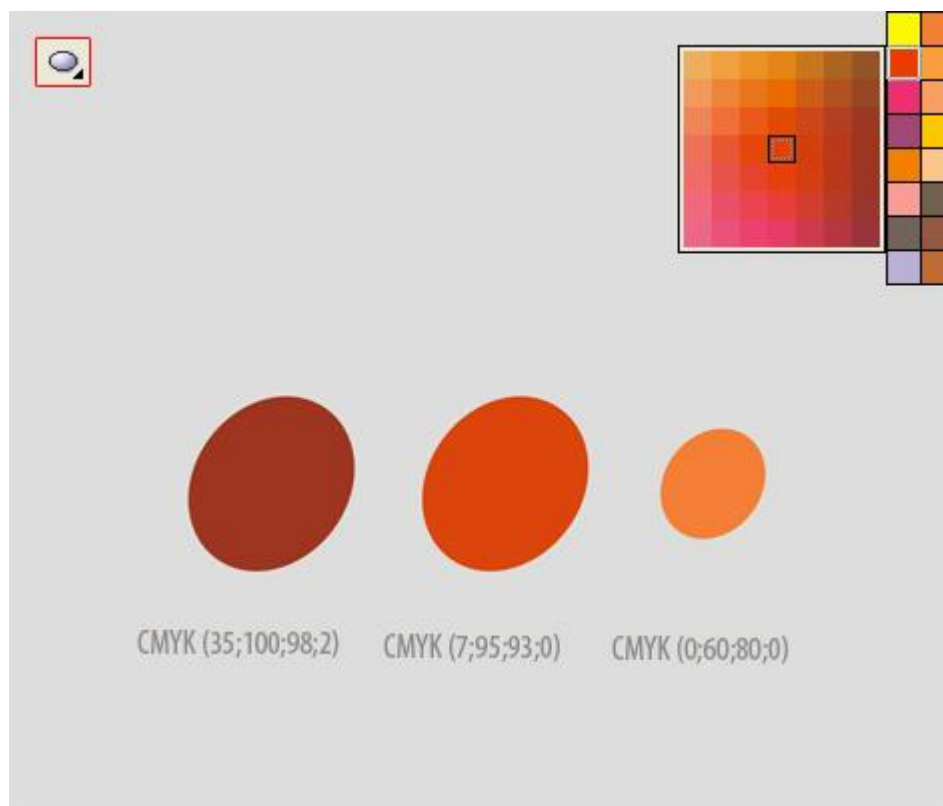
Для начала ознакомимся с инструментами, которые будем использовать при рисовании:



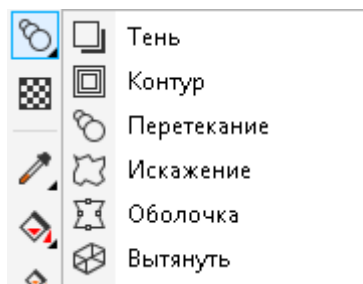
1) С помощью инструмента **Эллипс (F7)** рисуем три овала, каждый последующий меньше предыдущего. Заливаем их соответственно цветами палитры **СМУК (35;100;98;2), (7;95;93;0), (0;60;80;0)**.

**Не забудьте убрать контуры у овалов.**

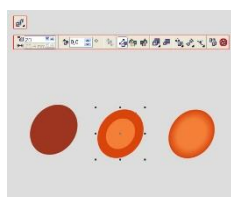
**Примечание!** Для открытия более детальной палитры, которая изображена на рисунке, необходимо удерживать левую клавишу мыши на выбранном цвете пару секунд для открытия палитры оттенков этого цвета.



2) Первый более темный **Эллипс** будет служить нам основанием, его мы не трогаем. Накладываем 2 последних эллипса друг на друга. С помощью клавиши **Ctrl** в **Диспетчере объектов** выделяем оба объекта и далее используем инструмент **Перетекание**. Для использования инструмента тянем от одной фигуры к другой



Число шагов между фигурами при перетекании оставляем по умолчанию – **20**. Это первый способ осуществить интерактивное перетекание, если у вас возникли проблемы, воспользуйтесь способом, указанным ниже.

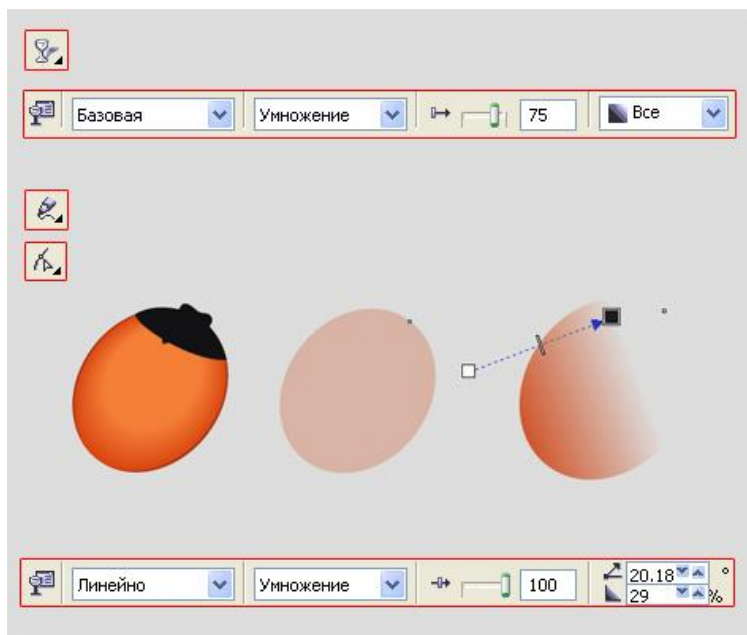


3) Теперь получившийся объект накладываем на третий овал, который больше предыдущих, его помещаем под полученные перетеканием формы, он будет служить некоей обводкой и подчеркивать форму брющка.

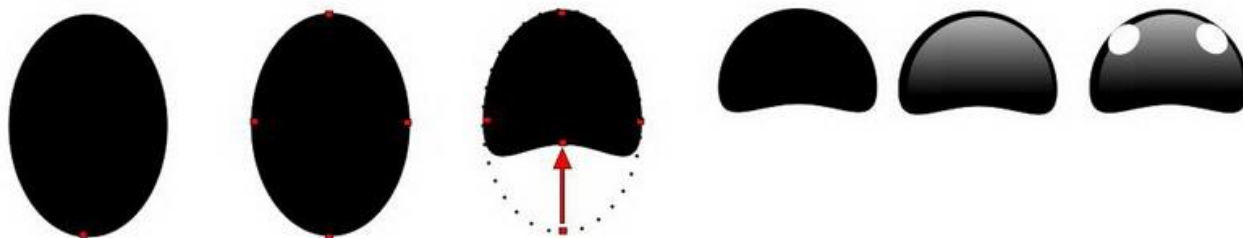
С помощью инструмента **Свободная фопма (F5)** или **Перо**, кому как удобно, рисуем верхнюю часть нашей букашки, подправляем полученный объект и сглаживаем в местах узлов с помощью инструмента **Форма (F10)**.

Снова рисуем 2 овала, с помощью **эллипса (F7)**. На оба применяем инструмент **Прозрачность**. В настройках данного инструмента для первого овала выбираем тип - **однородная прозрачность 75%**.

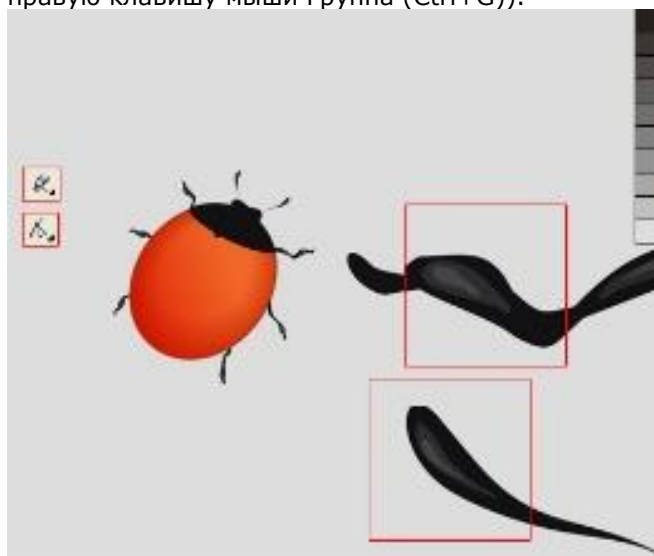
Для второго объекта используем тип - Фонтанную **линейную прозрачности с режимом умножение**. Накладываем получившиеся фигуры поверх заготовки для букашки выше всех слоев.



А это подсказка как нарисовать быстро мордочку не забудьте сначала перевести **эллипс** в кривые правой кнопкой мыши



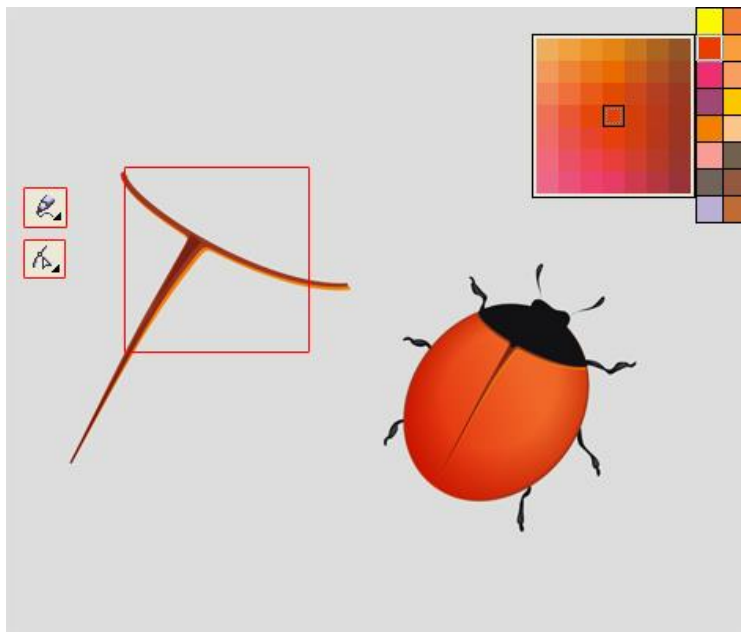
4) Далее рисуем ножки и усики нашей букашке. Заливаем объекты черным цветом, поверх них инструментом **Свободная форма** или **Перо** рисуем еще объекты, используем из палитры оттенки серого. Каждую лапку отдельно с находящимися на ней объектами группируем (выделяем и нажимаем правую клавишу мыши Группа (Ctrl+G)).



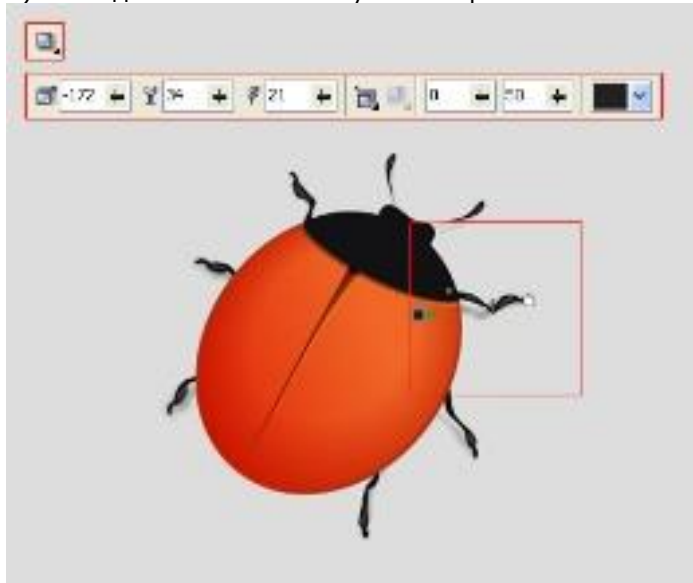
Крупный план ножек:



5) Рисуем букашке крылышки. Они состоят из объектов, различающихся только размером и цветом, при этом используем оттенки красного. Накладываем их один поверх другого.

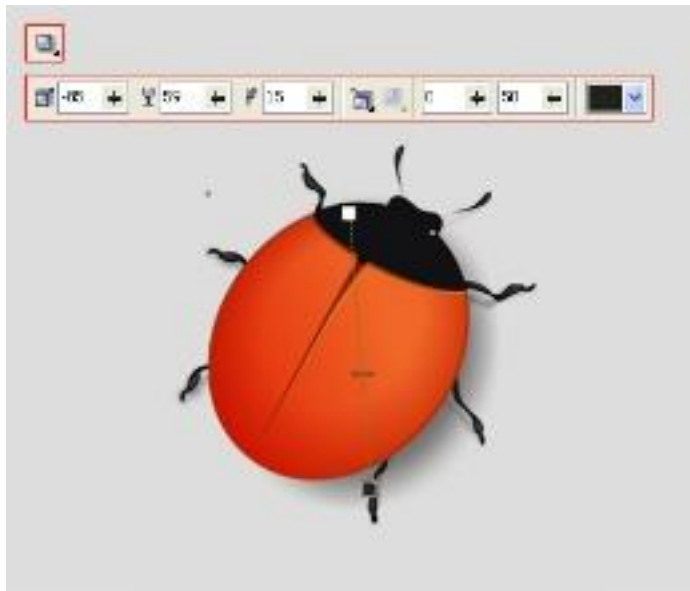


6) К каждой лапке нашей букашки применяем тень с настройками, как показано на рисунке.



7) Теперь выделяем спинку (основание) и также применяем **инструмент Тень** с настройками, как показано на рисунке.





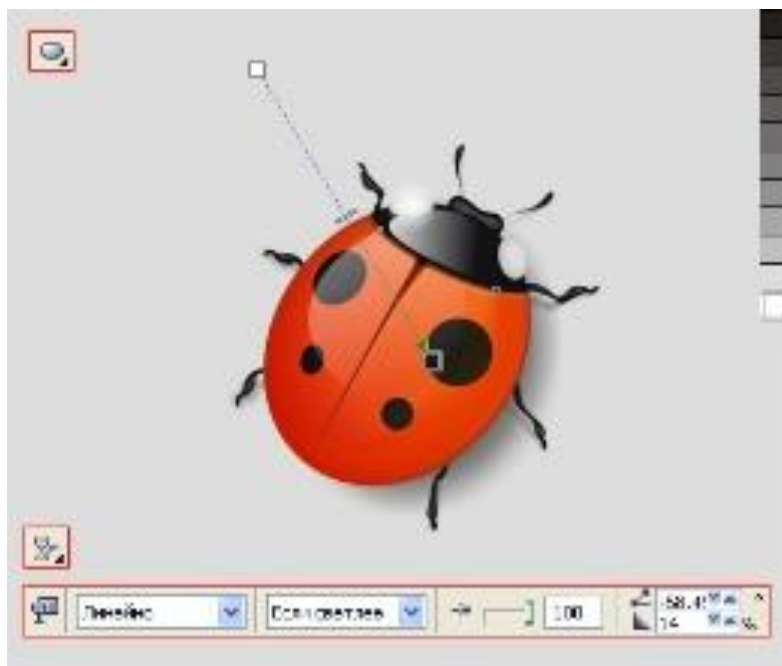
8) Дорисовываем объекты на самой мордочке букашки и рисуем на спинке пятна, эллипсы черного цвета, искажая их до определённой формы с помощью указателя. Накладываем на мордочку объекты белого цвета, один из объектов делаем меньше другого, придаем им стандартную базовую прозрачность.



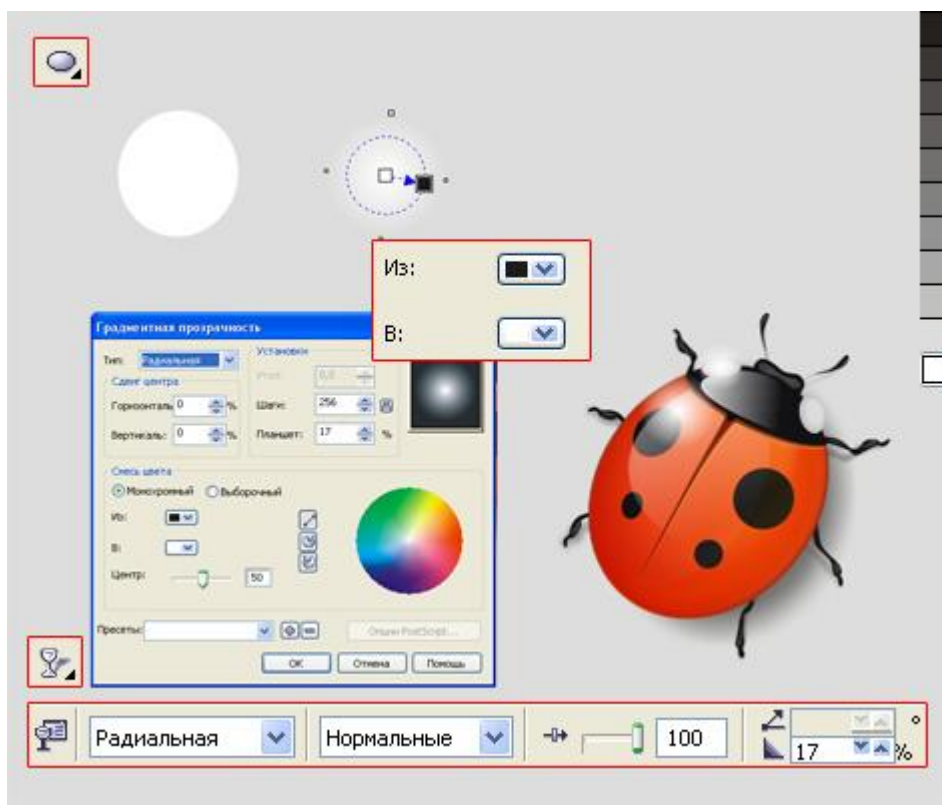
9) Нарисуем глазки. Инструментом Эллипс рисуем два объекта, используем **Искажение**, получаем овалы, заливаем определённым цветом, используем оттенки серого из палитры и применяем на выделенные объекты инструмент **Перетекание** с количеством в 20 шагов. Размещаем на букашке.



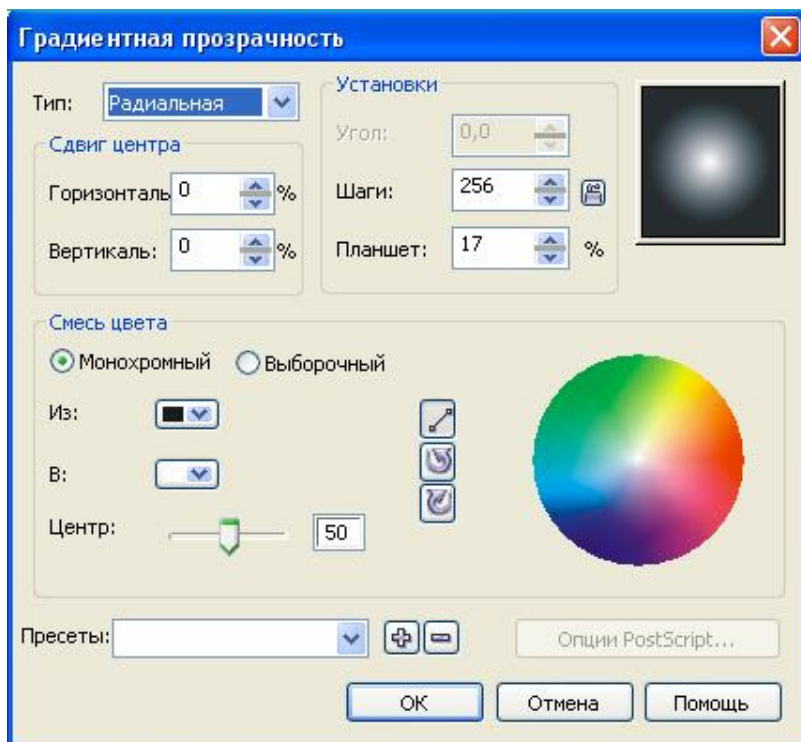
10) Создаем блики на спинке. Также рисуем овалы, заливаем белым цветом и применяем интерактивную **прозрачность линейного типа с режимом светлее**.



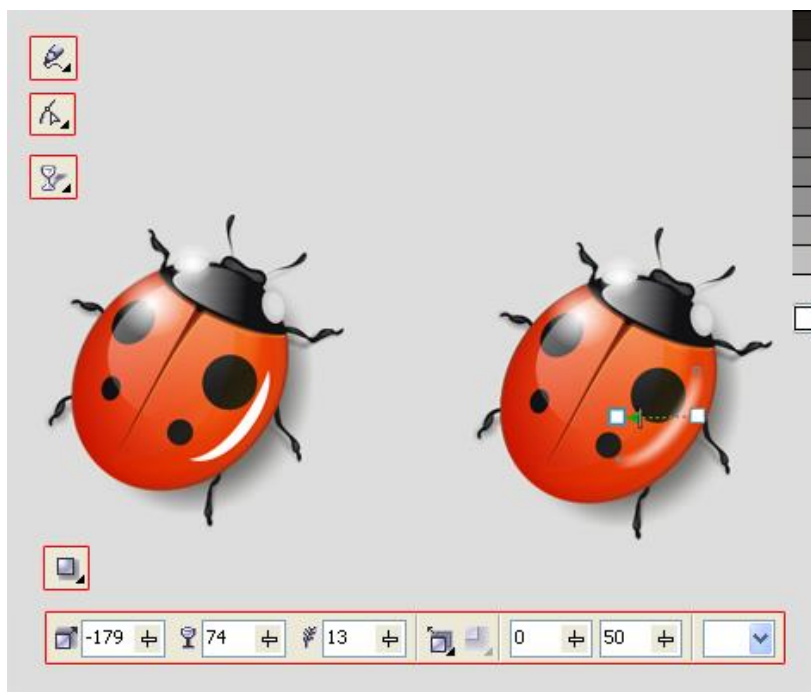
11) Нарисуем блик небольшого размера круглой формы. Инструментом Эллипс рисуем окружность, применяем на нее **полупрозрачность фонтанную, эллиптического типа**,. Размещаем блик на спинке.







12) Нарисуем еще один блик с другой стороны спинки букашки. Рисуем объект, как показано на рисунке, применяем на него тень, в верхнем меню, в настройках тени выставляем **цвет тени – белый**, далее к этому же объекту применяем Интерактивную **полупрозрачность, базового типа 100%**. Группируем все объекты и божья коровка готова!



Результат:



## Практическая работа №44 (ПР44)

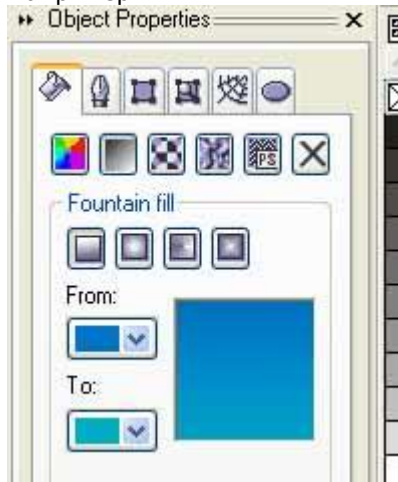
**Тема:** Векторный графический редактор CorelDraw

**Цель:** Рисуем объемный шар с надписью

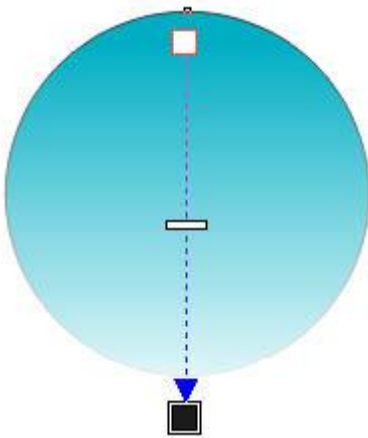
Начнем с того, что создадим несколько окружностей.

Выбираем инструмент **Эллипс (F7)**, и, удерживая клавишу **Ctrl**, рисуем две окружности, диаметр одной из которых чуть больше второй. Меньшую окружность заливаем чтобы в верхней части окружности цвет был чуть темнее, чем в нижней.

Например:

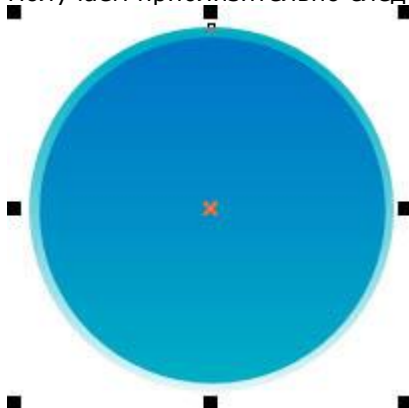


Большую окружность заливаем тем же цветом, что и меньшую окружность внизу. После этого применяем **Прозрачность**

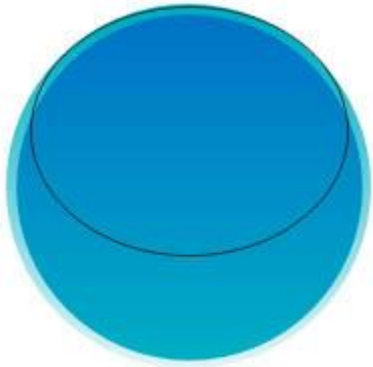


Убираем у обеих окружностей контур. Большая окружность должна находиться ПОД меньшей.

Получаем приблизительно следующее:



Снова берем инструмент **Эллипс (F7)** и рисуем овал, как на рисунке



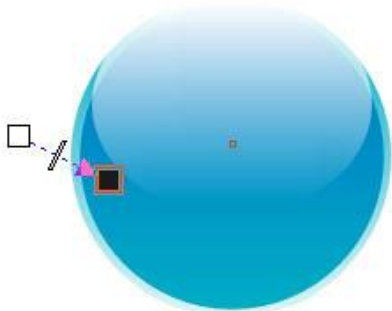
Заливаем овал белым, убираем контур и применяем **Прозрачность**:



Следующий шаг: делаем внутреннюю тень. Снова берем инструмент **Эллипс (F7)** и рисуем окружность. Заливаем ее любым цветом (например черным):



Нажимаем **Ctrl+Page Down**. Далее начинается самое интересное: используя **Прозрачность**, мы **полностью** обесцвечиваем черный круг, вот так:



Применяем **Тень** с настройками:



С помощью следующей окружности мы сделаем тень, падающую от шарика.



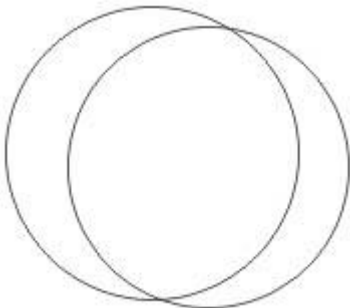
Последовательность действий такая же: обесцвечиваем с помощью **Прозрачности** и применяем **Тень** с настройками:



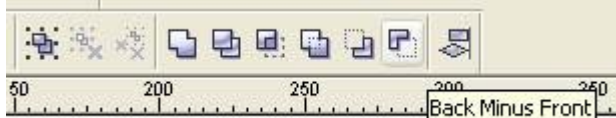
Нажимаем **Shift+Page Down**. Получаем приблизительно вот так:



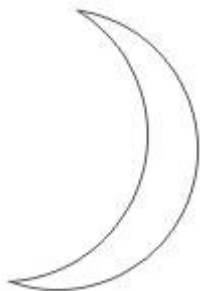
Следующий шаг – создание бликов. Рисуем 2 окружности:



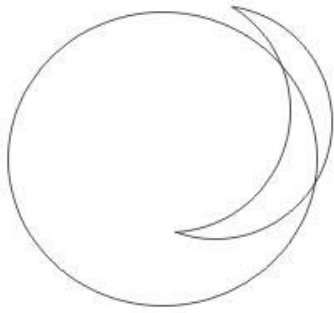
Выделяем их обе и нажимаем **Задние минус передние**.



Получаем приблизительно следующее:



Рисуем еще одну окружность



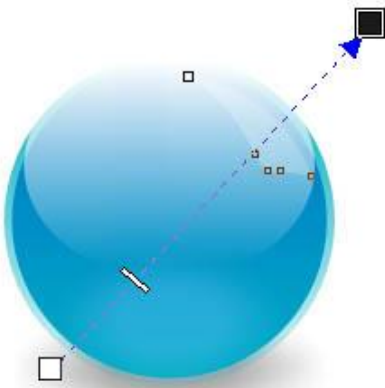
И снова жмем **Задние минус передние**:



Повторяем эту последовательность действий еще раз:



Вот и готов будущий блик. Заливаем его белым цветом, убираем контур, применяем **Прозрачность** следующим образом:



Дублируем блик, немного его уменьшаем и помещаем ближе к центру шарика:



Таким же образом делаем блики в нижней части шарика (если есть желание и терпение):



И последнее: надпись! Берем инструмент **Текст (F8)** , пишем необходимые слова и располагаем на шарике:



Применяем к надписи инструмент **Оболочка**



Заливаем белым цветом и делаем полупрозрачным с помощью **Прозрачность**. Вот и готовый шарик!



## **3.2 ПМ.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ**

### **УП.01.01. УЧЕБНАЯ ПРАКТИКА К МДК.01.01. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ**

#### **5 СЕМЕСТР**

#### **Практическая работа №1**

##### **1. Текст задания**

В соответствии с выбранным вариантом, разработать оконное приложение с использованием низкоуровневых средств программирования WinAPI, представляющее собой форму для ввода данных документа с последующей целью последующего сохранения и загрузки его данных в текстовом файле.

##### **Предъявляемые требования к приложению:**

- Приложение должно иметь интуитивный пользовательский интерфейс;
- Приложение должно быть разработано с использованием элементов управления WinAPI;
- Ввод данных пользователем производится в текстовые поля;
- Если какое-либо из полей не имеет данных, необходимо вывести сообщение с ошибкой об этом.
- Данные читаются из файла и сохраняются в файл, указанный пользователем для записи и чтения файла должны быть использованы стандартные диалоги.;
- Диалоги загрузки и сохранения вызываются из соответствующих пунктов меню («Загрузить файл» и «Сохранить файл») и по нажатию горячих клавиш (Ctrl+O и Ctrl+S);
- Для выхода из программы создать пункт меню «Выход» и сочетание клавиш Alt+F4;
- Для очистки полей ввода создать кнопку «Очистить»;
- Приложения должны быть персонализированы. По пункту меню "О программе" должен отображаться диалог с информацией об авторе, группе и назначении программы.

##### **Этапы работы:**

1. Выбор варианта и разбор постановки задачи 1.
2. Определение полей документа, предназначенных для заполнения.
3. Написание кода для задачи 1.
4. Проектирование формата файлов для хранения документов.
5. Разработка алгоритма программы 1 в виде блок-схемы.
6. Тестирование кода и отладка недочетов, выявленных при тестировании кода задачи 1
7. Разработка документации по задаче 1

##### **2. Пример документа (задания)**

В качестве примера рассмотрим **Журнал регистрации приходных и расходных кассовых документов** В оригинале документ выглядит так:

Унифицированная форма № КО-3  
Утвержденная постановлением Госкомстата  
России от 18.08.98 г. №88

Код
0310003
XXXXXXX

Форма по ОКУД \_\_\_\_\_ по ОКПО \_\_\_\_\_

ООО «Кондитер» \_\_\_\_\_  
(организация)

\_\_\_\_\_ (структурное подразделение)

**Журнал регистрации**  
**ПРИХОДНЫХ И РАСХОДНЫХ КАССОВЫХ ДОКУМЕНТОВ**  
\_\_\_\_\_ 2013 г. \_\_\_\_\_

бухгалтер \_\_\_\_\_ Ольга Петровна Иванова \_\_\_\_\_  
(должность) (фамилия, имя, отчество)

В данном документе видно, что есть поля для заполнения.

Унифицированная форма № КО-3  
Утвержденная постановлением Госкомстата  
России от 18.08.98 г. №88

Код
0310003
XXXXXXX

Форма по ОКУД \_\_\_\_\_ по ОКПО \_\_\_\_\_

ООО «Кондитер» \_\_\_\_\_  
(организация)

\_\_\_\_\_ (структурное подразделение)

**Журнал регистрации**  
**ПРИХОДНЫХ И РАСХОДНЫХ КАССОВЫХ ДОКУМЕНТОВ**  
\_\_\_\_\_ 2013 г. \_\_\_\_\_

бухгалтер \_\_\_\_\_ Ольга Петровна Иванова \_\_\_\_\_  
(должность) (фамилия, имя, отчество)

Это следующие поля:

- Организация
- ОКПО
- Структурное подразделение



- Год
- Должность
- ФИО

Чистый бланк журнала выглядит следующим образом:

Унифицированная форма № КО-3  
Утвержденная постановлением Госкомстата  
России от 18.08.98 г. №88

Код
0310003

Форма по ОКУД \_\_\_\_\_ по ОКПО \_\_\_\_\_

\_\_\_\_\_  
(организация)

\_\_\_\_\_  
(структурное подразделение)

**Журнал регистрации**  
**ПРИХОДНЫХ И РАСХОДНЫХ КАССОВЫХ ДОКУМЕНТОВ**

\_\_\_\_\_

\_\_\_\_\_  
(должность)                      \_\_\_\_\_  
(фамилия, имя, отчество)

### 3. Пример программы

1. Создайте окно для сохранения информации из документа в текстовый файл.

The screenshot shows a window titled "Загрузка и сохранение данных документа" with a menu bar containing "Файл" and "Справка". The main area contains the following text and form fields:

**Ввод данных Журнала регистрации:  
ПРИХОДНЫХ И РАСХОДНЫХ КАССОВЫХ ДОКУМЕНТОВ**

Организация

ОКПО

Структурное подразделение

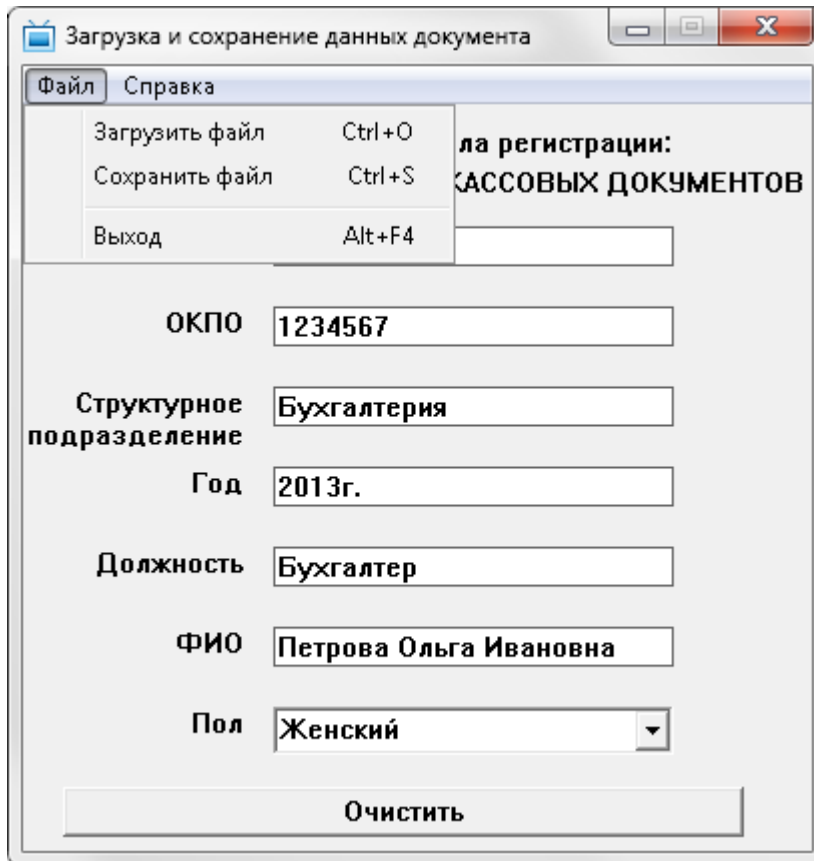
Год

Должность

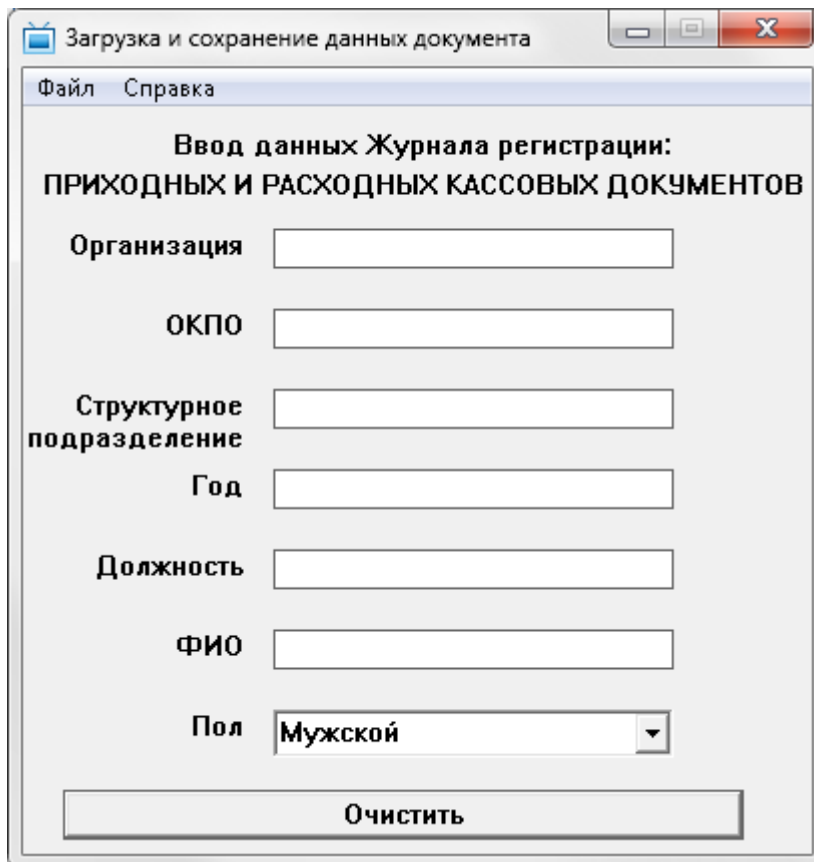
ФИО

Пол

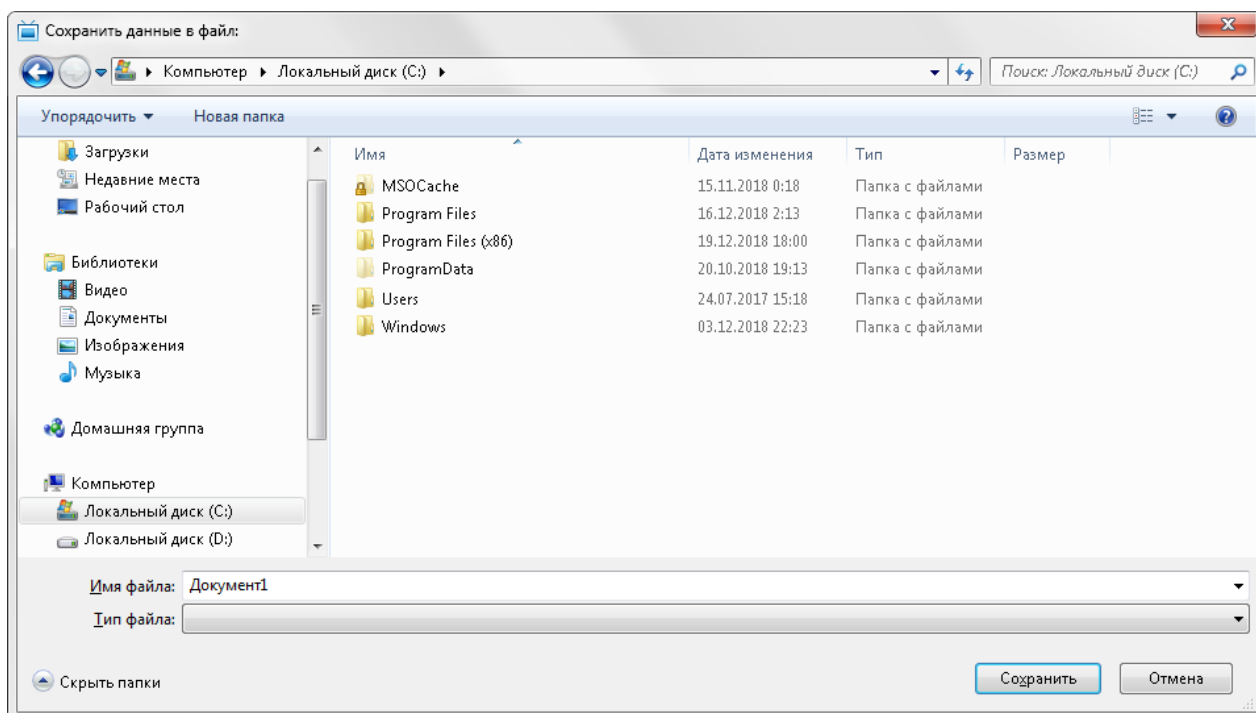
2. Добавьте подменю «Файл» с пунктами «Загрузить файл» и «Сохранить файл», а также пункт «Выход» для закрытия приложения



3. На кнопку «Очистить» реализуйте очистку полей



4. На пункт меню «Сохранить файл» вызовите стандартный диалог сохранения и запишите данные из полей в файл.



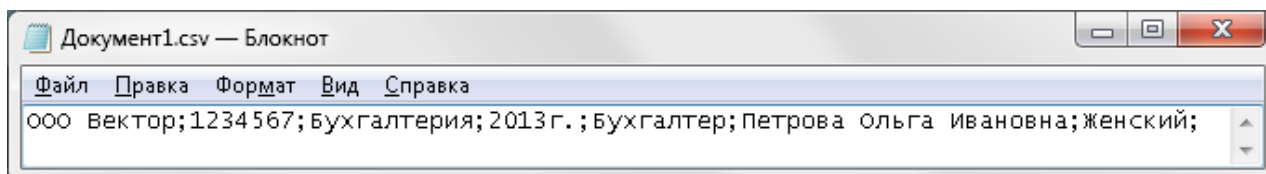
Также необходимо спроектировать формат хранения. При сохранении необходимо учитывать тот факт, что данные потом будут читаться. Поэтому лучше всего выбрать текстовые файлы. Для удобного программного разбора лучше использовать формат CSV.

Кроме программного разбора формат CSV удобен тем, что он легко просматривается и редактируется в табличном процессоре Microsoft Excel

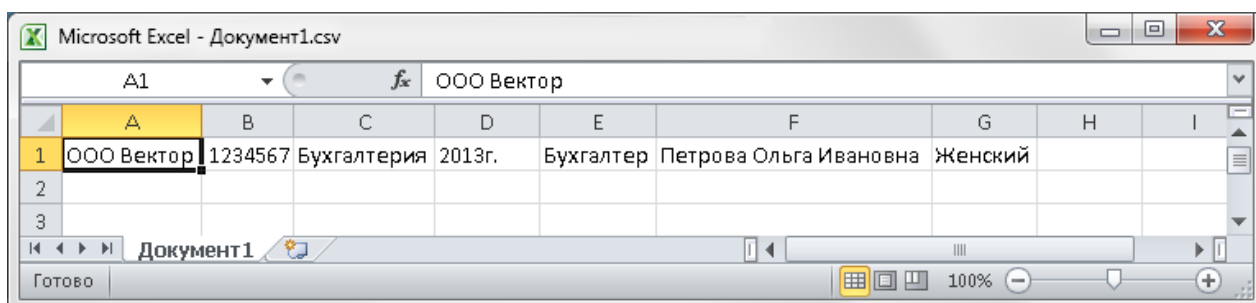
**Файл при сохранении должен выглядеть одним из следующих способов:**

### 1) Сохранение в строку (через точку с запятой)

Записываться данные будут следующим образом:

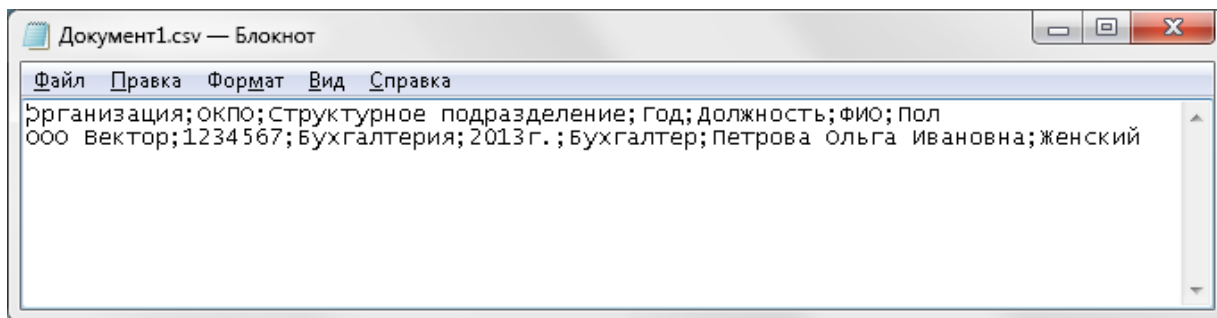


В Excel такой файл будет отображаться следующим образом:

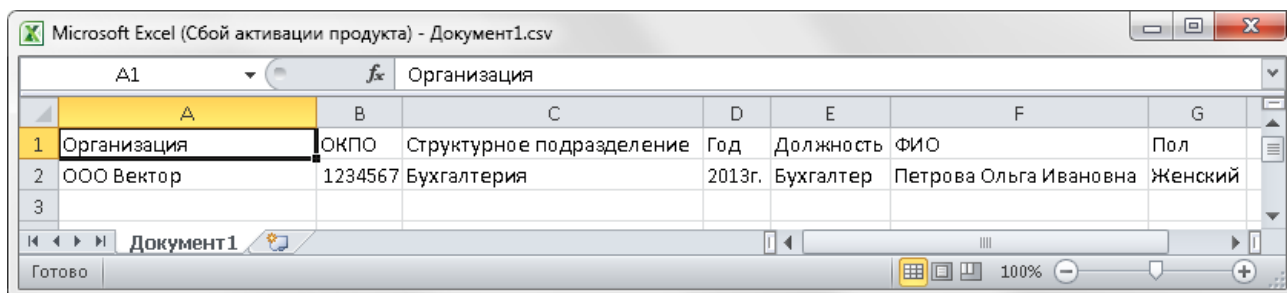


### 2) Сохранение в строку с наименованиями полей

Записываться данные будут следующим образом:

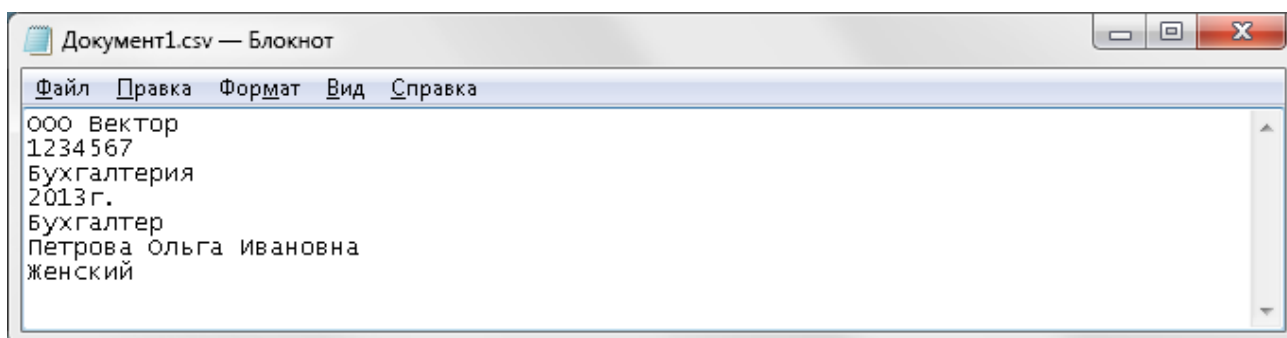


В Excel такой файл будет отображаться следующим образом:

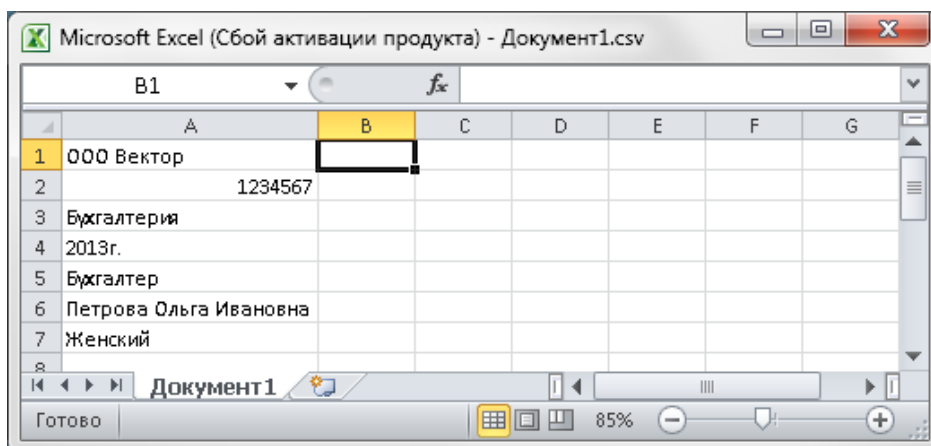


### 3) Сохранение в столбец (через символ перехода строки - '\n')

Записываться данные будут следующим образом:

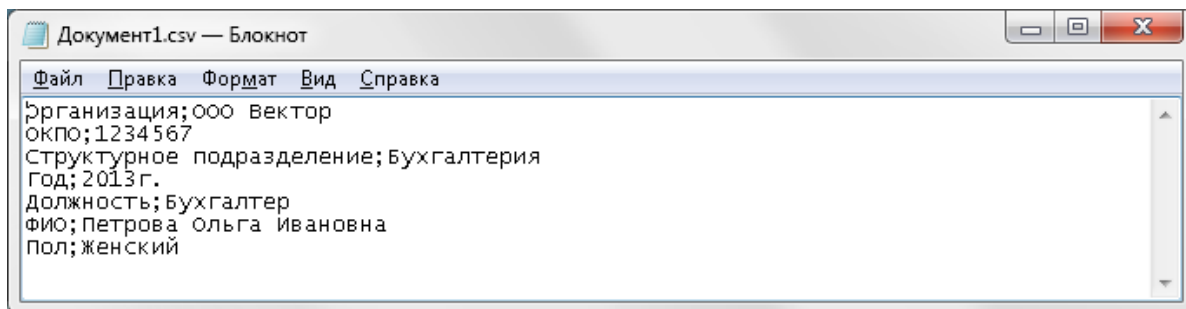


В Excel такой файл будет отображаться следующим образом:

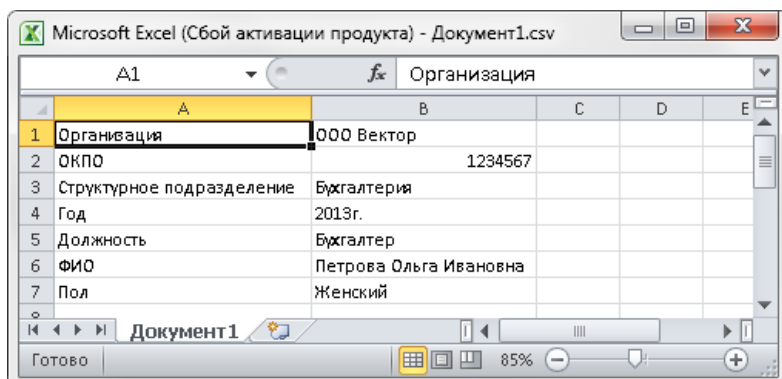


### 4) Сохранение в столбец с наименованием полей

Записываться данные будут следующим образом:

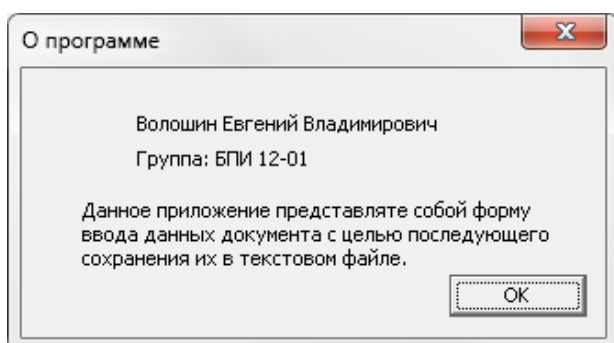
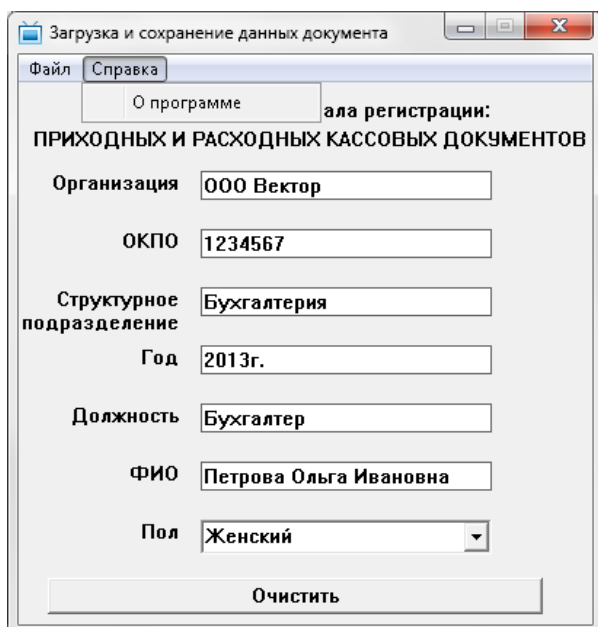


В Excel такой файл будет отображаться следующим образом:



**5. На пункт меню «Загрузить файл» вызовите стандартный диалог загрузки и прочитайте данные из файла обратно в поля.**

**6. Добавьте подменю «Справка» с пунктом «О программе» и вызовите диалог с информацией об авторе, группе и назначении программы.**



**7. Реализуйте вызов пунктов «Загрузить файл» и «Сохранить файл» на горячие клавиши Ctrl+O и Ctrl+S соответственно.**

## **Практическая работа №2**

### **Техническая документация**

**Напишите техническую документацию по разработанной программе 1.**

**Содержание документации к программе:**

1. Пояснительная записка
  - 1.1 Постановка задачи
  - 1.2 Алгоритм
  - 1.3 Входные данные
  - 1.4 Выходные данные
  - 1.5 Варианты для отладки алгоритма
  - 1.6 Требования к реализации алгоритма
  - 1.7 Блок-схема главной программы WinMain
  - 1.8 Блок-схема оконной функции WndProc
  - 1.9 Блок-схема сохранения данных
  - 1.10 Блок-схема загрузки данных
  - 1.11 Блок-схемы дополнительных функций... *(если есть)*
2. Описание программы
  - 2.1 Директивы препроцессора
  - 2.2 Оконные классы
  - 2.3 Обрабатываемые сообщения
  - 2.4 Глобальные переменные и дескрипторы
  - 2.5 Используемые типы данных
  - 2.6 Используемые функции
  - 2.7 Внешние зависимости
  - 2.8 Описание графического интерфейса программы
  - 2.9 Тестирование программы

## **Практическая работа №3**

### **1. Текст задания**

В соответствии с выбранным вариантом, разработать оконное приложение для отображения информации из текстового файла на окне в наглядном виде. Прочитать содержимое файла и отобразить текст полей на бланке документа. В последствии сохранить готовый бланк документа в виде изображения.

**Предъявляемые требования к приложению:**

- Приложение должно иметь интуитивный пользовательский интерфейс;
- Приложение должно быть разработано с использованием элементов управления WinAPI;
- Изображение документа должно быть отображено на фоне главного окна. Изображение должно иметь формат BMP;
- На пункт «Загрузить файл» должны загружаться данные из файла на изображение документа;
- На пункт «Сохранить изображение» должно сохраняться готовое изображение документа;
- Диалоги загрузки и сохранения вызываются из соответствующих пунктов меню («Загрузить файл» и «Сохранить изображение») и по нажатию горячих клавиш (Ctrl+O и Ctrl+S);
- Для выхода из программы создать пункт меню «Выход» и сочетание клавиш Alt+F4;

- Приложения должны быть персонализированы. По пункту меню "О программе" должен отображаться диалог с информацией об авторе, группе и назначении программы.

### Этапы работы:

8. Разбор постановки задачи 2.
9. Написание кода для задачи 2.
10. Разработка алгоритма программы 2 в виде блок-схемы.
11. Тестирование кода и отладка недочетов, выявленных при тестировании кода задачи 2
12. Разработка документации по задаче 2

## 2. Пример программы

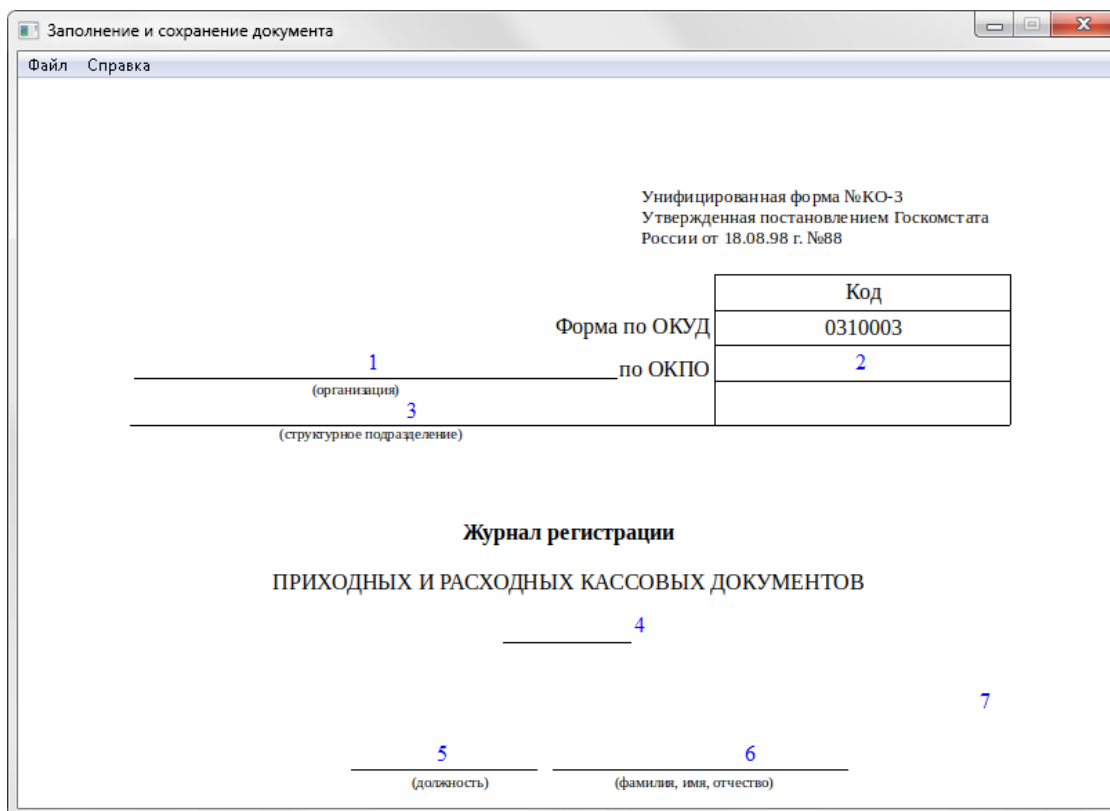
### 1. Выведите в окно изображение документа

Основное требование при этом - читабельность текста на изображении. Большие изображения можно уменьшить. Мелкие изображения, можно увеличить, либо самостоятельно найти изображение в лучшем качестве.

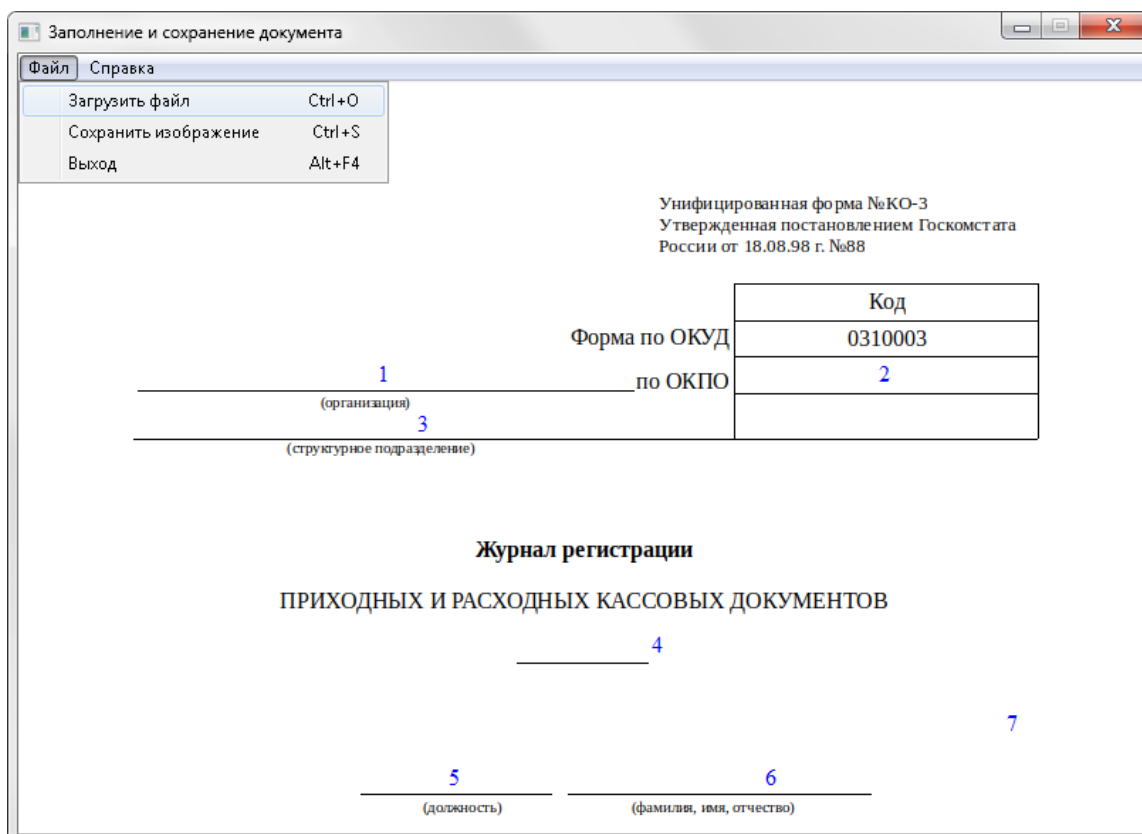
The screenshot shows a window titled "Заполнение и сохранение документа" with a menu bar containing "Файл" and "Справка". The main content area displays a document form. At the top right, it reads: "Унифицированная форма №КО-3 Утвержденная постановлением Госкомстата России от 18.08.98 г. №88". Below this is a table with two columns: "Форма по ОКУД" and "Код". The value "0310003" is entered in the "Код" cell. To the left of the table, there are two horizontal lines for text entry, labeled "(организация)" and "(структурное подразделение)", with the text "по ОКПО" positioned between them. In the center of the form, the title "Журнал регистрации ПРИХОДНЫХ И РАСХОДНЫХ КАССОВЫХ ДОКУМЕНТОВ" is displayed. At the bottom, there are two horizontal lines for text entry, labeled "(должность)" and "(фамилия, имя, отчество)".

Форма по ОКУД	Код
	0310003

### 2. Выведите предварительный статический текст в нужных позициях

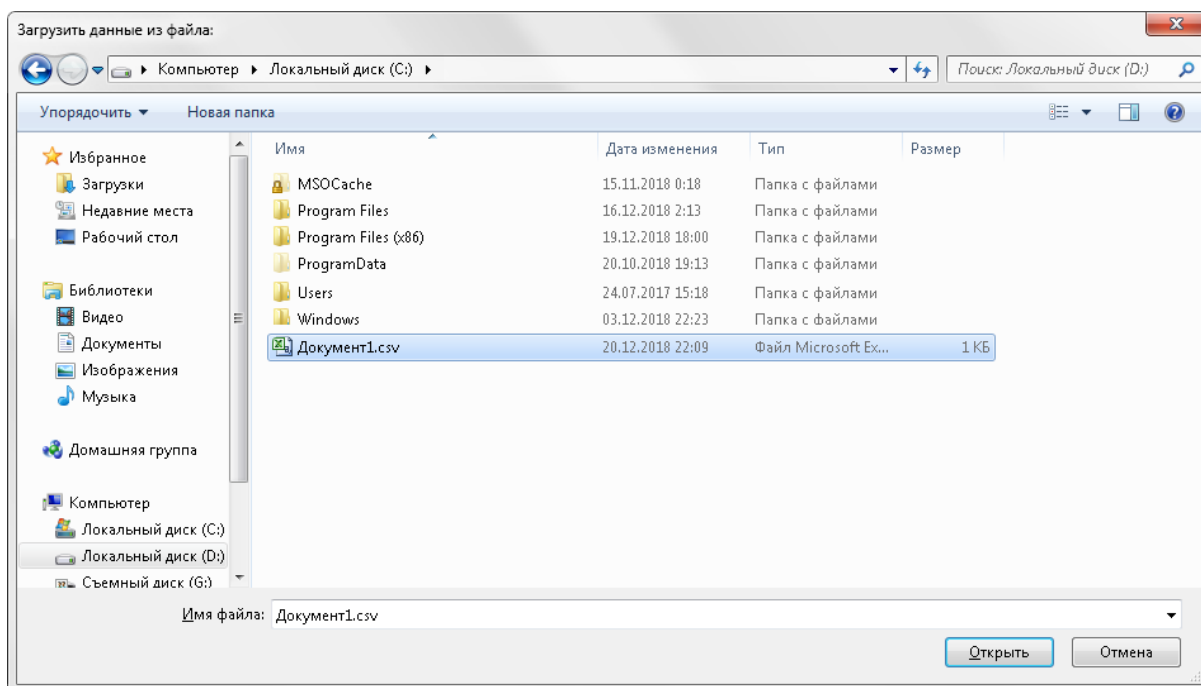


3. Добавьте подменю «Файл» с пунктами «Загрузить файл» и «Сохранить изображение», а также пункт «Выход» для закрытия приложения



4. На пункт меню «Загрузить файл» вызовите стандартный диалог загрузки и прочитайте данные из файла в статический текст на картинке





5. На пункт меню «Сохранить изображение» вызовите стандартный диалог сохранения и сохраните готовое изображение, отображенное на фоне главного окна
6. Добавьте подменю «Справка» с пунктом «О программе» и вызовите диалог с информацией об авторе, группе и назначении программы.
7. Реализуйте вызов пунктов «Загрузить файл» и «Сохранить изображение» на горячие клавиши Ctrl+O и Ctrl+S соответственно.
8. Напишите техническую документацию по разработанной программе 2.

## Практическая работа №4

### Техническая документация

Напишите техническую документацию по разработанной программе 2.

Содержание документации к программе:

1. Пояснительная записка
  - 1.1 Постановка задачи
  - 1.2 Алгоритм
  - 1.3 Входные данные
  - 1.4 Выходные данные
  - 1.5 Варианты для отладки алгоритма
  - 1.6 Требования к реализации алгоритма
  - 1.7 Блок-схема главной программы WinMain
  - 1.8 Блок-схема оконной функции WndProc
  - 1.9 Блок-схема функции сохранения изображения
  - 1.10 Блок-схема функции загрузки данных
  - 1.11 Блок-схема функции графических параметров MyStaticWndProc
  - 1.11 Блок-схемы дополнительных функций... *(если есть)*
2. Описание программы
  - 2.1 Директивы препроцессора
  - 2.2 Оконные классы
  - 2.3 Обрабатываемые сообщения
  - 2.4 Глобальные переменные и дескрипторы
  - 2.5 Используемые типы данных
  - 2.6 Используемые функции
  - 2.7 Внешние зависимости
  - 2.8 Описание графического интерфейса программы
  - 2.9 Тестирование программы

**3.3 ПМ.02 РАЗРАБОТКА И АДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ**  
**УП.02.01. УЧЕБНАЯ ПРАКТИКА К МДК 02.01. ИНФОКОММУНИКАЦИОННЫЕ**  
**СИСТЕМЫ И СЕТИ**  
**5 СЕМЕСТР**

**Практическая работа №1. IP-адресация**

**Цели работы:**

- научиться определять адрес подсети и адрес хоста по маске подсети;
- научиться определять количество и диапазон адресов возможных узлов в подсетях;
- научиться структурировать сети с использованием масок.

**Связь с проектом**

Для успешного решения задач администрирования необходимо хорошо разбираться в системе IP-адресации. Знание принципов использования масок и структуризации сетей поможет грамотно решать многие вопросы настройки локальной сети.

**Задание 1.** Определить, находятся ли два узла А и В в одной подсети или в разных подсетях, если адреса компьютера А и компьютера В соответственно равны: 26.219.123.6/10 и 26.218.102.31, маска подсети 255.192.0.0.

**Указания к выполнению**

1. Переведите адреса компьютеров и маску в двоичный вид.
2. Для получения двоичного представления номеров подсетей обоих узлов выполните операцию логического умножения AND над IP-адресом и маской каждого компьютера.
3. Двоичный результат переведите в десятичный вид.
4. Сделайте вывод.

Процесс решения можно записать следующим образом:

Компьютер А:  
 IP-адрес: 26.219.123.6 =                    00011010. 11011011. 01111011. 00000110  
 Маска подсети: 255.192.0.0 =            11111111. 11000000. 00000000. 00000000

Компьютер В:  
 IP-адрес: 26.218.102.31 =                00011010. 11011010. 01100110. 00011111  
 Маска подсети: 255.192.0.0 =            11111111. 11000000. 00000000. 00000000

Получаем номер подсети, выполняя операцию AND над IP-адресом и маской подсети.  
 Компьютер А:

AND	00011010. 11011011. 01111011. 00000110
	11111111. 11000000. 00000000. 00000000
	00011010. 11000000. 00000000. 00000000
	26                    192                    0                    0

Компьютер В:

AND	00011010. 11011010. 01100110. 00011111
	11111111. 11000000. 00000000. 00000000
	00011010. 11000000. 00000000. 00000000
	26                    192                    0                    0

**Ответ:** номера подсетей двух IP-адресов совпадают, значит компьютеры А и В находятся в одной подсети. Следовательно, между ними возможно установить прямое соединение без применения шлюзов.

**Задание 2.** Определить количество и диапазон IP-адресов в подсети, если известны номер подсети и маска подсети.

Номер подсети – 26.219.128.0, маска подсети – 255.255.192.0.

**Указания к выполнению**

1. Переведите номер и маску подсети в двоичный вид.

Номер подсети: 26.219.128.0 = 00011010. 11011011. 10000000. 00000000  
Маска подсети: 255.255.192.0 = 11111111. 11111111. 11000000. 00000000

2. По маске определите количество бит, предназначенных для адресации узлов (их значение равно нулю). Обозначим их буквой К.

3. Общее количество адресов равно  $2^K$ . Но из этого числа следует исключить комбинации, состоящие из всех нулей или всех единиц, так как данные адреса являются особыми. Следовательно, общее количество узлов подсети будет равно  $2^K - 2$ .

В рассматриваемом примере  $K = 14$ ,  $2^K - 2 = 16\ 382$  адресов.

4. Чтобы найти диапазон IP-адресов нужно найти начальный и конечный IP-адреса подсети. Для этого выделите в номере подсети те биты, которые в маске подсети равны единице. Это разряды, отвечающие за номер подсети. Они будут совпадать для всех узлов данной подсети, включая начальный и конечный:

Номер подсети: 26.219.128.0 = 00011010. 11011011. 10000000. 00000000  
Маска подсети: 255.255.192.0 = 11111111. 11111111. 11000000. 00000000

5. Чтобы получить начальный IP-адрес подсети нужно невыделенные биты в номере подсети заполнить *нулями*, за исключением крайнего правого бита, который должен быть равен единице. Полученный адрес будет первым из допустимых адресов данной подсети:

Начальный адрес: 26.219.128.1 = 00011010. 11011011. 10000000. 00000001  
Маска подсети: 255.255.192.0 = 11111111. 11111111. 11000000. 00000000

6. Чтобы получить конечный IP-адрес подсети нужно невыделенные биты в номере подсети заполнить *единицами*, за исключением крайнего правого бита, который должен быть равен нулю. Полученный адрес будет последним из допустимых адресов данной подсети:

Конечный адрес: 26.219.191.254 = 00011010. 11011011. 10111111. 11111110  
Маска подсети: 255.255.192.0 = 11111111. 11111111. 11000000. 00000000

**Ответ:** Для подсети 26.219.128.0 с маской 255.255.192.0:  
количество возможных адресов: 16 382,  
диапазон возможных адресов: 26.219.128.1 – 26.219.191.254.

**Задание 3.** Организации выделена сеть класса С: 212.100.54.0/24. Требуется разделить данную сеть на 4 подсети с количеством узлов в каждой не менее 50. Определить маски и количество возможных адресов новых подсетей.

**Указания к выполнению**

1. В сетях класса С (маска содержит 24 единицы – 255.255.255.0) под номер узла отводится 8 бит, т. е. сеть может включать  $2^8 - 2 = 254$  узла.

2. Требование деления на 4 подсети по 50 узлов в каждой может быть выполнено:  $4 \cdot 50 = 200 < 254$ . Однако число узлов в подсети должно быть кратно степени двойки. Относительно 50 ближайшая большая степень –  $2^6 = 64$ . Следовательно, для номера узла нужно отвести 6 бит, вместо 8, а маску расширить на 2 бита – до 26 бит (см. рис. 3).

3. В этом случае вместо одной сети с маской 255.255.255.0 образуется 4 подсети с маской 255.255.255.192 и количеством возможных адресов в каждой – 62 (не забывайте про два особых адреса).

4. Номера новых подсетей отличаются друг от друга значениями двух битов, отведенных под номер подсети. Эти биты равны 00, 01, 10, 11.

**Ответ:** маска подсети – 255.255.255.192, количество возможных адресов – 62.

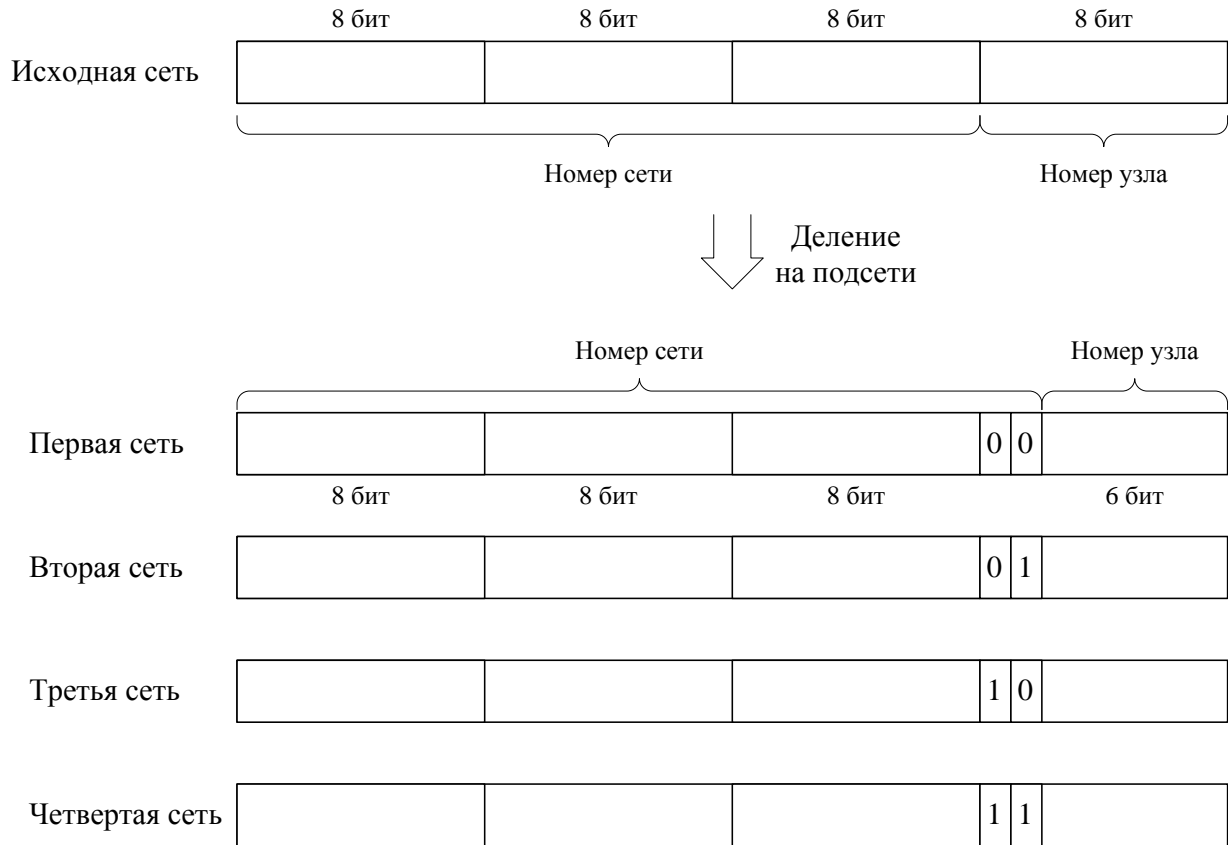


Рис. 3. Адреса подсетей после деления

### Самостоятельная работа

**Задание 1.** Определить, находятся ли два узла А и В в одной подсети или в разных подсетях.

1. IP-адрес компьютера А: 94.235.16.59;  
IP-адрес компьютера В: 94.235.23.240;  
Маска подсети: 255.255.240.0.
2. IP-адрес компьютера А: 131.189.15.6;  
IP-адрес компьютера В: 131.173.216.56;  
Маска подсети: 255.248.0.0.
3. IP-адрес компьютера А: 215.125.159.36;  
IP-адрес компьютера В: 215.125.153.56;  
Маска подсети: 255.255.224.0.

**Задание 2.** Определить количество и диапазон адресов узлов в подсети, если известны номер подсети и маска подсети.

1. Номер подсети: 192.168.1.0, маска подсети: 255.255.255.0.
2. Номер подсети: 110.56.0.0, маска подсети: 255.248.0.0.
3. Номер подсети: 88.217.0.0, маска подсети: 255.255.128.0.

**Задание 3.** Определить маску подсети, соответствующую указанному диапазону IP-адресов.

1. 119.38.0.1 – 119.38.255.254.
2. 75.96.0.1 – 75.103.255.254.
3. 48.192.0.1 – 48.255.255.254.

**Задание 4.** Организации выделена сеть класса В: 185.210.0.0/16. Определить маски и количество возможных адресов новых подсетей в каждом из следующих вариантов деления на подсети:

1. Число подсетей – 256, число узлов – не менее 250.
2. Число подсетей – 16, число узлов – не менее 4000.

#### **Задание 5 Ответьте на вопросы**


1. Может ли быть IP-адрес узла таким? Укажите неверные варианты IP-адрес. Ответ обоснуйте.

1. 192.168.255.0
2. 167.234.56.13
3. 224.0.5.3
4. 172.34.267.34
5. 230.0.0.7
6. 160.54.255.255

2. Может ли маска подсети быть такой? Укажите неверные варианты. Ответ обоснуйте.

1. 255.254.128.0
2. 255.255.252.0
3. 240.0.0.0
4. 255.255.194.0
5. 255.255.128.0
6. 255.255.255.244
7. 255.255.255.255

#### **Требования к отчету**

 В отчете запишите ответы на задания самостоятельной работы. Обоснуйте каждый шаг получения результата, аналогично тому, как это сделано в примерах.

### **Практическая работа №2. Ознакомление с интерфейсом программы Netemul.** **Соединение ЭВМ в сеть**

**Цель работы:** Ознакомиться с основами работы с программным эмулятором ЛВС NetEmul. Научиться строить простейшие модели ЛВС. Уяснить разницу в построении ЛВС на концентраторах и коммутаторах.

#### **Ход работы:**

Для запуска эмулятора Netemul необходимо либо воспользоваться соответствующим пунктом главного меню операционной системы, либо выполнить в терминале команду netemul.

#### **1.2. Теоретический материал**

##### ***1.2.1. Запуск эмулятора NetEmul***

Для запуска эмулятора NetEmul необходимо либо воспользоваться соответствующим пунктом главного меню операционной системы, либо выполнить в терминале команду netemul.

### 1.3. Порядок выполнения лабораторной работы

С помощью инструмента «Вставить текстовую надпись» добавить на рабочее поле эмулятора надпись, содержащую:

1. Номер группы.
2. ФИО студентов, выполняющих работу.
3. Номер варианта согласно номеру студента (одного из студентов бригады) в журнале.

Таблица 1.1 Варианты задания (указаны согласно номеру студента в журнале)

№	Адрес сети/маска	№	Адрес сети/маска	№	Адрес сети/маска
1	10.0.1.0/27	11	10.1.1.64/27	21	10.2.1.128/27
2	10.0.2.32/27	12	10.1.2.96/27	22	10.2.2.160/27
3	10.0.3.64/27	13	10.1.3.128/27	23	10.2.3.192/27
4	10.0.4.96/27	14	10.1.4.160/27	24	10.2.4.224/27
5	10.0.5.128/27	15	10.1.5.192/27	25	10.2.5.0/27
6	10.0.6.160/27	16	10.1.6.224/27	26	10.2.6.32/27
7	10.0.7.192/27	17	10.1.7.0/27	27	10.2.7.64/27
8	10.0.8.224/27	18	10.1.8.32/27	28	10.2.8.96/27
9	10.0.9.0/27	19	10.1.9.64/27	29	10.2.9.128/27
10	10.0.0.32/27	20	10.1.0.96/27	30	10.2.0.160/27

#### 1.3.1. Соединение двух ЭВМ напрямую

**1. Выбрать исходные данные для выполнения работы согласно своему варианту.**

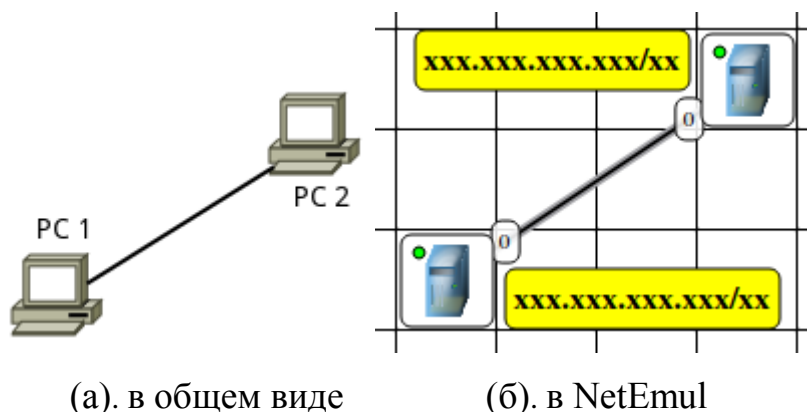


Рис. 1.1. Схема модели ЛВС при соединении двух ЭВМ напрямую

2. Добавить на рабочее поле эмулятора два компьютера (см. рис. 1.1), используя кнопку «Добавить компьютер» на панели инструментов.

3. Соединить добавленные компьютеры как показано на рис. 1.1. Для этого

а) нажать кнопку «Создать соединение» на панели инструментов;

б) навести указатель на один из компьютеров;

в) зажав ЛКМ, перевести курсор на второй компьютер — за курсором от первого компьютера должна тянуться прямая линия;

г) отпустить ЛКМ — после этого должно появиться окно начальных настроек с выбором соединяемых интерфейсов;

д) подтвердить соединение между интерфейсами eth0 и eth0, нажав «Соединить»;

е) если все сделано правильно, то компьютеры теперь соединены, на каждом конце соединения показан номер используемого интерфейса (в данном случае — 0), а индикатор соединения на иконке компьютера сменил цвет с красного на желтый (соединение есть, но интерфейсы не настроены).

4. Настроить компьютеры, задав каждому IP-адрес и маску подсети в соответствии с вариантом. Для этого а) выбрать инструмент «Перемещение объектов» на панели инструментов;

б) выделить первый компьютер щелчком ЛКМ;

в) вызвать контекстное меню щелчком ПКМ и выбрать пункт «Интерфейсы»;

г) в появившемся окне указать в соответствующих полях IP-адрес и маску подсети;

д) подтвердить ввод последовательным нажатием кнопок «Применить» и «ОК»;

е) если все сделано правильно, то индикатор соединения на иконке компьютера должен сменить цвет с желтого на зеленый (соединение есть, и интерфейсы настроены);

ж) добавить возле каждого компьютера надпись с его IP-адресом и маской подсети как показано на рис. 1.1.

5. Проверить работоспособность построенной модели ЛВС, передав пакеты от одного компьютера до другого. Для этого

а) выбрать инструмент «Отправить данные» на панели инструментов;

б) под курсором (на рабочем поле программы) должен появиться красный круг;

в) привести курсор с красным кругом на передающий компьютер и нажать ЛКМ;

г) в появившемся окне «Отправка» указать: протокол TCP, размер данных 5 KB;

д) нажать «Далее» — окно пропадет, а кружок под курсором сменит цвет на зеленый;

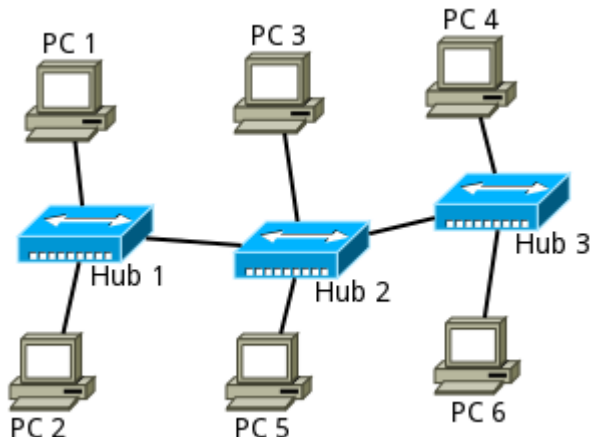
е) привести курсор с зеленым кругом на принимающий компьютер и нажать ЛКМ;

ж) в появившемся окне подтвердить интерфейс на принимающем компьютере eth0, нажав «Отправка»;

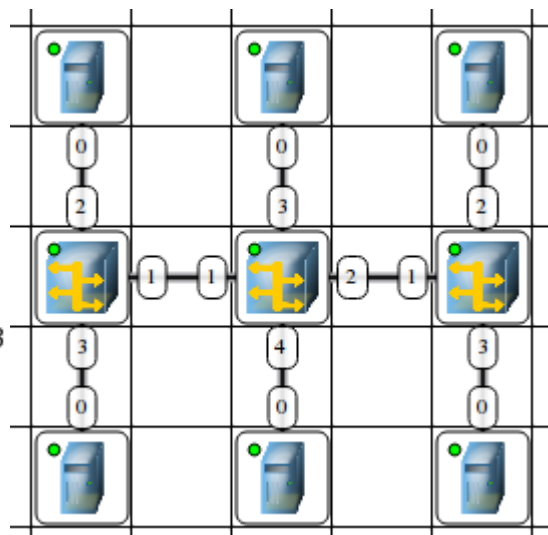
з) проследить за перемещением пакетов.

### 1.3.2. Построение ЛВС на концентраторах





(а). в общем виде



(б). в NetEmul

**Рис. 1.2. Схема модели ЛВС на основе концентраторов**

Выбрать исходные данные для выполнения работы согласно своему варианту.

2. Добавить на рабочее поле эмулятора шесть компьютеров и три концентратора как показано на рис. 1.2.

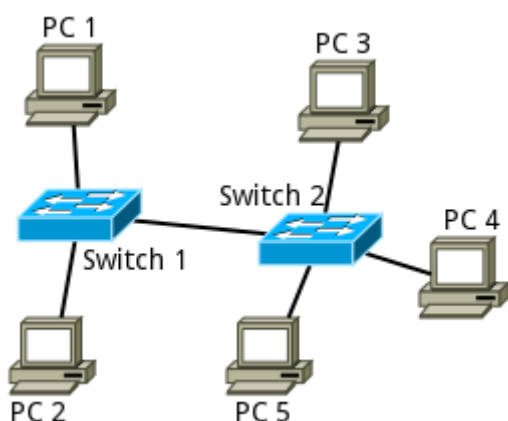
3. Соединить устройства как показано на рис. 1.2.

4. Настроить компьютеры, задав каждому IP-адрес и маску подсети в соответствии с вариантом.

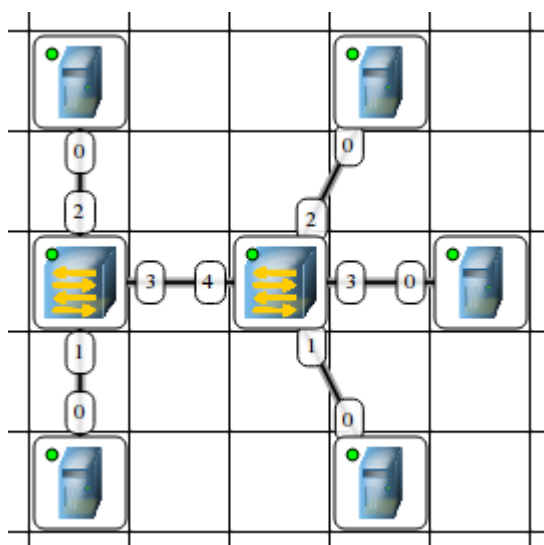
5. Добавить возле каждого компьютера надпись с его IP-адресом и маской подсети.

6. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от одного компьютера до другого. Проследить за перемещением пакетов и сделать выводы об особенностях работы ЛВС на основе концентраторов.

### 1.3.3. Построение ЛВС на коммутаторах



(а). в общем виде



(б). в NetEmul

### Рис. 1.3. Схема модели ЛВС на основе коммутаторов

1. Выбрать исходные данные для выполнения работы согласно своему варианту.
2. Добавить на рабочее поле эмулятора пять компьютеров и два коммутатора как показано на рис. 1.3.
3. Соединить устройства как показано на рис. 1.3.
4. Настроить компьютеры, задав каждому IP-адрес и маску подсети в соответствии с вариантом.
5. Добавить возле каждого компьютера надпись с его IP-адресом и маской подсети.
6. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от одного компьютера до другого. Проследить за перемещением пакетов и сделать выводы об особенностях работы ЛВС на основе коммутаторов.

После выполнения работы продемонстрировать преподавателю работоспособность построенной модели.

Проект сохранить для отчета.

#### 1.4. Форма представления и содержание отчета

***Отчет в формате DOC или PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.***

#### ***Содержание отчета***

- 1. Заголовок согласно приложению.***
- 2. Цель работы.***
- 3. По каждому пункту лабораторной должна быть приведена схема модели с указанием IP-адресов устройств и номеров интерфейсов.***
- 4. По каждому пункту лабораторной должны быть приведены выводы по работе.***

#### 1.5. Контрольные вопросы

- 1. Что такое IP-адрес?***
- 2. Что такое маска подсети?***
- 3. Как работает концентратор?***
- 4. Как работает коммутатор?***

## **Практическая работа №3. Использование маршрутизаторов. Статическая маршрутизация**

**Цель работы:** Ознакомиться с работой маршрутизаторов. Научиться формировать статические маршруты и прописывать их в таблицы маршрутизации сетевых устройств

### 2.2. Теоретический материал

#### 2.2.1. Запуск эмулятора NetEmul

Для запуска эмулятора NetEmul необходимо либо воспользоваться соответствующим пунктом главного меню операционной системы, либо выполнить в терминале команду netemul.

### 2.3. Порядок выполнения лабораторной работы

С помощью инструмента «Вставить текстовую надпись» добавить на рабочее поле эмулятора надпись, содержащую:

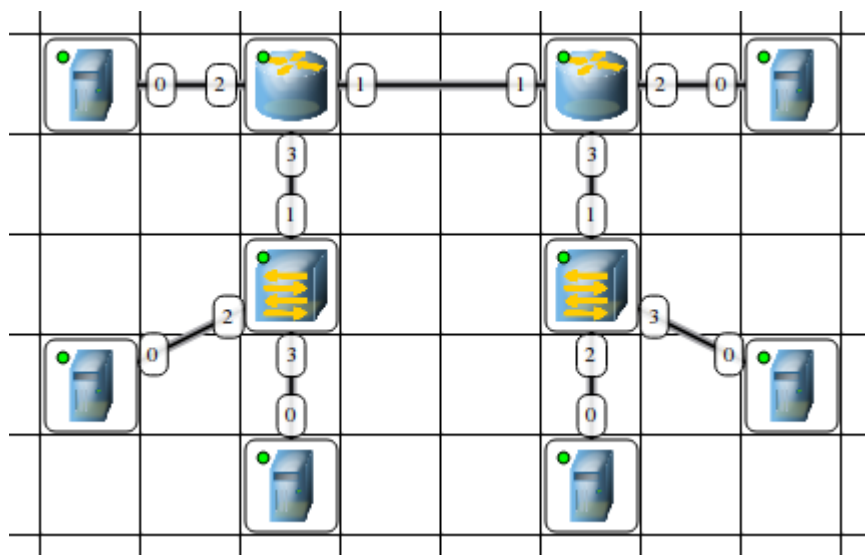
1. Номер группы.
2. ФИО студента, выполняющего работу.
3. Вариант задания, согласно номеру студента в журнале.

*Таблица 1.1 Варианты задания (указаны согласно номеру студента в журнале)*

<b>Вариант задания</b>	<b>Диапазон адресов 1</b>	<b>Диапазон адресов 2</b>
1	10.1.0.0/16	192.168.1.0/24
2	172.20.2.0/24	192.168.0.0/16
3	10.3.0.0/24	172.16.0.0/12
4	192.168.4.0/24	10.4.0.0/16
5	172.30.5.0/24	10.0.0.0/8
6	10.6.0.0/16	192.168.0.0/16
7	10.7.0.0/24	172.17.7.0/24
8	172.18.8.0/24	192.168.0.0/16
9	192.168.9.0/24	10.0.0.0/8
10	192.168.10.0/24	10.10.0.0/16
11	172.21.11.0/24	192.168.0.0/16
12	10.12.0.0/16	192.168.0.0/16
13	192.168.13.0/24	10.13.0.0/16
14	172.24.0.0/12	10.14.0.0/16
15	10.15.0.0/24	192.168.0.0/16
16	192.168.16.0/24	10.16.0.0/16
17	172.27.17.0/24	10.0.0.0/24
18	10.18.0.0/16	192.168.0.0/16
19	192.168.19.0/24	10.19.0.0/24
20	192.168.20.0/24	172.20.1.0/24
21	172.21.1.0/24	10.0.0.0/16
22	172.23.22.0/24	10.10.0.0/16
23	192.168.0.0/16	172.23.23.0/24
24	10.24.0.0/16	192.168.24.0/24
25	172.27.0.0/16	10.25.1.0/24
26	192.168.26.0/24	10.26.1.0/24
27	10.27.0.0/24	172.17.110.0/24

28	172.28.0.0/24	192.168.0.0/16
29	10.29.0.0/16	192.168.0.0/16
30	172.29.30.0/24	10.30.0.0/16
31	192.168.31.0/24	10.31.0.0/16
32	172.16.32.0/24	192.168.32.0/24
33	10.0.73.0/24	172.31.33.0/24
34	192.168.2.0/24	172.18.34.0/24
35	10.35.0.0/16	192.168.20.0/24
36	172.19.34.0/24	10.36.36.0/24
37	10.0.37.0/24	192.168.3.0/24
38	172.21.34.0/24	192.168.38.0/24
39	192.168.39.0/24	10.39.0.0/16
40	172.23.34.0/24	10.40.0.0/16
41	172.17.41.0/24	192.168.41.0/24
42	10.42.42.0/24	172.18.0.0/16
43	172.26.34.0/24	10.9.10.0/24
44	172.22.44.0/24	192.168.44.0/24
45	172.28.34.0/24	10.45.1.0/24
46	10.46.0.0/16	192.168.46.0/24
47	10.10.47.0/24	172.24.7.0/24
48	172.31.34.0/24	10.48.0.0/16
49	10.0.0.0/16	172.16.1.0/24
50	10.50.0.0/24	192.168.50.0/24
51	172.27.34.0/24	10.1.51.0/24
52	10.52.0.0/16	192.168.52.0/24
53	192.168.53.0/24	172.25.53.0/24
54	172.25.0.0/16	192.168.0.0/16
55	192.168.5.0/24	172.26.0.0/16
56	10.56.0.0/16	172.26.56.0/24
57	172.27.57.0/24	10.57.0.0/16
58	172.28.58.0/24	10.58.1.0/24
59	172.30.0.0/24	10.59.0.0/16
60	10.0.1.0/24	172.20.60.0/24
61	10.61.0.0/16	172.21.61.0/24
62	172.30.34.0/24	10.62.0.0/16
63	10.63.0.0/16	172.23.63.0/24
64	192.168.64.0/24	172.24.64.0/24
65	172.29.34.0/24	192.168.65.0/24
66	10.66.66.0/24	192.168.66.0/24
67	172.27.67.0/24	192.168.67.0/24
68	192.168.68.0/24	172.30.68.0/24
69	10.69.0.0/16	172.29.96.0/24
70	172.20.70.0/24	10.70.0.0/16
71	172.22.34.0/24	192.168.71.0/24
72	10.72.0.0/16	172.22.72.0/24

### ***2.3.1. Построение модели сети***



(a). в NetEmul

Рис. 2.1. Связь сетей посредством маршрутизаторов

1. Выбрать исходные данные для выполнения работы согласно своему варианту.
2. Используя соответствующие инструменты на панели эмулятора, построить сеть в соответствии с рис. 2.1. В свойствах каждого маршрутизатора необходимо указать количество интерфейсов, равное 4.

3. Настроить интерфейсы компьютеров и маршрутизаторов, задав каждому IP-адрес и маску подсети в соответствии с вариантом. Добавить возле каждого компьютера и интерфейса роутера надписи с их IP-адресом и маской подсети.

4. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от одного устройства до другого в пределах одной подсети.

После выполнения работы продемонстрировать преподавателю работоспособность построенной модели.

Проект сохранить для отчета.

## Контрольный пример

**Задание.** Создать сеть согласно рис. 26. Прописать IP-адреса всем интерфейсам. Использовать адреса 3 классов сетей (A, B, C). В свойствах компьютеров прописать IP-адреса шлюзов. В маршрутизаторах установить протокол RIP. Проверить работоспособность сети. Оформить отчет.

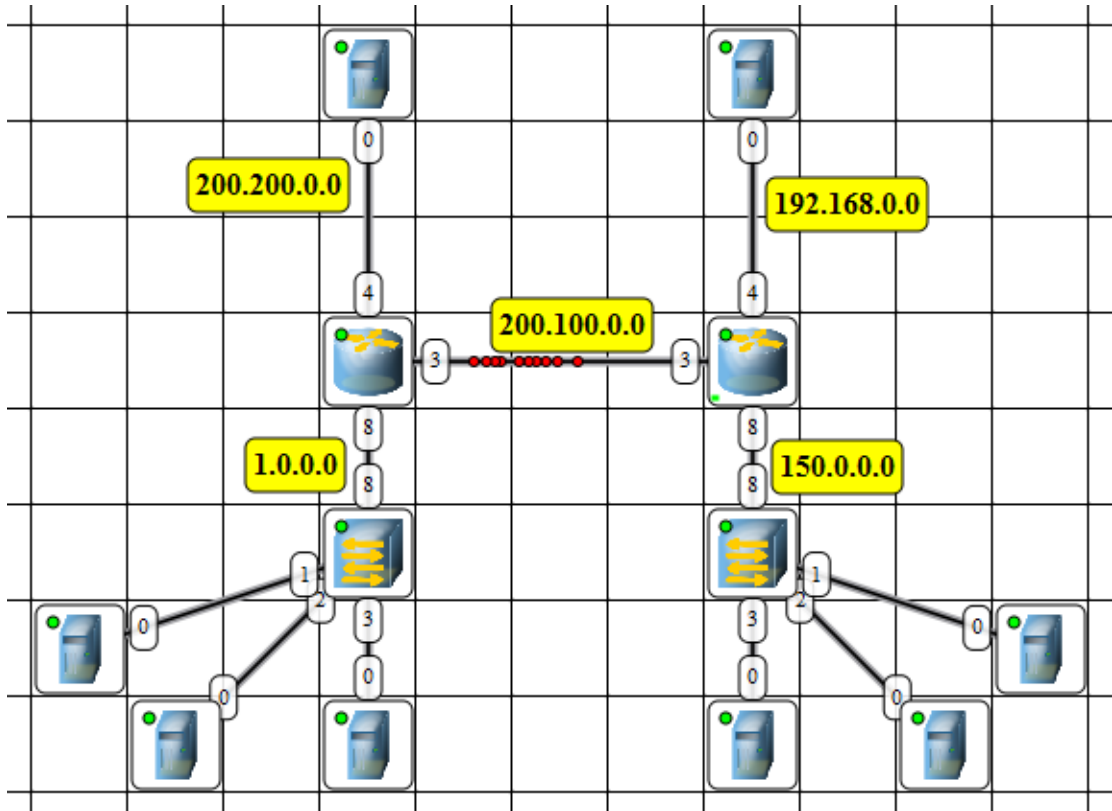


Рис. 2.2. Связь сетей посредством маршрутизаторов (пример)

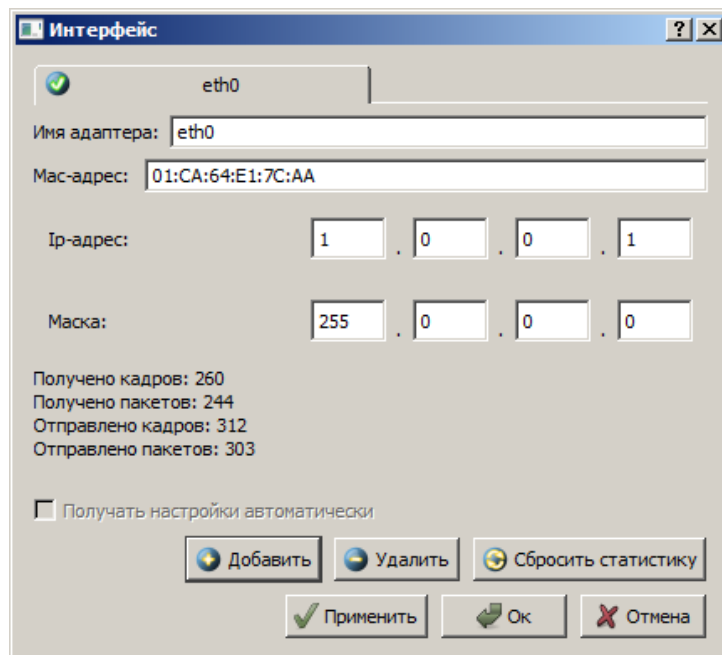
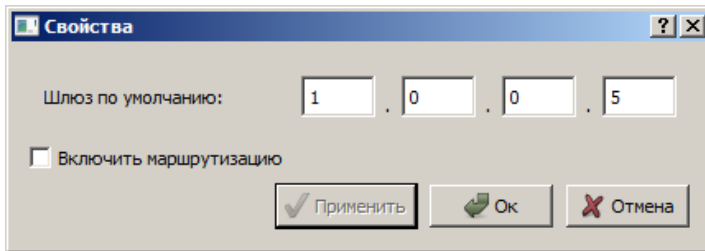
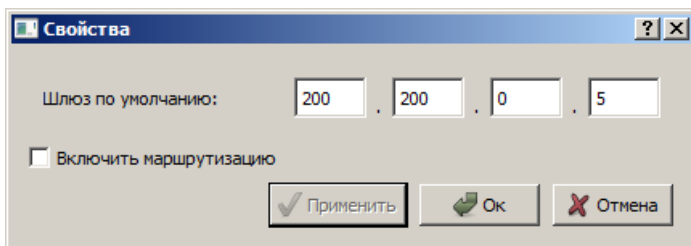
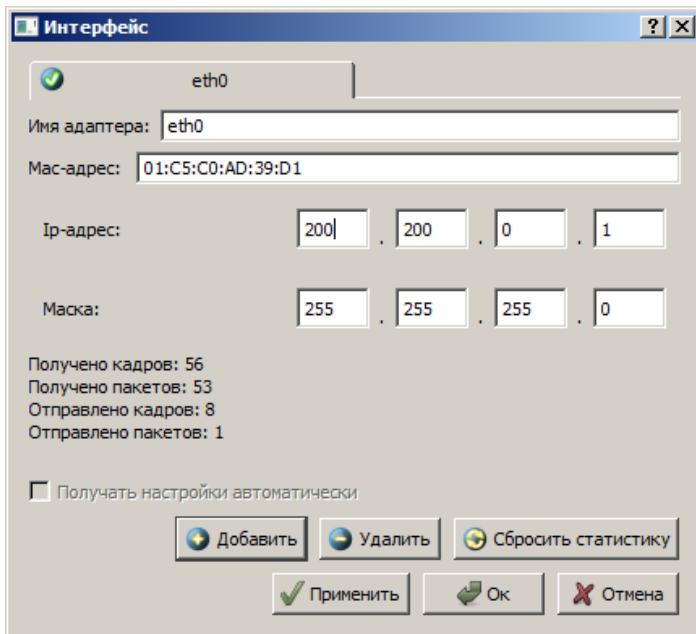


Рис. 2.3. Настройка интерфейса ПК1 в подсети 1.0.0.0

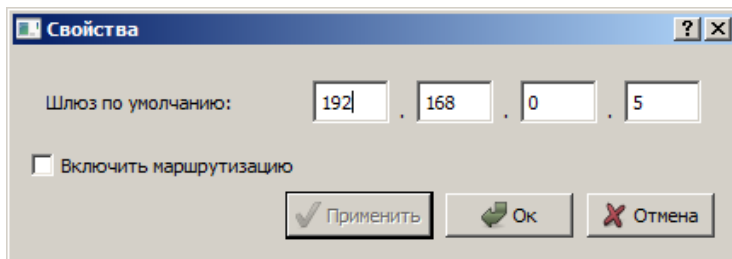
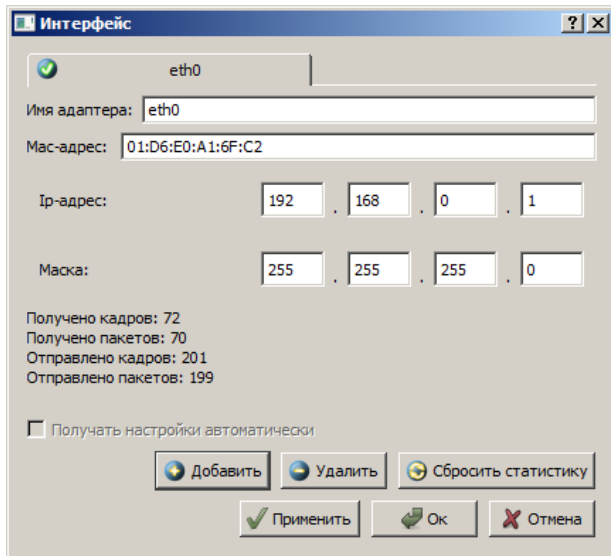
*Остальным 2м компьютерам присваиваете IP-адреса аналогично.  
Далее каждому компьютеру подсети 1.0.0.0 присваиваете  
следующий шлюз:*



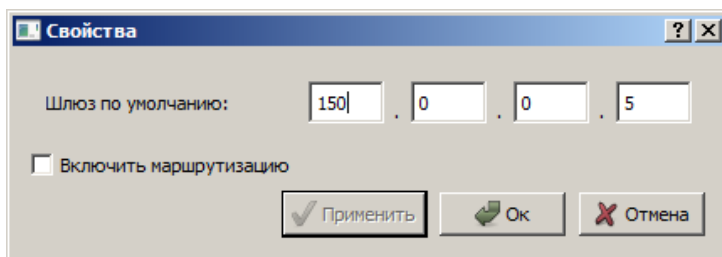
*Далее настройте подсеть 200.200.0.0 для этого присваиваете  
настройки интерфейса и шлюз:*



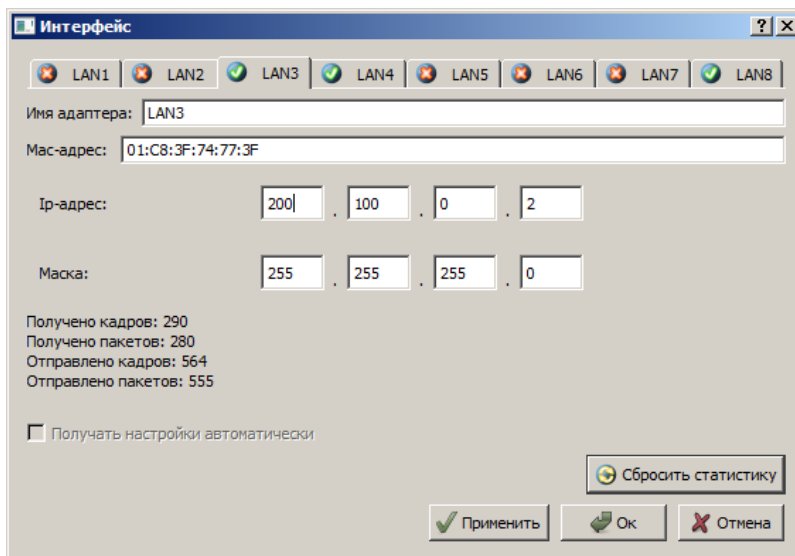
*Аналогично настройте подсеть 192.168.0.0 для этого  
присваиваете настройки интерфейса и шлюз:*



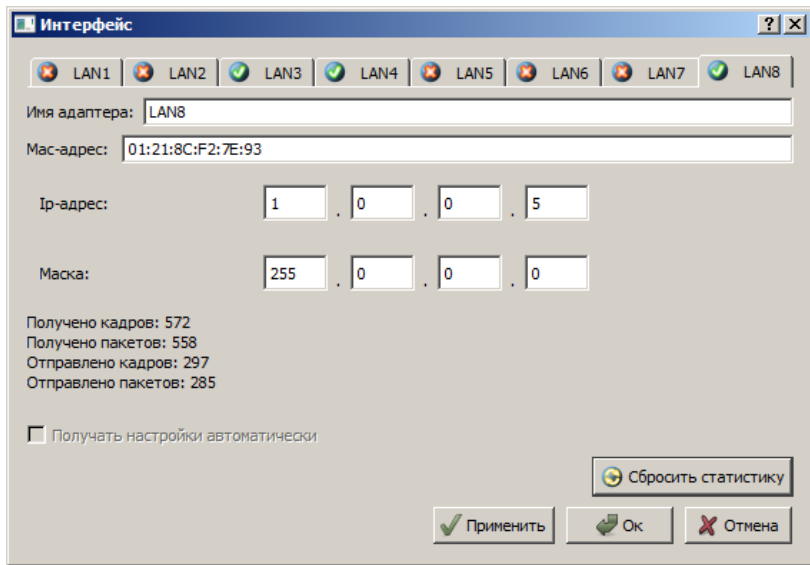
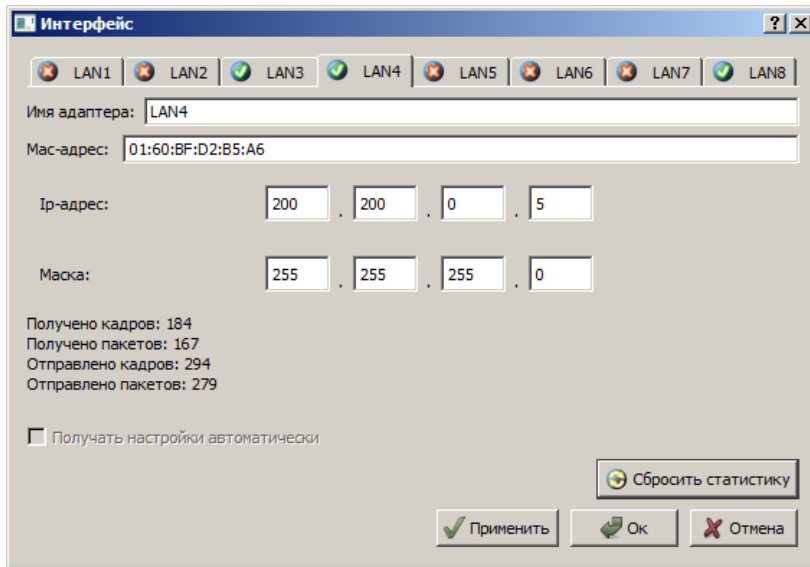
*Далее настройте подсеть 150.0.0.0 для этого присваиваете настройки интерфейса в диапазоне 150.0.0.1-150.0.0.3 и шлюз:*



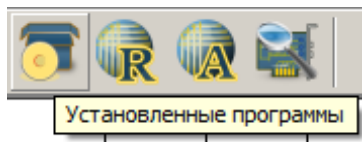
*Настройте левый маршрутизатор:*



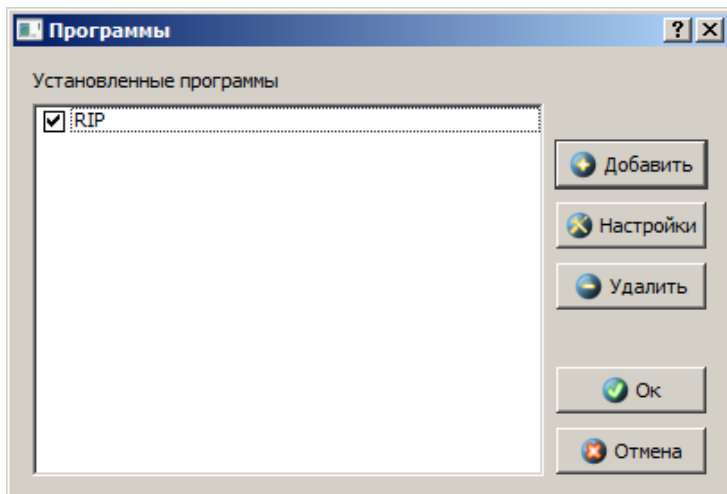




*В маршрутизаторах установить протокол RIP, для этого запустите*



*пункт меню , нажмите по кнопке добавить и выберите из списка протокол RIP:*



## Настройте правый маршрутизатор:

Интерфейс

LAN1 LAN2 LAN3 LAN4 LAN5 LAN6 LAN7 LAN8

Имя адаптера: LAN3

Мас-адрес: 01:65:08:FB:F9:2F

Ир-адрес: 200 . 100 . 0 . 3

Маска: 255 . 255 . 255 . 0

Получено кадров: 580  
Получено пакетов: 571  
Отправлено кадров: 306  
Отправлено пакетов: 296

Получать настройки автоматически

Сбросить статистику

Применить Ок Отмена

Интерфейс

LAN1 LAN2 LAN3 LAN4 LAN5 LAN6 LAN7 LAN8

Имя адаптера: LAN4

Мас-адрес: 01:EE:B9:B4:60:74

Ир-адрес: 192 . 168 . 0 . 5

Маска: 255 . 255 . 255 . 0

Получено кадров: 370  
Получено пакетов: 362  
Отправлено кадров: 267  
Отправлено пакетов: 259

Получать настройки автоматически

Сбросить статистику

Применить Ок Отмена

Интерфейс

LAN1 LAN2 LAN3 LAN4 LAN5 LAN6 LAN7 LAN8

Имя адаптера: LAN8

Мас-адрес: 01:3C:7B:AA:09:1A

Ир-адрес: 150 . 0 . 0 . 5

Маска: 255 . 255 . 0 . 0

Получено кадров: 62  
Получено пакетов: 54  
Отправлено кадров: 775  
Отправлено пакетов: 769

Получать настройки автоматически

Сбросить статистику

Применить Ок Отмена

**Таблица маршрутизации левого роутера будет выглядеть следующим образом:**

Таблица маршрутизации

	Адрес назначения	Маска	Шлюз	Интерфейс	Метрика	Источник
1	192.168.0.0	255.255.255.0	200.100.0.3	200.100.0.2	1	RIP
2	200.100.0.0	255.255.255.0	200.100.0.2	200.100.0.2	0	Подключена
3	200.200.0.0	255.255.255.0	200.200.0.5	200.200.0.5	0	Подключена
4	150.0.0.0	255.255.0.0	200.100.0.3	200.100.0.2	1	RIP
5	1.0.0.0	255.0.0.0	1.0.0.5	1.0.0.5	0	Подключена

Адрес назначения:  .  .  .

Маска:  .  .  .

Шлюз:  .  .  .

Интерфейс:

Метрика:

*Таблица маршрутизации правого роутера будет выглядеть следующим образом:*

	Адрес назначения	Маска	Шлюз	Интерфейс	Метрика	Источник
1	192.168.0.0	255.255.255.0	192.168.0.5	192.168.0.5	0	Подключена
2	200.100.0.0	255.255.255.0	200.100.0.3	200.100.0.3	0	Подключена
3	200.200.0.0	255.255.255.0	200.100.0.2	200.100.0.3	1	RIP
4	150.0.0.0	255.255.0.0	150.0.0.5	150.0.0.5	0	Подключена
5	1.0.0.0	255.0.0.0	200.100.0.2	200.100.0.3	1	RIP

Адрес назначения:  .  .  .

Маска:  .  .  .

Шлюз:  .  .  .

Интерфейс:

Метрика:

## 2.4. Форма представления и содержание отчета

***Отчет в формате DOCили PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.***

### ***Содержание отчета***

- 1. Заголовок согласно приложению.***
- 2. Цель работы.***
- 3. По каждому пункту лабораторной должна быть приведена схема модели с указанием IP-адресов устройств и номеров интерфейсов.***
- 4. По каждому пункту лабораторной должны быть приведены выводы по работе.***

## 2.5. Контрольные вопросы

- 1. Что такое IP-адрес?***
- 2. Что такое маска подсети?***

*3. Как работает маршрутизатор?*

*4. Принципы статической маршрутизации?*

## **Практическая работа №4. Разрешение адресов по протоколу ARP. ARP-спуфинг**

**4.1 Цель работы:** Ознакомиться с механизмом работы протокола ARP. Научиться формировать и отправлять пользовательские пакеты. Ознакомиться с журналом работы сетевого устройства в эмуляторе. Научиться проводить сетевую атаку вида ARP-спуфинг.

### 4.2. Теоретический материал

#### **4.2.1. Запуск эмулятора NetEmul**

Для запуска эмулятора NetEmul необходимо либо воспользоваться соответствующим пунктом главного меню операционной системы, либо выполнить в терминале команду netemul.

#### **4.2.2. Протокол ARP**

ARP (Address Resolution Protocol — протокол определения адреса) — протокол в компьютерных сетях, предназначенный для определения MAC-адреса сетевого устройства по известному IP-адресу.

Наибольшее распространение ARP получил благодаря повсеместности сетей IP, построенных поверх Ethernet, поскольку в подавляющем большинстве случаев при таком сочетании используется ARP. В семействе протоколов IPv6 протокола ARP не существует, его функции возложены на ICMPv6.

Описание протокола было опубликовано в ноябре 1982 г. в RFC 826.

ARP был спроектирован для случая передачи IP-пакетов через сегмент Ethernet. При этом общий принцип, предложенный для ARP, был использован и для сетей других типов.

Существуют следующие типы сообщений ARP: запрос ARP (ARP-request) и ответ ARP (ARP-reply). Система-отправитель при помощи запроса ARP запрашивает физический адрес системы-получателя. Ответ (физический адрес узла-получателя) приходит в виде ответа ARP.

**Принцип работы протокола:** узел (хост *A*), которому нужно выполнить отображение IP-адреса на MAC-адрес, формирует ARP-запрос, вкладывает его в кадр протокола канального уровня, указывая в нем известный IP-адрес (хост *B*), и рассылает запрос широкоэвещательно (в поле MAC-адрес назначения заголовка Ethernet указывается широкоэвещательный MAC-адрес FF:FF:FF:FF:FF:FF). Все узлы локальной сети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел (хост *B*) формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес и отправляет его уже направленно, так как в ARP запросе отправитель (хост *A*) указывает свой локальный адрес.

Схема работы показана на рис. 3.1.

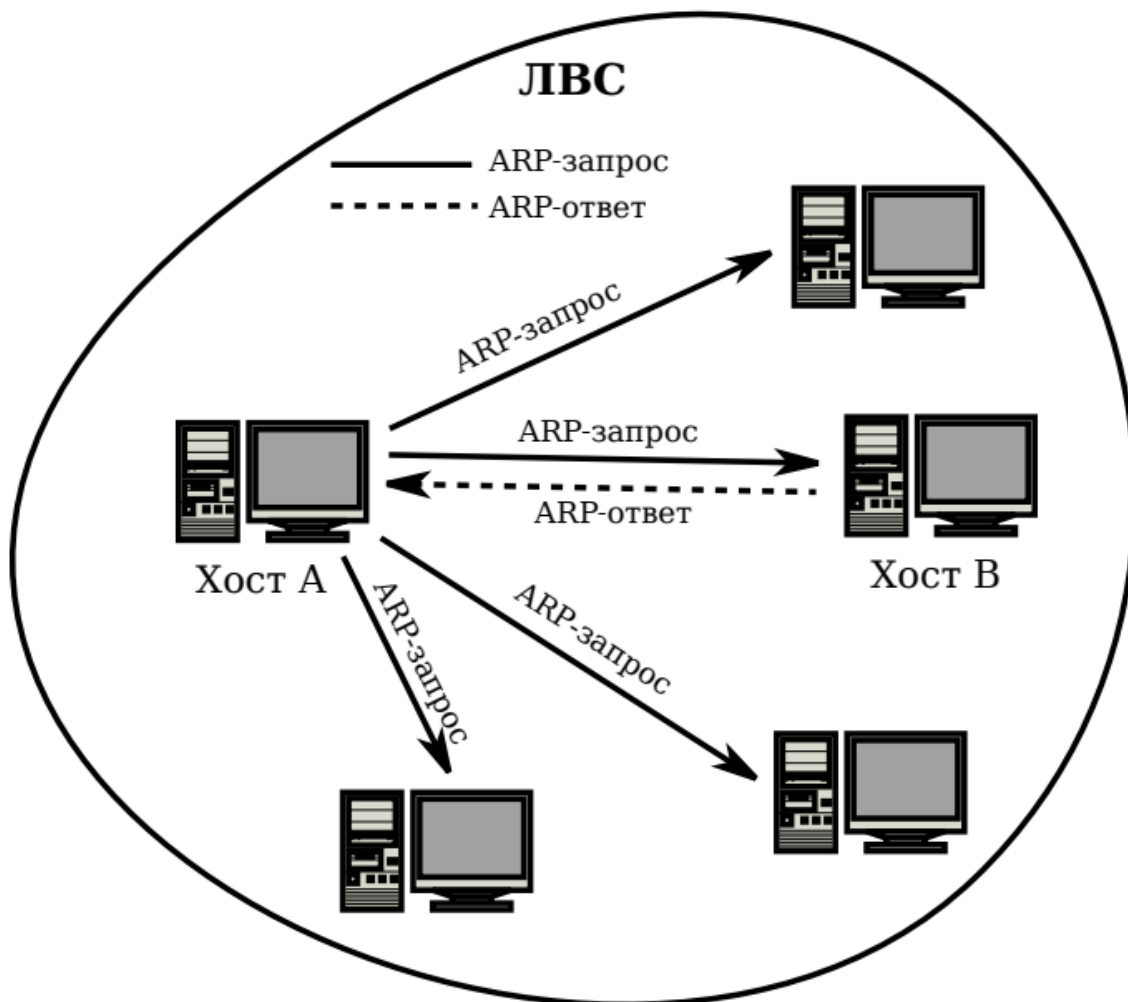


Рис. 3.1. Схема работы протокола ARP

При получении ARP-ответа хост *A* записывает в кэш ARP запись с соответствием IP-адреса хоста *B* и MAC-адреса хоста *B*, полученного из ARP-ответа. Время хранения такой записи ограничено. По истечении времени хранения хост *A* посылает повторный запрос, теперь уже адресно, на известный MAC-адрес хоста *B*. В случае, если ответ не получен, снова посылается широковещательный запрос.

Структура кадра ARP с учетом заголовка Ethernet показана на рис. 3.2.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Destination MAC						Source MAC						ETH TYPE		HTYPE	
PTYPE		HLEN	PLEN	OP CODE		Sender MAC						Sender IP			
Target MAC						Target IP									

Рис. 3.2. Кадр протокола ARP

Значения полей заголовка кадра ARP

Поле	Значение
<b>HTYPE</b>	Номер протокола передачи канального уровня (0x0001 для протокола Ethernet)
<b>PTYPE</b>	Код протокола сетевого уровня (0x0800 для протокола IPv4)
<b>HLEN</b>	Длина физического адреса в байтах. Адреса Ethernet имеют длину 6 байт
<b>PLEN</b>	Длина логического адреса в байтах. IPv4 адреса имеют длину 4 байта
<b>OP CODE</b>	Код операции: 0x01 в случае ARP-запроса и 0x02 в случае ARP-ответа
<b>Sender MAC</b>	Физический адрес отправителя
<b>Sender IP</b>	Сетевой адрес отправителя
<b>Target MAC</b>	Физический адрес получателя. При запросе поле заполняется нулями
<b>Target IP</b>	Сетевой адрес получателя

Самопроизвольный ARP (*gratuitous ARP*) — такое поведение ARP, когда ARP-ответ присылается, когда в этом (с точки зрения получателя) нет особой необходимости. Самопроизвольный ARP-ответ это пакет-ответ ARP, присланный без запроса. Он применяется для определения конфликтов IP-адресов в сети: как только станция получает адрес по DHCP или адрес присваивается вручную, рассылается ARP-ответ *gratuitous ARP*.

Самопроизвольный ARP может быть полезен в следующих случаях:

- обновление ARP-таблиц, в частности, в кластерных системах;
- информирование коммутаторов;
- извещение о включении сетевого интерфейса.

Несмотря на эффективность самопроизвольного ARP, он является особенно небезопасным, поскольку с его помощью можно уверить удаленный узел в том, что MAC-адрес какой-либо системы, находящейся с ней в одной сети, изменился, и указать, какой адрес используется теперь.

### 4.2.3. Сетевая атака ARP-спуфинг

Сетевая атака ARP-спуфинг (*ARP-spoofing*) основана на использовании самопроизвольного ARP.

Чтобы перехватить сетевые пакеты, которые атакуемый хост (А) отправляет на хост В, атакующий хост (С) формирует ARP-ответ, в котором ставит в соответствие IP-адресу хоста В свой MAC-адрес. Далее этот пакет отправляется на хост А. В том случае, если хост А поддерживает самопроизвольный ARP, он модифицирует собственную ARP-таблицу и



помещает туда запись, где вместо настоящего MAC-адреса хоста В стоит MAC-адрес атакующего хоста С.

Теперь пакеты, отправляемые хостом А на хост В, будут передаваться хосту С.

Схема атаки показана на рис. 3.3.

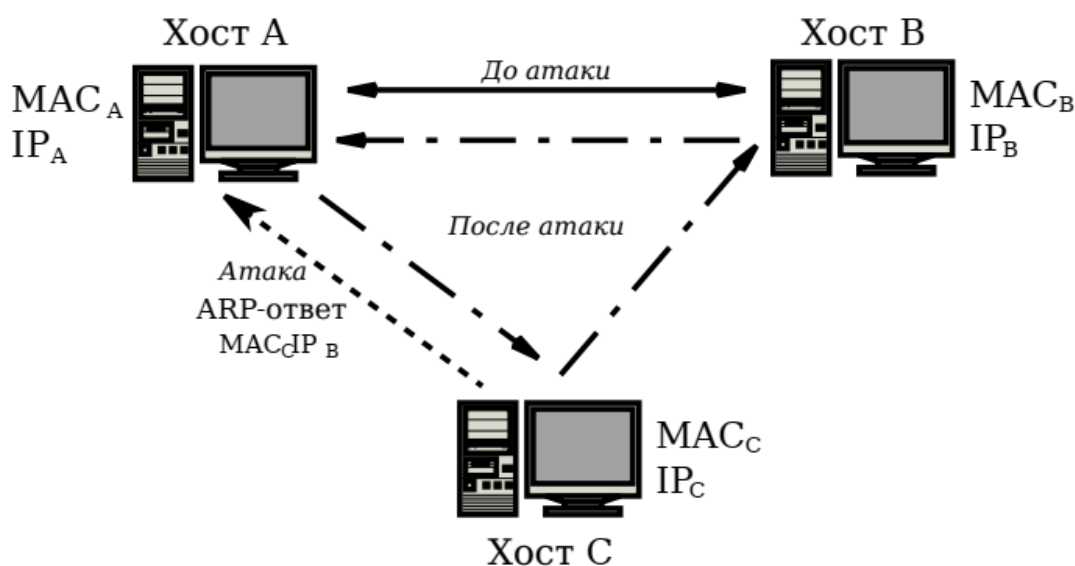


Рис. 3.3. Схема сетевой атаки ARP-спуфинг

#### 4.3. Порядок выполнения лабораторной работы

С помощью инструмента «Вставить текстовую надпись» добавить на рабочее поле эмулятора надпись, содержащую:

1. Номер группы.
2. ФИО студента, выполняющего работу.
3. Номер варианта согласно номеру студента в журнале.

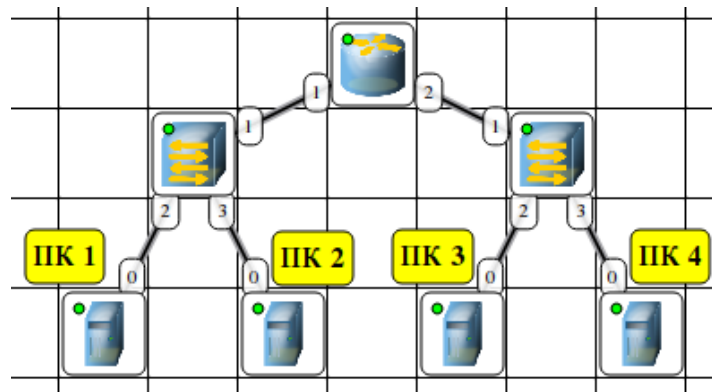
Таблица 3.2

Варианты задания (указаны согласно номеру студента в журнале)

№ Адрес сети/маска	№ Адрес сети/маска	№ Адрес сети/маска
1 10.0.1.0/27	11 10.1.1.64/27	21 10.2.1.128/27
2 10.0.2.32/27	12 10.1.2.96/27	22 10.2.2.160/27
3 10.0.3.64/27	13 10.1.3.128/27	23 10.2.3.192/27
4 10.0.4.96/27	14 10.1.4.160/27	24 10.2.4.224/27
5 10.0.5.128/27	15 10.1.5.192/27	25 10.2.5.0/27
6 10.0.6.160/27	16 10.1.6.224/27	26 10.2.6.32/27
7 10.0.7.192/27	17 10.1.7.0/27	27 10.2.7.64/27
8 10.0.8.224/27	18 10.1.8.32/27	28 10.2.8.96/27
9 10.0.9.0/27	19 10.1.9.64/27	29 10.2.9.128/27
10 10.0.0.32/27	20 10.1.0.96/27	30 10.2.0.160/27

##### 4.3.1. Построение модели сети

1. Выбрать исходные данные для выполнения работы согласно своему варианту. Полученную согласно варианту сеть с маской /27 разбить на две подсети с маской /28 каждая.



(a). в NetEmul

Рис. 3.4. Сеть для изучения протокола ARP

2. Используя соответствующие инструменты на панели эмулятора, построить сеть в соответствии с рис. 3.4. В свойствах маршрутизатора необходимо указать количество интерфейсов, равное 2.

3. Настроить интерфейсы компьютеров и маршрутизаторов, задав каждому IP-адрес и маску подсети (слева — первая подсеть в заданной сети, справа — вторая подсеть). Добавить возле каждого компьютера и интерфейса роутера надписи с их IP-адресом и маской подсети.

4. Настроить на компьютерах маршруты "по-умолчанию" (IP сети = 0.0.0.0; маска подсети = 0.0.0.0). Можно воспользоваться «Таблицей маршрутизации» либо вызвать свойства компьютера двойным щелчком, указать шлюз по умолчанию и включить маршрутизацию.

5. Включить маршрутизацию на маршрутизаторе.

6. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от компьютера в левой подсети до компьютера в правой подсети.

7. Задать каждому компьютеру имя-описание, воспользовавшись пунктом контекстного меню «Задать описание».

#### ***4.3.2. Определение MAC-адреса с помощью ARP-запроса***

1. Запустить для компьютеров 1 и 2 журналы пакетов (пункт меню «Показать журнал»).

2. Очистить ARP-таблицу компьютера 1.

3. Выделить компьютер 1 и с помощью инструмента «Конструктор пакетов» сформировать пакет ARP-запроса для определения MAC-адреса

компьютера 2. Помните, что ARP-запрос рассылается широковещательно (MAC-адрес получателя в заголовке Ethernet — FF:FF:FF:FF:FF:FF), а MAC-адрес искомого узла в заголовке ARP приравнивается к нулевому 00:00:00:00:00:00.

MAC-адрес компьютера 1 указан в окне «Интерфейсы» для компьютера 1.

4. Запустить ARP-запрос, проследить за ним и за сгенерированным для него ARP-ответом по схеме сети и журналам компьютеров 1 и 2.

5. Открыть ARP-таблицу компьютера 1 и убедиться, что запись добавилась в таблицу.

6. Сохранить скриншот экрана (с открытыми журналами) для отчета.

### ***4.3.3. Реализация атаки ARP-спуфинг***

1. Запустить для компьютеров 1 и 2 журналы пакетов (пункт меню «Показать журнал»). При необходимости очистить их.

2. Очистить ARP-таблицу компьютера 1.

3. Выделить компьютер 2 и с помощью инструмента «Конструктор пакетов» сформировать пакет ARP-ответа, в котором будут указаны

- MAC отправителя — MAC компьютера 2;
- IP отправителя — IP интерфейса роутера в левой подсети;
- MAC получателя — MAC компьютера 1;
- IP получателя — IP компьютера 1.

4. Запустить ARP-ответ, проследить за ним. Может возникнуть окно о дублировании IP-адресов в сети — это происходит в том случае, если из-за действий коммутатора пакет-атаку получает и роутер. Окно быстро закрыть.

5. Сразу же запустить передачу пакетов (UDP, 5 KB) от компьютера 1 на компьютер 3. Убедиться, что пакеты вначале приходят на компьютер 2 и лишь потом (если на компьютере 2 включена маршрутизация) отправляются на компьютер 3 (через маршрутизатор).

6. Сохранить скриншот экрана (с открытыми журналами) для отчета.

После выполнения работы продемонстрировать преподавателю работоспособность построенной модели. Проект сохранить для отчета.

### **4.4. Форма представления и содержание отчета**

***Отчет в формате DOC или PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.***

### ***Содержание отчета***

1. Заголовок согласно приложению.
2. Цель работы.
3. Разбиение заданной сети /27 на две подсети /28.
4. Схема модели с указанием IP-адресов устройств и номеров интерфейсов.
5. Скриншоты с результатами разрешения адреса и сетевой атаки.
6. По каждому пункту лабораторной должны быть приведены выводы по работе.

#### **4.5. Контрольные вопросы**

1. Протокол ARP.
2. Формат пакета ARP.
3. Самопроизвольный ARP.
4. IP-адрес.
5. MAC-адрес.
6. ARP-спуфинг.

## **Практическая работа №5. Динамическая маршрутизация по протоколу RIP. Получение сетевых настроек по DHCP**

**5.1 Цель работы:** Ознакомиться с механизмом динамической маршрутизации по протоколу RIP. Научиться настраивать компьютеры и серверы для автоматизации получения компьютерами сетевых настроек.

### 5.2. Теоретический материал

#### **5.2.1. Запуск эмулятора NetEmul**

Для запуска эмулятора NetEmul необходимо либо воспользоваться соответствующим пунктом главного меню операционной системы, либо выполнить в терминале команду netemul.

### 5.3. Порядок выполнения лабораторной работы

С помощью инструмента «Вставить текстовую надпись» добавить на рабочее поле эмулятора надпись, содержащую:

1. Номер группы.
2. ФИО студента, выполняющего работу.
3. Номер варианта согласно номеру студента в журнале.

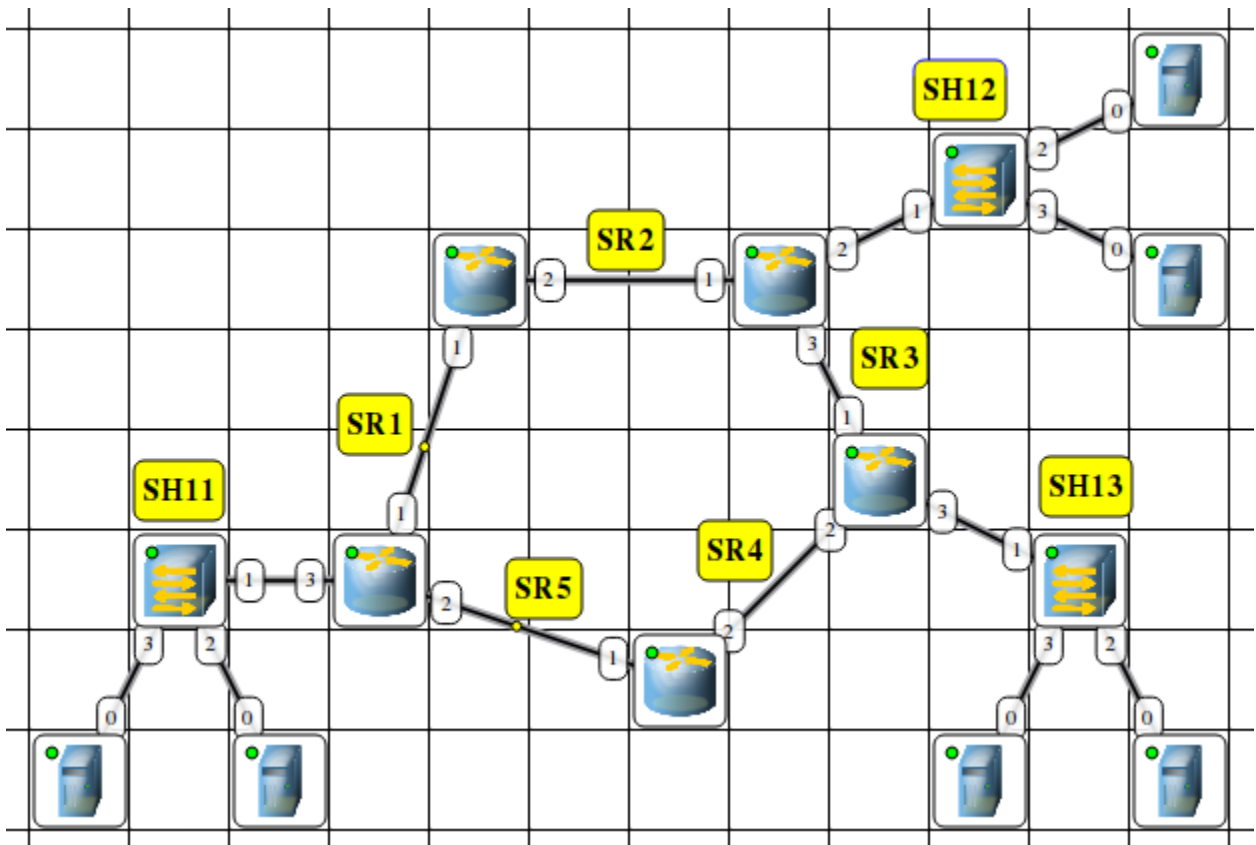
Таблица 4.1

Варианты задания (указаны согласно номеру студента в журнале)

<b>№ Адрес сети/маска</b>	<b>№ Адрес сети/маска</b>	<b>№ Адрес сети/маска</b>
<b>1 10.0.1.0/26</b>	11 10.1.1.128/26	21 10.2.1.0/26
<b>2 10.0.2.64/26</b>	12 10.1.2.192/26	22 10.2.2.64/26
<b>3 10.0.3.128/26</b>	13 10.1.3.0/26	23 10.2.3.128/26
<b>4 10.0.4.192/26</b>	14 10.1.4.64/26	24 10.2.4.192/26
<b>5 10.0.5.0/26</b>	15 10.1.5.128/26	25 10.2.5.0/26
<b>6 10.0.6.64/26</b>	16 10.1.6.192/26	26 10.2.6.64/26
<b>7 10.0.7.128/26</b>	17 10.1.7.0/26	27 10.2.7.128/26
<b>8 10.0.8.192/26</b>	18 10.1.8.64/26	28 10.2.8.192/26
<b>9 10.0.9.0/26</b>	19 10.1.9.128/26	29 10.2.9.0/26
<b>10 10.0.0.64/26</b>	20 10.1.0.192/26	30 10.2.0.64/26

#### **5.3.1. Построение модели сети**

1. Выбрать исходные данные для выполнения работы согласно своему варианту. Полученную согласно варианту сеть с маской /26 разбить на 8 подсетей с маской /29 каждая.



(a). в NetEmul

Рис. 4.1. Модель сети для изучения работы протоколов RIP и DHCP

2. Используя соответствующие инструменты на панели эмулятора, построить сеть в соответствии с рис. 4.1.

3. Распределить полученные ранее адреса сетей между сетями SR1– SR5 и SH11–SH13. Добавить возле каждой сети надпись с ее IP-адресом.

4. Настроить интерфейсы маршрутизаторов, задав каждому IP-адрес и маску подсети в соответствии с выбранным распределением.

### **5.3.2. Настройка динамической маршрутизации по протоколу RIP**

1. На каждом маршрутизаторе добавить и запустить программу RIP.

Пункт контекстного меню «Программы». Кнопка «Добавить». Не забудьте поставить флажок для активации программы.

2. Включить маршрутизацию на маршрутизаторе.

3. Открыть журнал одного из маршрутизаторов. Проследить за перемещением пакетов протокола RIP по сети.

4. Поочередно открыть таблицы маршрутизации каждого маршрутизатора и убедиться, что таблица заполнилась.

### **5.3.3. Настройка автоматического получения сетевых настроек**

*по*

#### **протоколу DHCP**

1. На маршрутизаторах, которые отвечают за сети SH11–SH13 добавить и запустить программу DHCP-сервер. Не забудьте поставить флаг для активации программы.

2. В настройках каждого DHCP-сервера указать интерфейс, «смотрящий» в сторону сети SH, тип адресов — динамические, диапазон адресов, выделяемых для динамической адресации, маску подсети и IP-адрес шлюза.

3. На каждом компьютере добавить и запустить программу DHCP-клиент. Не забудьте поставить флаг для активации программы.

4. В настройках каждого DHCP-клиента укажите интерфейс, который должен автоматически получать сетевые настройки.

5. Открыть диалог настройки интерфейсов каждого компьютера и убедиться, что стоит флаг «Получать настройки автоматически».

6. Дождаться, пока все компьютеры не получат сетевые настройки.

7. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) между компьютерами в разных подсетях.

После выполнения работы продемонстрировать преподавателю работоспособность построенной модели. Проект сохранить для отчета.

#### **5.4. Форма представления и содержание отчета**

Отчет в формате DOC или PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.

#### ***Содержание отчета***

1. Заголовок согласно приложению.
2. Цель работы.
3. Схема модели с указанием IP-адресов устройств и номеров интерфейсов.

4. По каждому пункту лабораторной должны быть приведены выводы по работе.

#### 4.5. Контрольные вопросы

1. Протокол RIP.
2. Протокол DHCP



## **Практическая работа №6. Командная строка управления устройствами CLI. Виртуальные локальные сети VLAN**

**Целью** данной практической работы является ознакомление с командной строкой управления устройствами CLI и локальными портами VLAN

### **Задачи:**

- Ознакомиться с командной строкой CLI и ее командами.
- Во время выполнения заданий отработать навыки работы с CLI (пример 1 и 2)
- Выполнить проектирование различных сетей с помощью CLI (задания 3-6)
- Составить отчет с последовательным выполнением заданий и скриншотами (задания 3-6)

### **Теоретическая часть**

#### **Консоль**

Большинство сетевых устройств компании CISCO допускают *конфигурирование*. Для этого *администратор* сети должен подключиться к устройству через прямое кабельное (консольное) подключение (рис. 5.1).



Итак, *программирование* устройств CISCO чаще всего производят через консольный *порт RJ-45*. На рис. 5.2 и рис. 5.3 приведены фотографии консольных разъемов на маршрутизаторе и 2 варианта консольного кабеля.





**Рис. 5.3.** Варианты консольных кабелей

**Примечание**

Классический консольный кабель имеет разъем DB9 для подключения к COM-порту компьютера и разъем RG-45 для подключения к консольному порту маршрутизатора. Сейчас Cisco активно продвигает новые маршрутизаторы серий 28xx, 38xx и т.д. В них предусмотрена возможность конфигурирования через USB-интерфейс (используются обычные USB-кабели).

Подключив *консоль* и получив *доступ* к устройству через командную строку, *пользователь* (*администратор* сети или *сетевой инженер*) может задавать различные команды и, тем самым, определять параметры конфигурации оборудования.

**Режимы работы с устройством при использовании CLI**

Командная строка представляет собой место, куда пользователь вводит символы, формирующие управляющее воздействие. Работа с командной строкой осуществляется в нескольких режимах ( [таблица 5.1](#)).

Таблица 5.1. Режимы командного интерфейса

Режим	Переход в режим	Вид командной строки	Выход из режима
Пользовательский	Подключение	Router>	logout
Привилегированный	Enable.	Router#	disable
Глобальная конфигурация	Configure terminal	Router(config)#	exit,end или Ctrl-Z
Настройка интерфейсов	Interface	Router(config-if)	exit

Несколько слов о виде командной строки:

**Router>** Приглашение, которое характеризует пользовательский режим, в котором можно просматривать некоторую статистику и проводить самые простые операции вроде пинга. Это режим для сетевого оператора, инженера первой линии техподдержки, чтобы он ничего не повредил и лишнего не узнал. Иными словами, команды в этом режиме позволяют выводить на экран информацию без смены установок сетевого устройства.

**Router#** Приглашение в привилегированном режиме. Привилегированный режим поддерживает команды настройки и тестирования, детальную проверку сетевого устройства, манипуляцию с конфигурационными файлами и доступ в режим конфигурирования. Попасть в него можно, введя команду `enable`.

**Router(config)#**Приглашение в режиме глобальной конфигурации. Он позволяет нам вносить изменения в настройки устройства. Команды режима глобального конфигурирования определяют поведение системы в целом. Активируется командой `#configure terminal` из привилегированного режима.

## Пример 1. Знакомство с командами Cisco IOS

В Cisco *Packet Tracer* интерфейс командной строки для устройств доступен в окне настроек параметров сетевого устройства на вкладке "CLI". Это окно имитирует прямое кабельное (консольное) подключение к сетевому устройству. Работа с командной строкой (CLI) для настройки (программирования) сетевого производится с помощью команд операционной системы Cisco *IOS* (рис. 5.4).

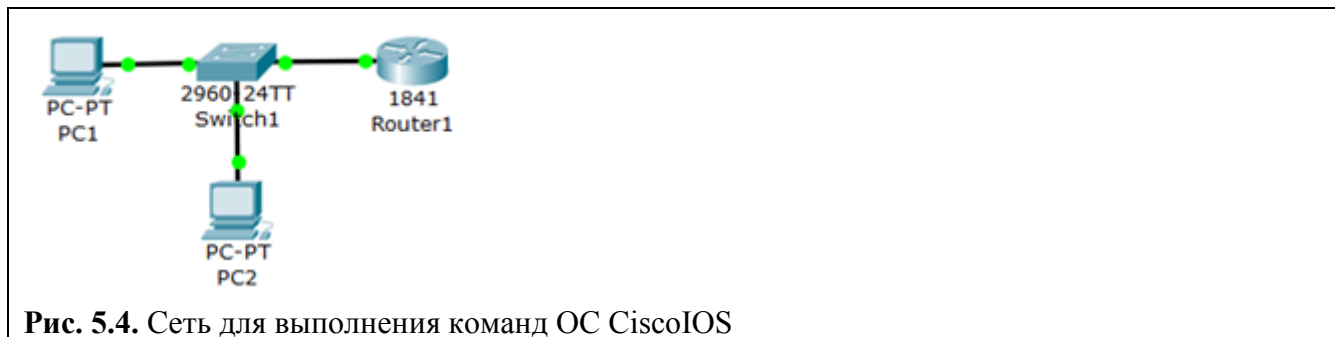


Рис. 5.4. Сеть для выполнения команд ОС CiscoIOS

Выше мы говорили о режимах командного интерфейса – пользовательском, привилегированном и глобального *конфигурация*. Прodelайте все команды входа и выхода в эти режимы для Router1. При входе в сетевое устройство Router1 и нажатии на клавишу *Enter* командная строка имеет вид как на рис. 5.5. *Выход* из пользовательского режима – **logout**.

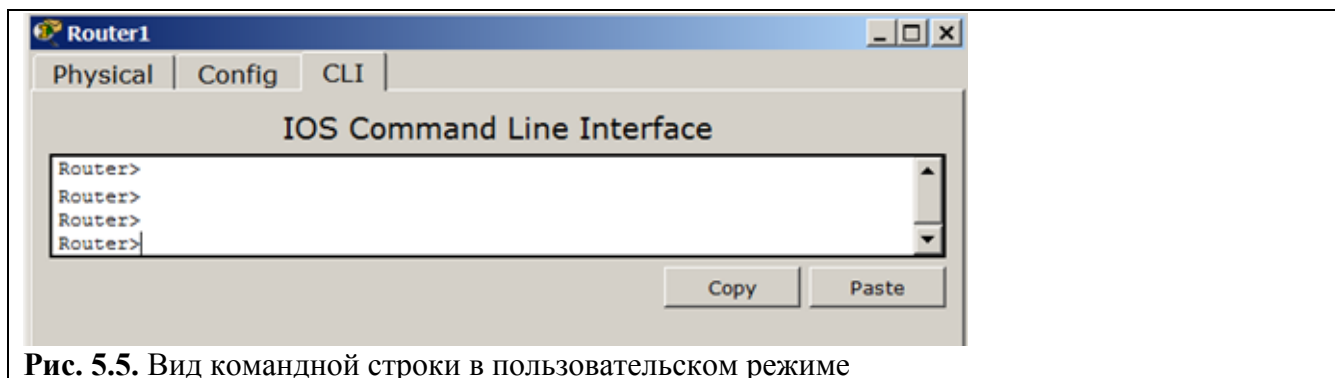


Рис. 5.5. Вид командной строки в пользовательском режиме

Чтобы получить *доступ* к полному набору команд, необходимо сначала активизировать привилегированный режим командой **enable**. О переходе в привилегированный режим будет свидетельствовать появление в командной строке приглашения в виде знака #. *Выход* из привилегированного режима производится командой **disable**.

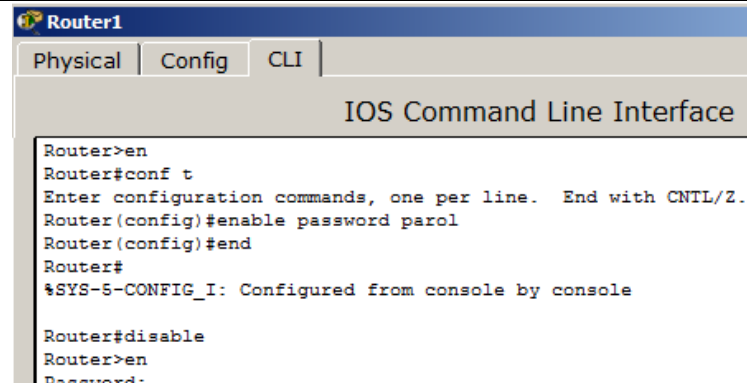
### Примечание

Вместо **enable** можно было набрать **en**. Команды в любом режиме IOS распознаёт по первым уникальным символам.

Режим глобального конфигурирования — реализует мощные однострочные команды, которые решают задачи конфигурирования. Для входа в режим глобального конфигурирования используется команда привилегированного режима **configure terminal**. *Выход* командой **exit** или **end**.

## Установка пароля на вход в привилегированный режим

Пароль доступа позволяет вам контролировать *доступ* в привилегированный режим от неопытных пользователей и злоумышленников. Напомним, что только в привилегированном режиме можно вносить конфигурационные изменения. На Router1 установите *пароль* доступа в этот режим как "parol" командой Router1(config)#enable password parol, затем выйдите из привилегированного режима сетевого устройства, то есть перейдите в пользовательский режим. Попробуйте снова зайти в привилегированный режим. Как видите, без ввода пароля это теперь невозможно ( рис. 5.6).

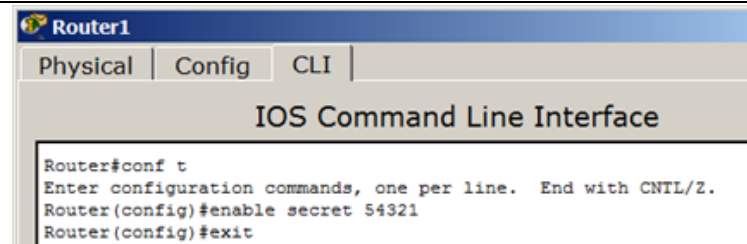


```
Router1
Physical | Config | CLI
IOS Command Line Interface
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#enable password parol
Router(config)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#disable
Router>en
Password:
```

Рис. 5.6. Установка пароля на вход в привилегированный режим


Для изменения пароля введем новый *пароль* привилегированного режима ( рис. 5.7).



```
Router1
Physical | Config | CLI
IOS Command Line Interface
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#enable secret 54321
Router(config)#exit
```

Рис. 5.7. Был пароль 12345, стал пароль 54321

Для сброса пароля можно произвести перезагрузку роутера (рис. 5.8).



```
Router1
Physical | Config | CLI
IOS Command Line Interface
Router>en
Password:
Router#reload
Proceed with reload? [confirm]ySystem Bootstrap, Version 12.3(8r)T8,
RELEASE SOFTWARE (fc1)
Initializing memory for ECC
..
c2811 processor with 524288 Kbytes of main memory
Main memory is configured to 64 bit mode with ECC enabled

Readonly ROMMON initialized

Self decompressing the image :
#####
## [OK]
Continue with configuration dialog? [yes/no]: n

Press RETURN to get started!

Router>en
Router#
```

Рис. 5.8. Перезагрузка R1 командой reload

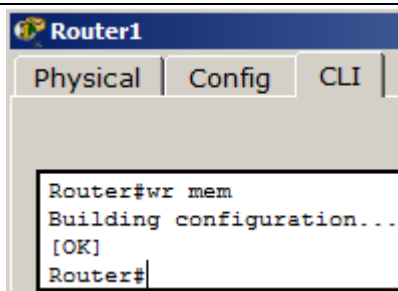
## Советы при работе с CLI

Все команды в консоли можно сокращать, но, важно, чтобы сокращение однозначно указывало на команду. Используйте клавишу **Tab** и знак вопроса (?). По нажатию Tab сокращенная команда дописывается до полной, а знак вопроса (?), следующий за командой, выводит список дальнейших возможностей и небольшую справку по ним. Можно перейти к следующей команде, сохранённой в буфере. Для этого нажмите на Стрелку вниз или **Ctrl + N**. Можно вернуться к командам, введённым ранее. Нажмите на Стрелку вверх или **Ctrl + P** (рис. 5.9).



**Рис. 5.9.** Стрелки Вверх или Вниз на клавиатуре позволяют листать ранее использованные вами команды

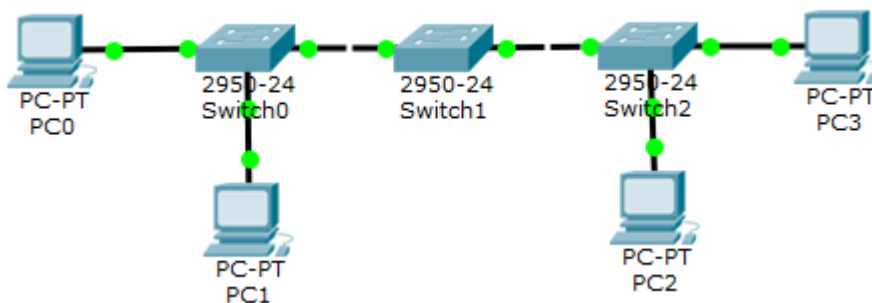
Активная *конфигурация* автоматически не сохраняется и будет потеряна в случае сбоя электропитания. Чтобы сохранить настройки роутера используйте команду **write memory** (рис. 5.10).



**Рис. 5.10.** Сохранение текущей конфигурации R1

## Пример 2.

Схема сети показана на рис. 5.11.



**Рис. 5.11.** Схема сети

Нужно:

1. Построить такую сеть
2. Изменить имя коммутаторов Cisco;

В режиме **глобальной конфигурации** изменяем имя коммутатора  
**Switch(config)#hostname Switch1**

3. Обеспечить парольный доступ к привилегированному режиму на коммутаторах;

В режиме **глобальной конфигурации** изменяем пароль  
**Switch1(config)#enable password parol**

4. Задать ip-адреса и маски коммутаторам (172.16.1.11/24, 172.16.1.12/24, 172.16.1.13/24);

Заходим в настройку интерфейса vlan1 для коммутатора Switch1  
**Switch1(config)#interface vlan1**

Устанавливаем  
**Switch1(config-if)#ip address 172.16.1.11 255.255.255.0**

Включаем vlan1  
**Switch1(config-if)#no shutdown**

5. Задать ip-адреса и маски сетей персональным компьютерам. (172.16.1.1/24, 172.16.1.2/24, 172.16.1.3/24, 172.16.1.4/24);

6. Убедиться в достижимости всех объектов сети по протоколу IP;

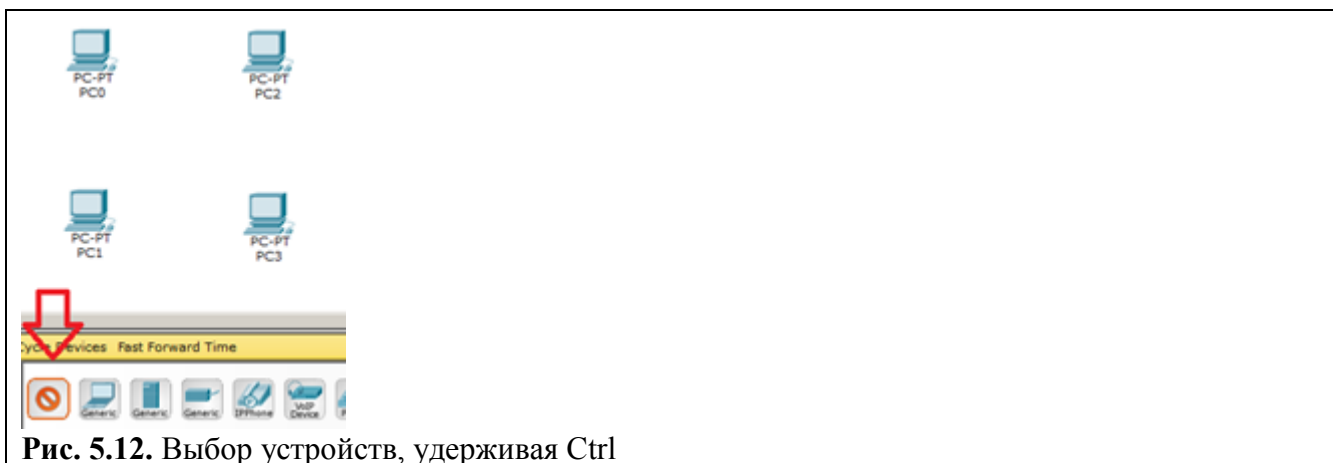
7. Переключившись в "Режим симуляции" и рассмотреть и пояснить процесс обмена данными по протоколу ICMP между устройствами (выполнив команду Ping с одного компьютера на другой).

### Задание 3. VLAN с одним коммутатором

#### Виртуальные локальные сети VLAN

*VLAN (Virtual Local Area Network)* — виртуальная локальная *компьютерная сеть* из группы хостов с общим набором требований. *VLAN* позволяют хостам группироваться или дистанцироваться между собой. Устройства, в пределах одной *VLAN* могут общаться, а узлы, находящиеся в разных *VLAN*'ах, невидимы друг для друга.

Для рисования ПК выбираем в конечных устройства настольный компьютер и, удерживая **Ctrl**, (так быстрее) нажмите 1 раз на ПК а затем рисуйте нужное кол-во ПК, щелкая мышкой ( рис. 5.12). Этим приемом вы сможете за один раз нарисовать сразу 4 ПК.



Устанавливаем коммутатор и, удерживая Ctrl, создаем подключение прямым кабелем, выбирая порты коммутатора. После инициализации портов все лампы загорятся зеленым. На схему будет две подсети (рис. 5.13).



### Примечание

Имя VLAN1 используется по умолчанию, его лучше в нашем примере не использовать.

На коммутаторе набираем команду **en** и входим в привилегированный режим. Затем набираем команду **conf t** для входа в режим глобального конфигурирования. Если подвести курсор мыши к портам коммутатора, то вы увидите какие порты в каком сегменте задействованы. Для VLAN3 – это Fa0/3 и Fa0/4 (предположим, что это будет бухгалтерия - buh) и для VLAN2 – это Fa0/1 и Fa0/2 (предположим, что это будет склад – sklad). Сначала будем конфигурировать второй сегмент сети VLAN2 (sklad) – рис. 5.14.

```
Switch#
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 2
Switch(config-vlan)#name sklad
```

**Рис. 5.14.** VLAN2 получает имя sklad

В виртуальной сети VLAN2 настраиваем порты коммутатора Fa0/1 и Fa0/2 как access порты, т.е. порты для подключения пользователей (рис. 5.15).



```

Switch0
-----
Physical | Config | CLI |
-----
IOS Command Line Interface

Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
Switch(config-if)#exit
Switch(config)#int fa0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
Switch(config-if)#end
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#

```

Рис. 5.15. Указываем порты коммутатора для подключения пользователей

Теперь командой **show vlan** можно проверить результат ( рис. 5.16).

```

Switch0
-----
Physical | Config | CLI |
-----
IOS Command Line Interface

Switch#show vlan

VLAN Name                Status    Ports
-----
1    default                 active    Fa0/3, Fa0/4, Fa0/5, Fa0/6
                                           Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                           Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                           Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                           Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                           Fa0/23, Fa0/24, Gig0/1, Gig0/2
2    sklad                   active    Fa0/1, Fa0/2
1002 fddi-default          act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default      act/unsup
1005 trnet-default        act/unsup

```

Рис. 5.16. Подсеть VLAN2 склад настроена

Далее работаем с VLAN3 (рис. 5.17).

```

Switch#
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config-vlan)#vlan 3
Switch(config-vlan)#name buh
Switch(config-vlan)#exit
Switch(config)#

```

Рис. 5.17. VLAN3 получает имя buh

В виртуальной сети VLAN3 настраиваем порты коммутатора Fa0/3 и Fa0/4 как **access** порты, т.е. порты для подключения пользователей, затем командой **show vlan** можно проверить и убедиться, что мы создали в сети 2 сегмента на разные порты коммутатора (рис. 5.18).



```

Switch0
Physical | Config | CLI
IOS Command Line Interface

Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fa0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#int fa0/4
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#end
Switch#
%SYS-5-CONFIG_I: Configured from console by console

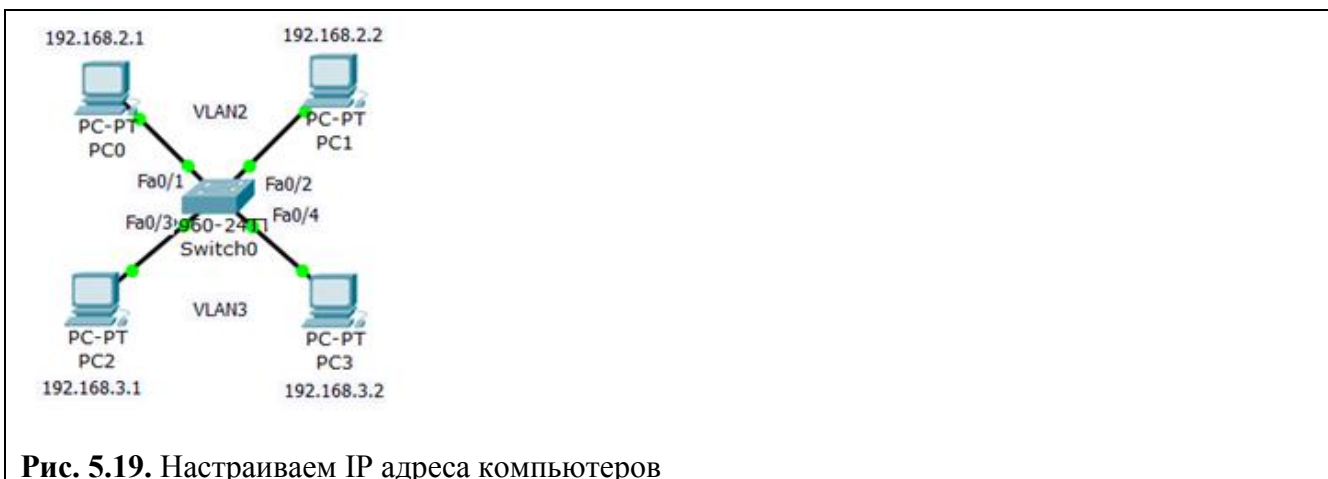
Switch#sh vlan

VLAN Name                Status    Ports
-----
1  default                 active    Fa0/5, Fa0/6, Fa0/7, Fa0/8
                               Fa0/9, Fa0/10, Fa0/11, Fa0/12
                               Fa0/13, Fa0/14, Fa0/15, Fa0/16
                               Fa0/17, Fa0/18, Fa0/19, Fa0/20
                               Fa0/21, Fa0/22, Fa0/23, Fa0/24
                               Gig0/1, Gig0/2
2  sklud                   active    Fa0/1, Fa0/2
3  buh                     active    Fa0/3, Fa0/4
1002 fddi-default          act/unsup
1003 token-ring-default    act/unsup
1004 fddinet-default       act/unsup
1005 trnet-default         act/unsup

```

**Рис. 5.18.** Мы настроили VLAN2 и VLAN3

Настраиваем IP адреса компьютеров – для VLAN2 из сети 192.168.2.0, а для VLAN3 из сети 192.168.3.0 (рис. 5.19).



**Рис. 5.19.** Настраиваем IP адреса компьютеров

Проверяем связь ПК в пределах VLAN и отсутствие связи между VLAN2 и VLAN3 (рис. 5.20).

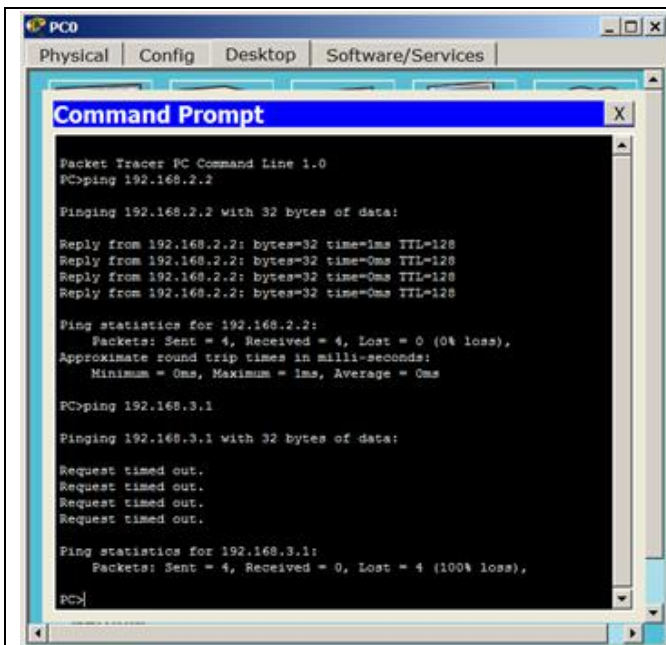


Рис. 5.20. Все работает так, как было задумано

Итак, на компьютере ПК0 мы убедились, что компьютер в своем сегменте видит ПК, а в другом сегменте – нет.

#### Задание 4. Настройка виртуальной сети на коммутаторе 2960

Итак, на компьютере ПК0 мы убедились, что компьютер в своем сегменте видит ПК, а в другом сегменте – нет.

Создайте *сеть*, топология которой представлена на рис. 5.21.

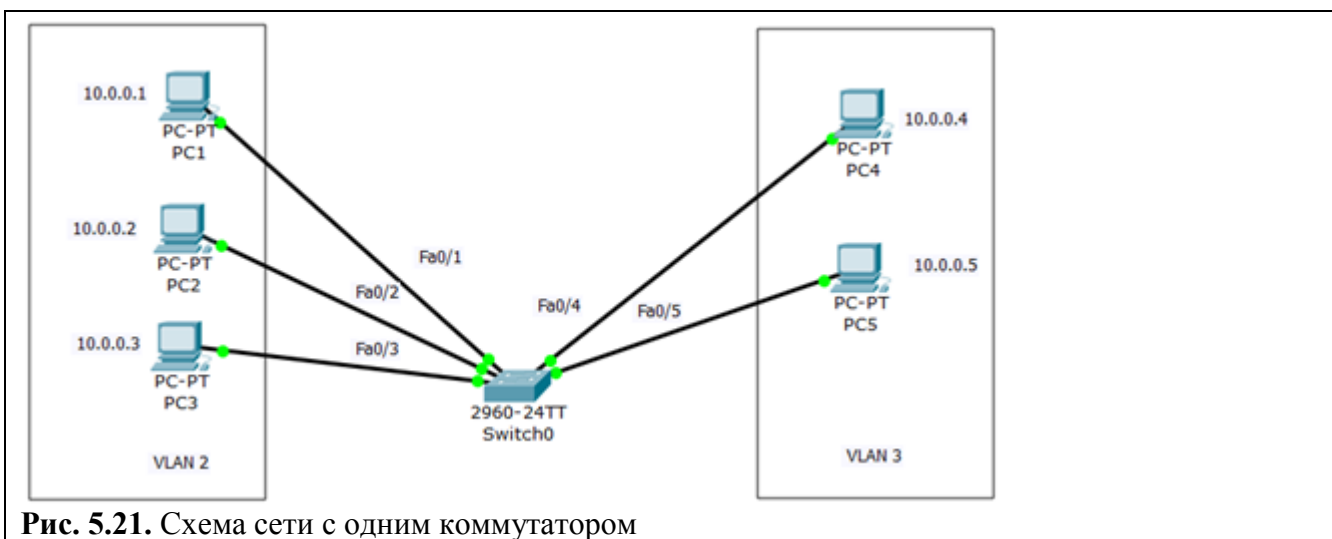


Рис. 5.21. Схема сети с одним коммутатором

Задача данной работы является создание 2х независимых групп компьютеров: ПК1-ПК3 должны быть доступны только друг для друга, а вторая независимая *группа* - компьютеры ПК4 и ПК5.

#### Настройка коммутатора

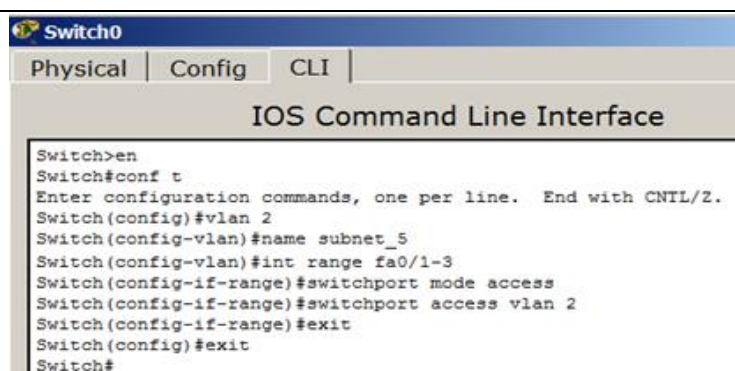
Первоначально сформируем VLAN2. Дважды щелкните левой кнопкой мыши по коммутатору. В открывшемся окне перейдите на вкладку **CLI**. Вы увидите окно консоли. Нажмите на клавишу **Enter** для того, чтобы приступить к вводу команд. Перейдем в привилегированный режим, выполнив команду **enable**:

## Switch>en

По умолчанию все ПК объединены в VLAN1. Для реализации сети, которую мы запланировали, создадим на коммутаторе еще два VLAN (2 и 3). Для этого в привилегированном режиме выполните следующую команду для перехода в режим конфигурации:

## Switch#conf t

Теперь вводим команду **VLAN 2**. Данной командой вы создадите на коммутаторе VLAN с номером 2. Указатель ввода Switch (config)# изменится на Switch (config-vlan)# это свидетельствует о том, что вы конфигурируете уже не весь коммутатор в целом, а только отдельный VLAN, в данном случае VLAN номер 2 (рис. 5.22).



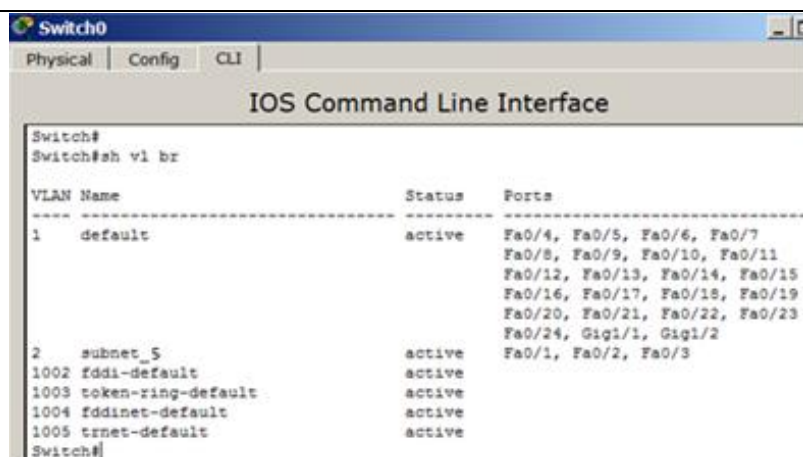
```
Switch0
Physical | Config | CLI |
IOS Command Line Interface
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 2
Switch(config-vlan)#name subnet_5
Switch(config-vlan)#int range fa0/1-3
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 2
Switch(config-if-range)#exit
Switch(config)#exit
Switch#
```

Рис. 5.22. Листинг команд для формирования VLAN2

## Примечание

Командой **VLAN2**, мы создаем на коммутаторе новый VLAN с номером 2. Команда **name subnet\_5** присваивает имя subnet\_5 виртуальной сети номер 2. Выполняя команду **interface range fast Ethernet 0/1-3** мы переходим к конфигурированию интерфейсов fastEthernet 0/1, fastEthernet 0/2 и fastEthernet 0/3 коммутатора. Слово **range** в данной команде, указывает на то, что мы будем конфигурировать не один порт, а диапазон портов. Команда **switchport mode access** конфигурирует выбранный порт коммутатора, как порт доступа (access порт). Команда **switchport access vlan 2** указывает, что данный порт является портом доступа для VLAN номер 2.

Выйдите из режима конфигурирования, дважды набрав команду **exit** и просмотрите результат конфигурирования (рис. 5.23), выполнив команду **sh vl br**. Как видим, на коммутаторе появился VLAN с номером 2 и именем subnet\_5, портами доступа которого являются fastEthernet 0/1, fastEthernet 0/2 и fastEthernet 0/3.



```
Switch0
Physical | Config | CLI |
IOS Command Line Interface
Switch#
Switch#sh vl br
VLAN Name                Status    Ports
-----
1  default                 active    Fa0/4, Fa0/5, Fa0/6, Fa0/7
                                   Fa0/8, Fa0/9, Fa0/10, Fa0/11
                                   Fa0/12, Fa0/13, Fa0/14, Fa0/15
                                   Fa0/16, Fa0/17, Fa0/18, Fa0/19
                                   Fa0/20, Fa0/21, Fa0/22, Fa0/23
                                   Fa0/24, Gig1/1, Gig1/2
2  subnet_5                active    Fa0/1, Fa0/2, Fa0/3
1002 fddi-default         active
1003 token-ring-default   active
1004 fddinet-default      active
1005 trnet-default        active
Switch#
```

Рис. 5.23. Просмотр информации о VLAN на коммутаторе

Команда **sh vl br** выводит информацию о существующих на коммутаторе VLAN-ах. В результате выполнения команды на экране появится: **номера VLAN** (первый столбец), **название VLAN** (второй столбец), **состояние VLAN** (работает он или нет) – третий столбец, **порты**, принадлежащие к данному VLAN (четвертый столбец).

Далее аналогичным образом создадим **VLAN 3** с именем **subnet\_6** и сделаем его портами доступа интерфейсы fastEthernet 0/4 и fastEthernet 0/5. Результат показан на рис. 5.24.

```
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 3
Switch(config-vlan)#name subnet_6
Switch(config-vlan)#int range fa0/4-5
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 3
Switch(config-if-range)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#sh vl br
VLAN Name                Status    Ports
-----
1    default                active    Fa0/6, Fa0/7, Fa0/8,
Fa0/9
Fa0/10, Fa0/11, Fa0/12,
Fa0/13
Fa0/14, Fa0/15, Fa0/16,
Fa0/17
Fa0/18, Fa0/19, Fa0/20,
Fa0/21
Fa0/22, Fa0/23, Fa0/24,
Gig0/1
Gig0/2
2    subnet_5                active    Fa0/1, Fa0/2, Fa0/3
3    subnet_6                active    Fa0/4, Fa0/5
1002 fddi-default           active
1003 token-ring-default   active
1004 fddinet-default      active
1005 trnet-default        active
Switch#
```

Рис. 5.24. Результат – настройка на коммутаторе VLAN2 и VLAN3

### Проверка результатов работы

Сеть настроена и нужно ее протестировать. Результат положителен, если в пределах своей VLAN компьютеры доступны, а компьютеры из разных VLAN не доступны ( рис. 5.25). У нас все пять компьютеров находя в одной сети 10.0.0/8, но они находятся в разных виртуальных локальных сетях.

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Рис. 5.25. Пинг с PC1 на PC3 и PC4

## Задание 5. VLAN с двумя коммутаторами. Разделяемый общий канал (транк)

На практике часто возникает задача разделения устройств, подключенных к одному или нескольким коммутаторам на несколько непересекающихся локальных сетей. В случае, если используется только один *коммутатор*, то эта задача решается путем конфигурирования портов коммутатора, указав каждому порту к какой локальной сети он относится. Если же используется несколько коммутаторов (рис. 5.26), то необходимо между коммутаторами помимо данных передавать информацию к какой локальной сети относится *кадр*. Для этого был разработан стандарт 802.1Q.



Рис. 5.26. Виртуальные локальные сети (VLAN) с использованием двух коммутаторов

От теории перейдем к практике и произведем дублирование нашей сети (той, которая была показана ранее на рис. 5.21). Для этого выделим всю *сеть* инструментом **Select** (Выделить), и, удерживая клавишу **Ctrl**, перетащим на новое место в рабочей области программы. Так мы произведем *копирование* (рис. 5.27).

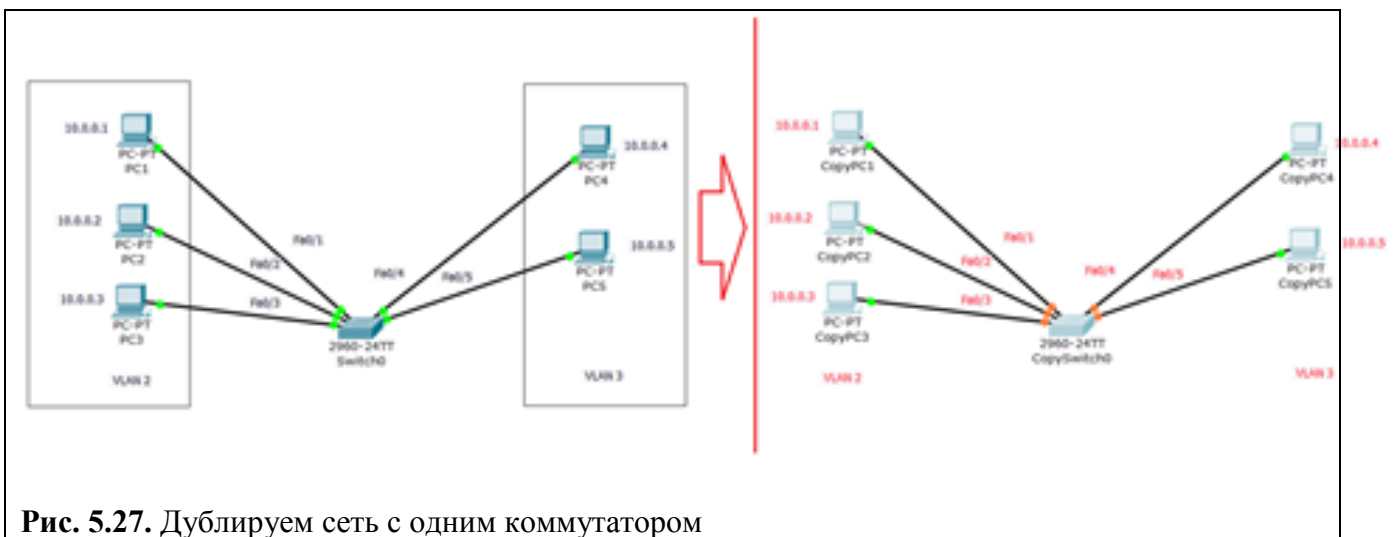
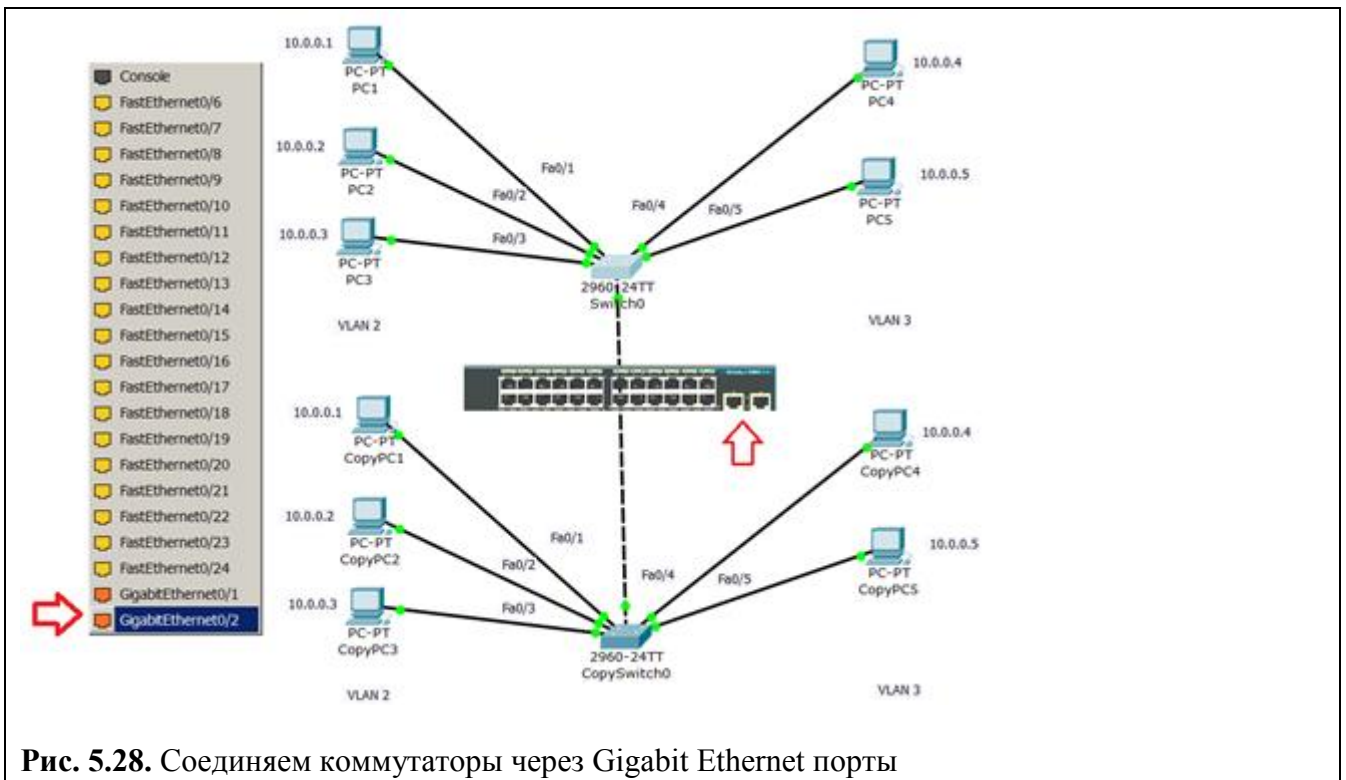


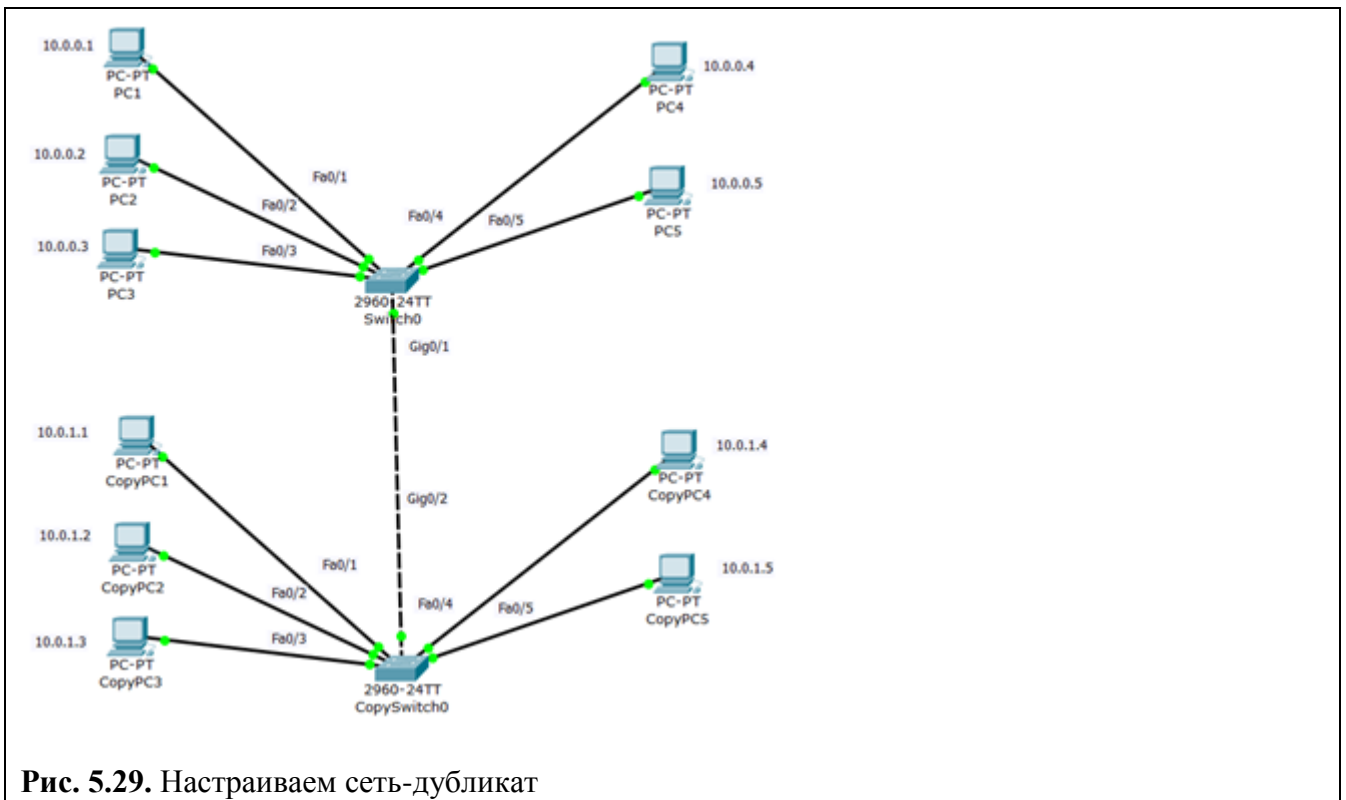
Рис. 5.27. Дублируем сеть с одним коммутатором

Соединим коммутаторы перекрестным кабелем (кроссом) через самые производительные порты – *Gigabit Ethernet* (рис. 5.28).





Теперь поправим настройки на дубликате исходной сети (рис. 5.29).



Укажем новый вариант подсетей VLAN2 и VLAN3, а также выделим **trunk** (транк) связь коммутаторов (рис. 5.30).

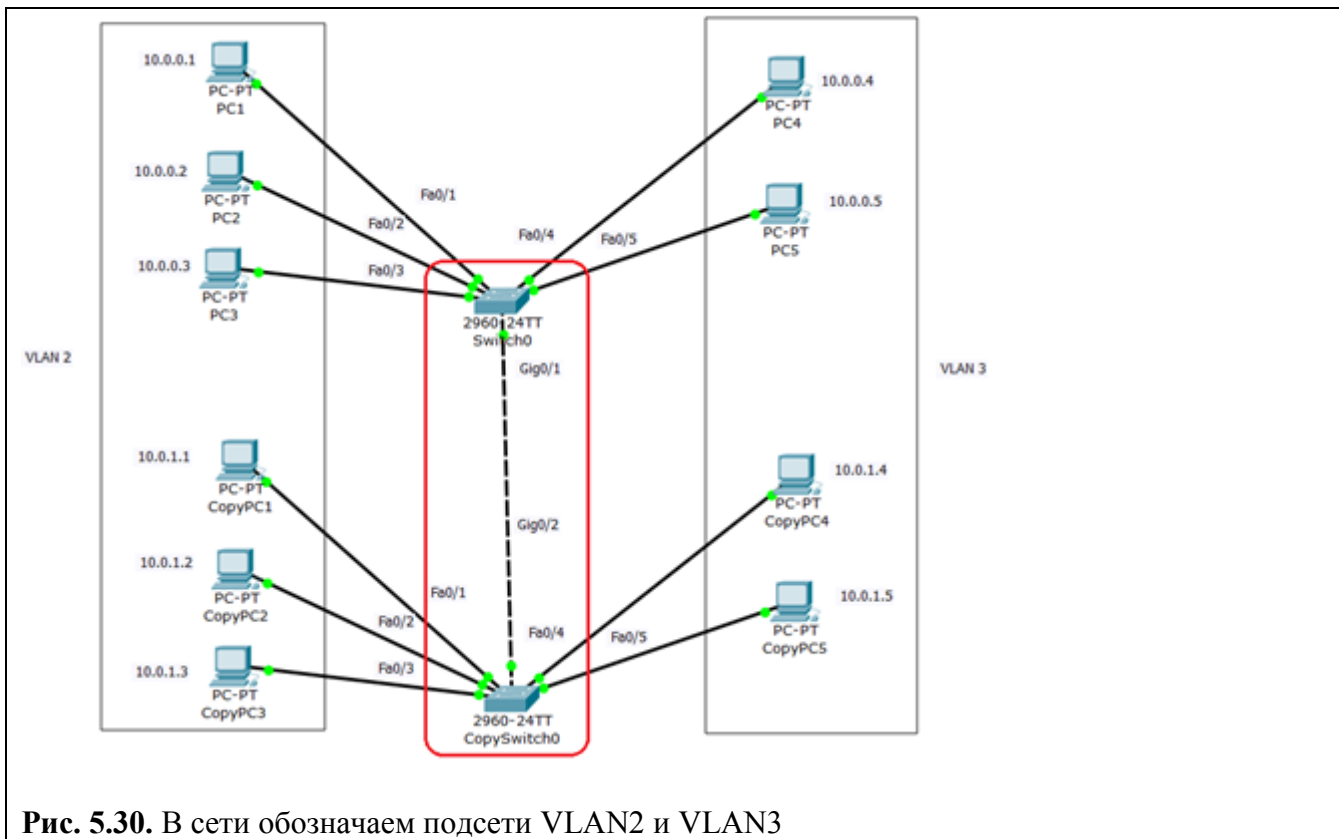


Рис. 5.30. В сети обозначаем подсети VLAN2 и VLAN3

### Настраиваем транк порт Gig0/1

При настройке Gig0/1 на коммутаторе Switch0 мы меняем состояние порта и указываем vlan 2 и 3 для работы с ним (рис. 5.31).

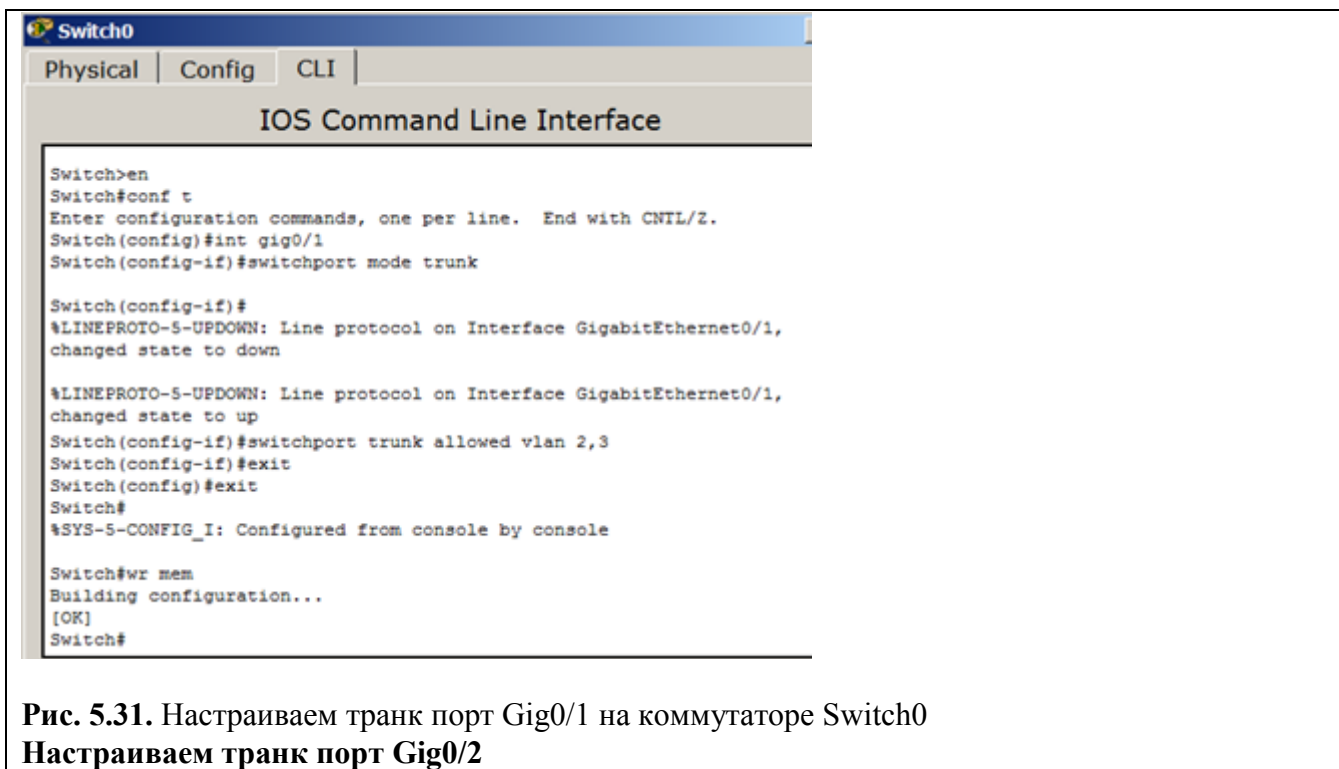


Рис. 5.31. Настраиваем транк порт Gig0/1 на коммутаторе Switch0  
**Настраиваем транк порт Gig0/2**

Транк порт Gig0/2 на коммутаторе CopySwitch0 настраиваем аналогично (рис. 5.32).

```

Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int gi0/2
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allow vlan 2,3
Switch(config-if)#end
Switch#
%SYS-5-CONFIG_I: Configured from console by console

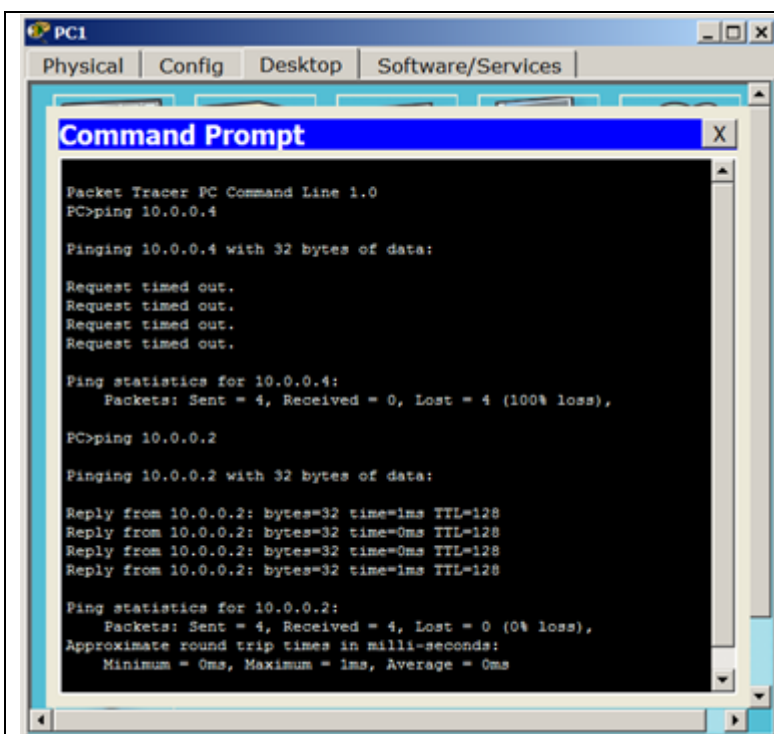
Switch#end

```

**Рис. 5.32.** Настраиваем trunk порт Gig0/2 на коммутаторе CopySwitch0

### Диагностика результатов работы

Проверяем пинг с PC1 в разные vlan (рис. 5.33). Все отлично: в пределах своей vlan ПК доступны, а между ПК разных vlan связи нет.



**Рис. 5.33.** Пинг с PC1 в разные vlan

### Задание 6. Настройка виртуальной сети из двух коммутаторов и четырех ПК.

Ниже мы рассмотрим, как настроить *VLAN* из двух свитчей и четырех ПК.

Создайте *сеть*, топология которой представлена на рис. 5.34. Пока в сети 10.0.0.0 нет разделения на *VLAN* - все компьютеры доступны между собой.



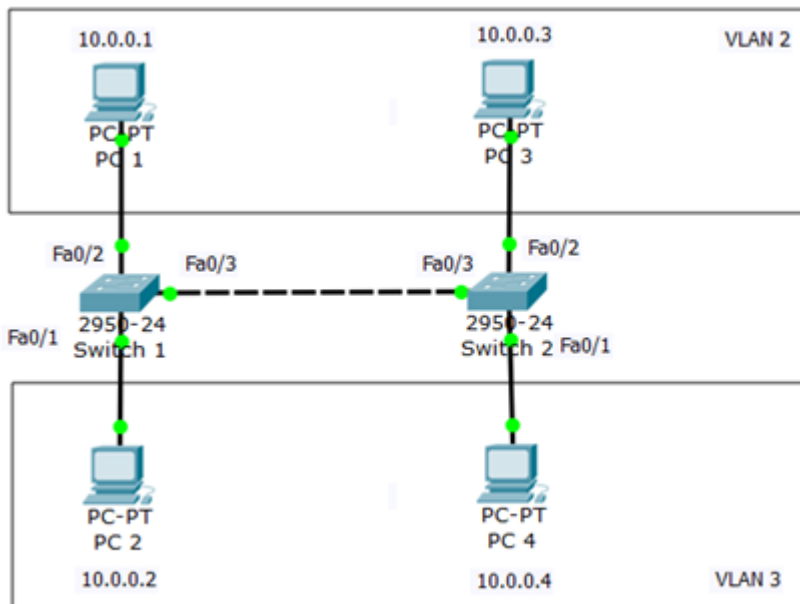


Рис. 5.34. Схема сети

Итак, подсети *Vlan 2* принадлежат порты коммутаторов Fa0/2, а *Vlan 3* принадлежат порты коммутаторов Fa0/1.

### Настройка VLAN 2 и VLAN3

Перейдите к настройке коммутатора Switch1. Откройте его консоль. В открывшемся окне перейдите на вкладку CLI, войдите в привилегированный режим и настройте VLAN 2 и VLAN3. Затем просмотрите информацию о существующих на коммутаторе VLAN-ах командой: Switch1#sh vl br (рис. 5.35).

```

Switch 1
Physical | Config | CLI
IOS Command Line Interface

Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 2
Switch(config-vlan)#int fa0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
Switch(config-if)#exit
Switch(config)#vlan 3
Switch(config-vlan)#int fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

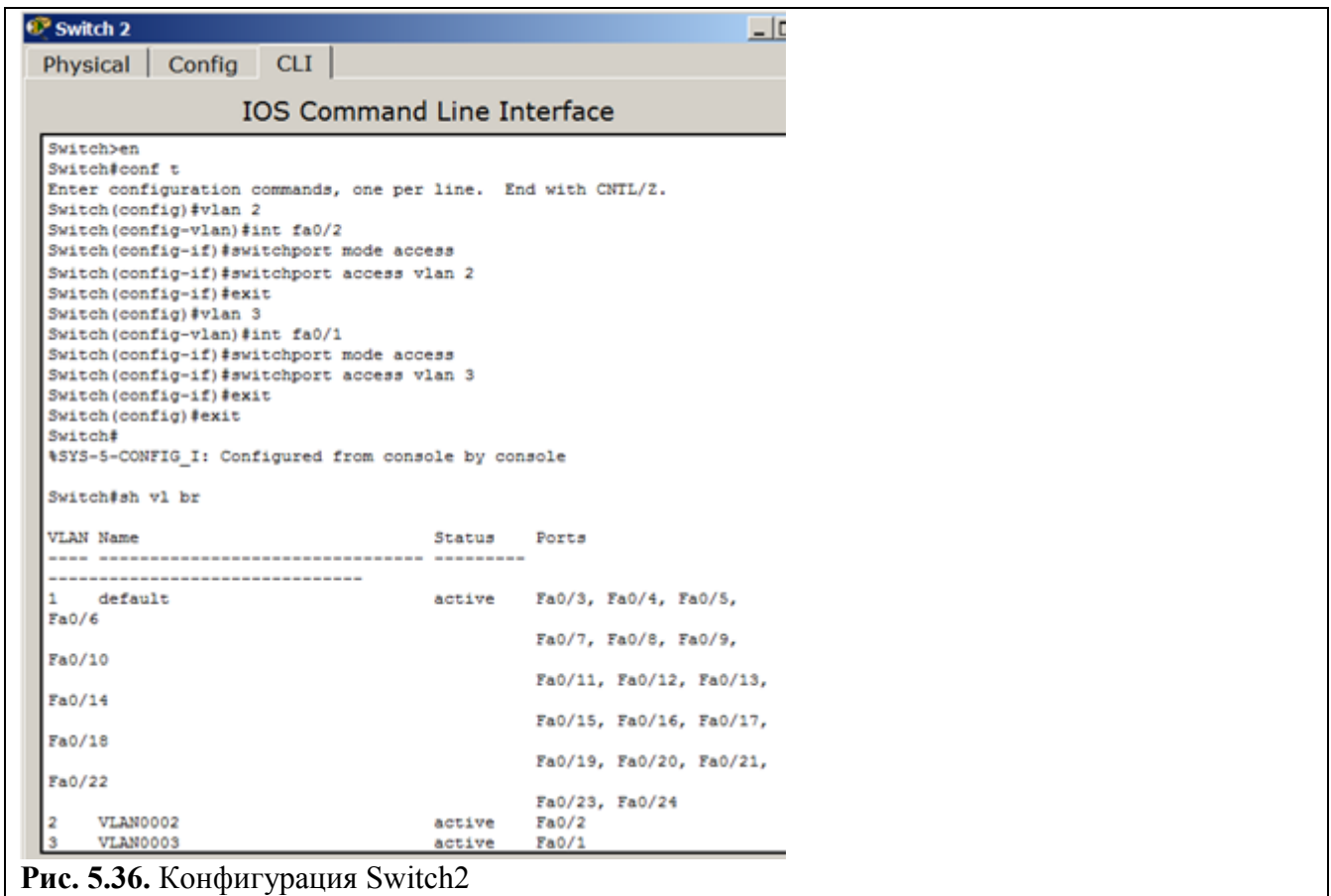
Switch#sh vl br

VLAN Name                Status    Ports
-----
1  default                active    Fa0/3, Fa0/4, Fa0/5,
Fa0/6
Fa0/7, Fa0/8, Fa0/9,
Fa0/10
Fa0/11, Fa0/12, Fa0/13,
Fa0/14
Fa0/15, Fa0/16, Fa0/17,
Fa0/18
Fa0/19, Fa0/20, Fa0/21,
Fa0/22
2  VLAN0002              active    Fa0/23, Fa0/24
Fa0/2
3  VLAN0003              active    Fa0/1

```

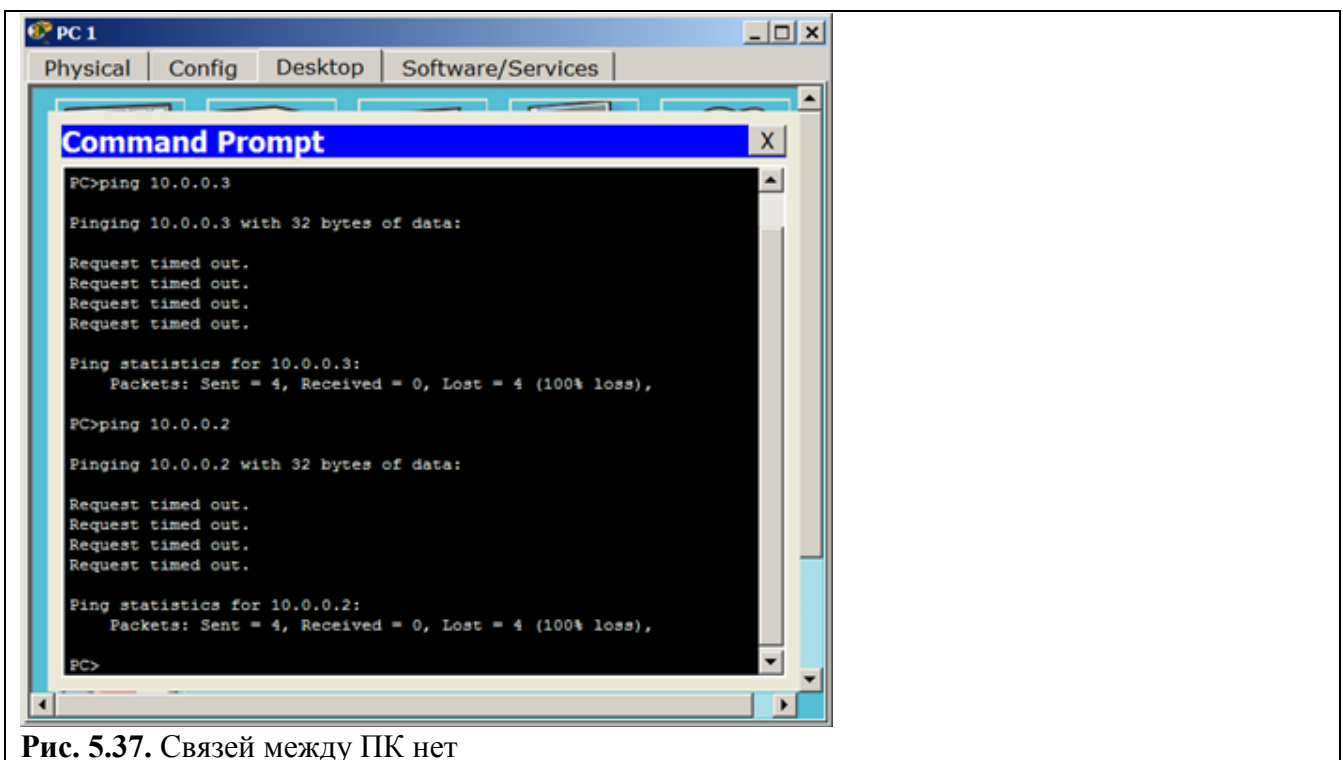
Рис. 5.35. Конфигурация Switch1

Аналогичным образом сконфигурируйте Switch2, исходя из того, что по условиям задачи у нас Fa0/2 расположен в Vlan2, а Fa0/1 находится в Vlan 3 (это не всегда так). Результат конфигурирования S2 показан на [рис. 5.36](#).



**Рис. 5.36.** Конфигурация Switch2

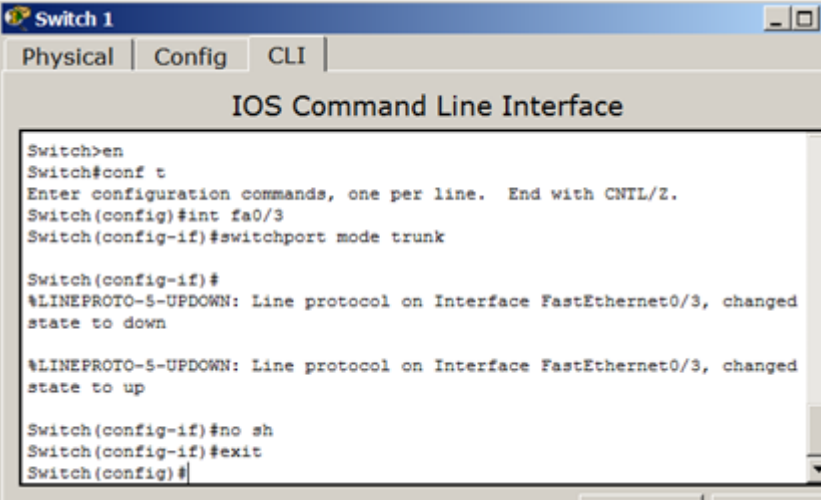
Итак, подсети Vlan 2 принадлежат порты коммутаторов Fa0/2, а Vlan 3 принадлежат порты коммутаторов Fa0/1. Поскольку в данный момент нет обмена информацией о vlan'ах, то все компьютеры разобщены ([рис. 5.37](#)).



**Рис. 5.37.** Связей между ПК нет

## Настройка связи коммутаторов через транковый порт

Теперь организуем магистраль обмена между коммутаторами. Для этого настроим третий порт Fa0/3 на каждом коммутаторе как транковый. Войдите в консоль коммутатора **Switch1** и задайте транковый порт (рис. 5.38).



```
Switch 1
Physical | Config | CLI |
IOS Command Line Interface
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fa0/3
Switch(config-if)#switchport mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed
state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed
state to up

Switch(config-if)#no sh
Switch(config-if)#exit
Switch(config)#
```

Рис. 5.38. Настраиваем транковый порт на S1

Откройте конфигурацию коммутатора S1 на интерфейсе FastEthernet 0/3 и убедитесь, что порт транковый (рис. 5.39).

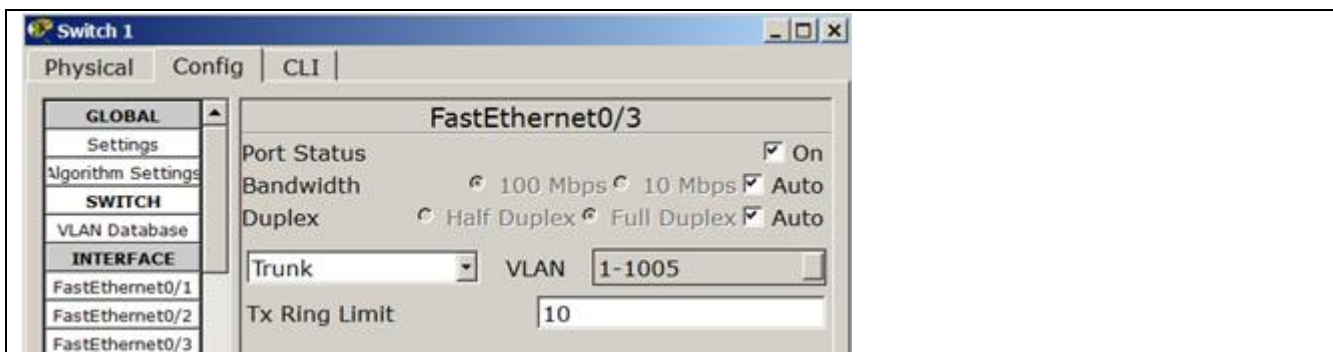


Рис. 5.39. Конфигурация интерфейса FastEthernet0/3 на Switch1

На коммутаторе Switch2 интерфейс FastEthernet 0/3 автоматически настроится как транковый (рис. 5.40).

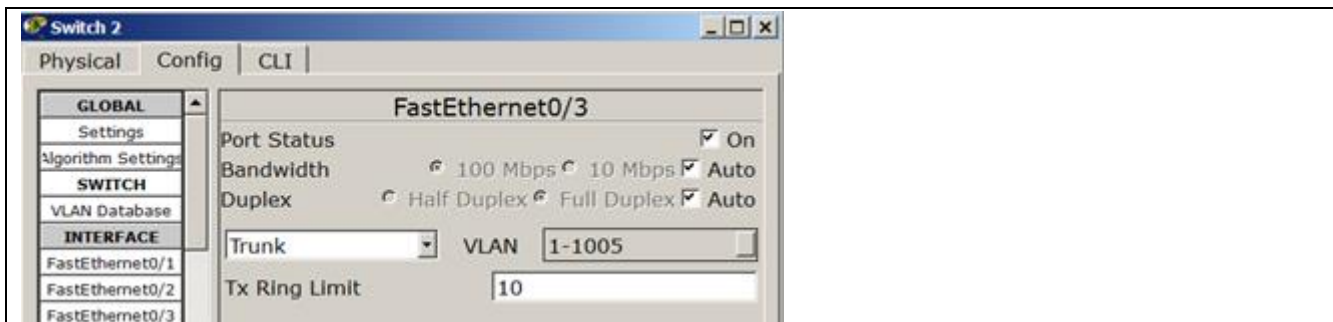


Рис. 5.40. Конфигурация интерфейса FastEthernet0/3 на Switch2

Теперь компьютеры, входящие в один vlan должны пинговаться, а компьютеры в разных vlan'ах будут взаимно недоступны (рис. 5.41).

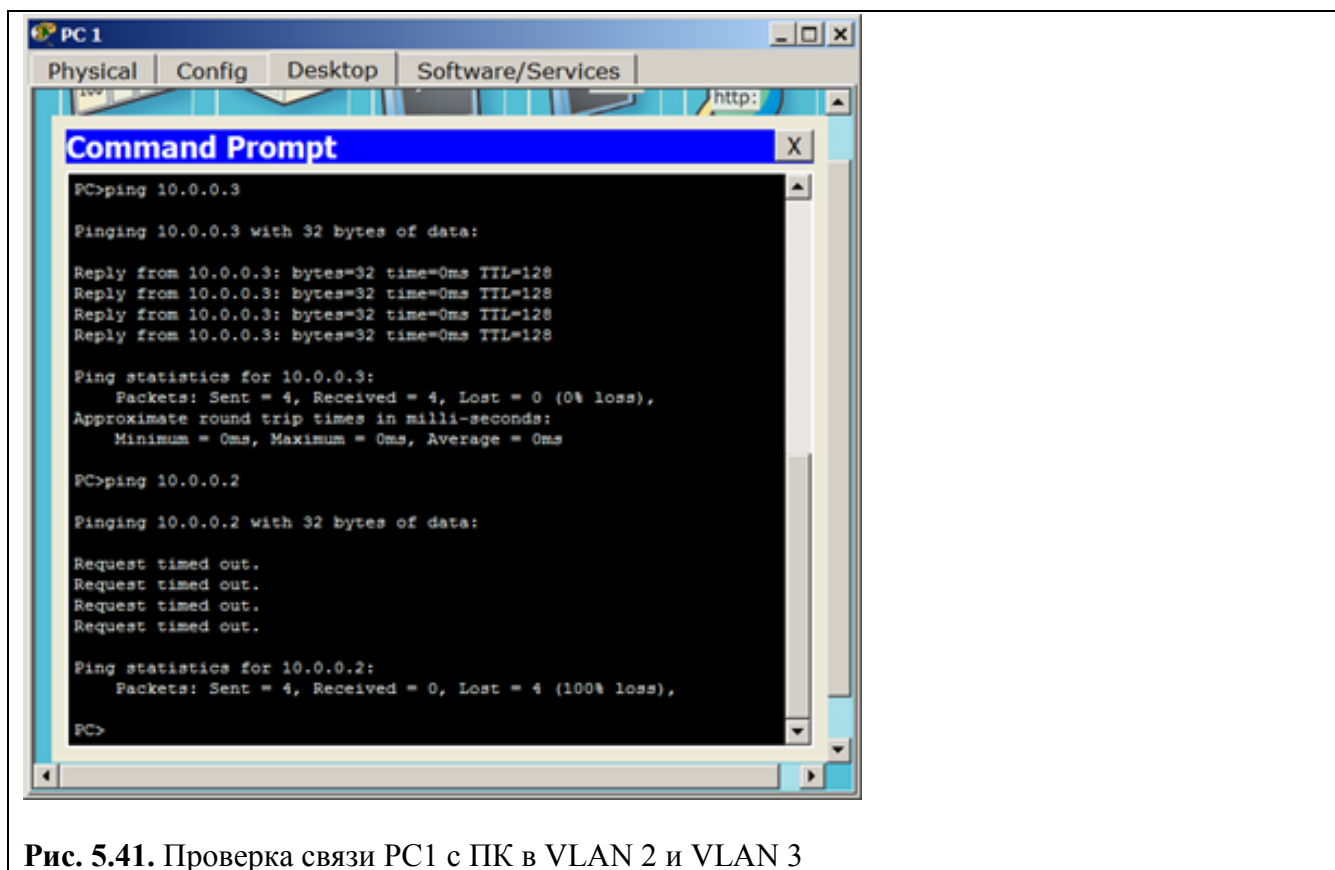


Рис. 5.41. Проверка связи PC1 с ПК в VLAN 2 и VLAN 3

Форма представления и содержание отчета

Отчет в формате DOC или PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.

### *Содержание отчета*

1. Заголовок согласно приложению.
2. Цель работы.
3. Ход работы:
  - a. Данный раздел состоит из последовательного описания значимых выполняемых шагов (с указанием их сути) Пояснения работы команды ping и **содержимго протоколов**.
  - b. копий экранов (должна быть видна набранная команда и реакция системы, если она есть).
4. По каждому пункту практической работы должны быть приведены **ВЫВОДЫ**.

## Практическая работа №7. Cisco Server. Типы серверов

**Целью** данной практической работы является знакомство с типами разных серверов и настройка на них различных сетевых сервисов.

### **Задачи:**

- Ознакомиться с различными типами серверов
- Выполнить задания по настройке различных серверов (задания 1-4)
- Составить отчет с последовательным выполнением заданий и скриншотами (задания 1-4)

### **Теоретическая часть**

Как правило, *сервер* отдает в *сеть* свои ресурсы, а клиент эти ресурсы использует. Также, на серверах устанавливаются специализированное программное и *аппаратное обеспечение*. На одном компьютере может работать одновременно несколько программ-серверов. Сервисы серверов часто определяют их название:

**Cisco HTTP (WEB) сервер** – позволяет создавать простейшие веб-странички и проверять прохождение пакетов на 80-ый *порт* сервера. Эти серверы предоставляют *доступ* к веб-страницам и сопутствующим ресурсам, например, картинкам.

**DHCP сервер** – позволяет организовывать пулы сетевых настроек для автоматического конфигурирования сетевых интерфейсов. *Dynamic Host Configuration Protocol* обеспечивает автоматическое распределение *IP*-адресов между компьютерами в сети. Такая технология широко применяется в локальных сетях с общим выходом в *Интернет*.

**DNS сервер** – позволяет организовать службу разрешения доменных имён. *Функция DNS*-сервера заключается в преобразовании доменных имен серверов в *IP*-адреса.

**Cisco EMAIL** – *почтовый сервер*, для проверки почтовых правил. Электронное *письмо* нельзя послать непосредственно получателю – сначала оно попадает на *сервер*, на котором зарегистрирована учетная *запись* отправителя. Тот, в свою *очередь*, отправляет "посылку" серверу получателя, с которого последний и забирает сообщение.

**FTP** – файловый *сервер*. В его задачи входит хранение файлов и обеспечение доступа к ним клиентских ПК, например, по протоколу *FTP*. Ресурсы *файл*-сервера могут быть либо открыты для всех компьютеров в сети, либо защищены системой идентификации и правами доступа.

Итак, эмулятор сетевой среды *Cisco Packet Tracer* позволяет проводить настройку таких сетевых сервисов, как: *HTTP, DHCP, DNS, EMAIL, FTP* и ряда других в составе сервера сети. Рассмотрим настройку некоторых из них.

## Задание 1. Настраиваем WEB сервер

Топология для наших исследований приведена на рис. 6.1.

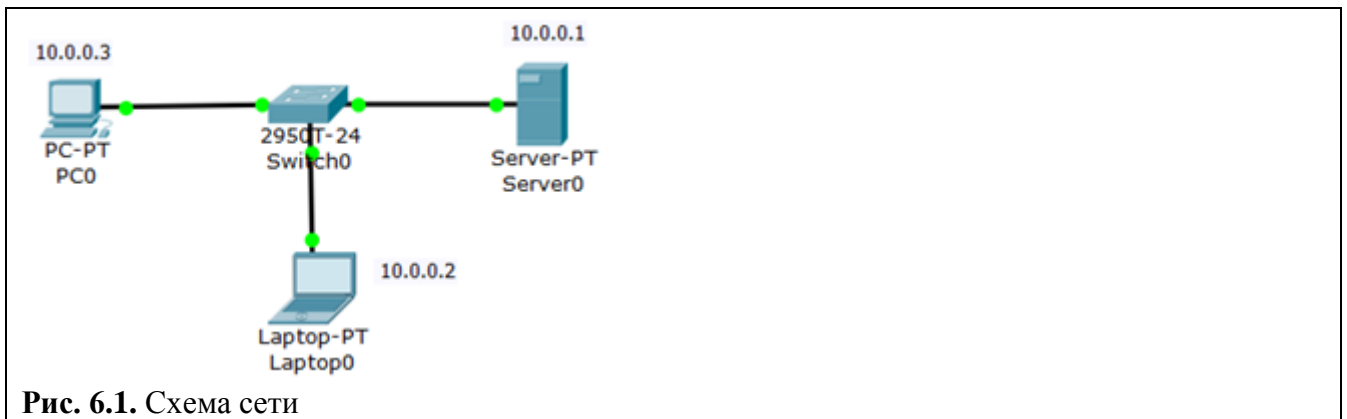


Рис. 6.1. Схема сети

### Создаем WEB-документ на сервере

Для создания HTTP-сервера открываем на сервере вкладку HTTP и редактируем первую страницу сайта с названием **index.html**. Включаем службу HTTP переключателем On ( рис. 6.2).

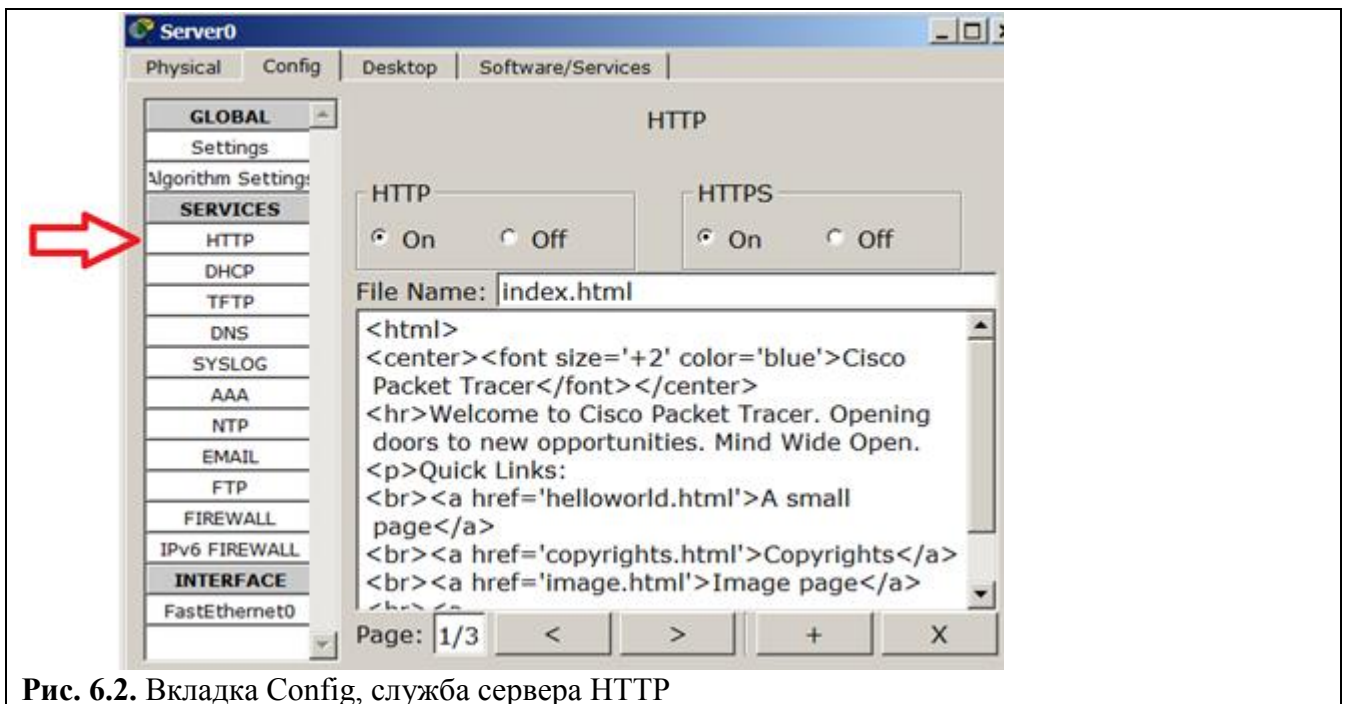


Рис. 6.2. Вкладка Config, служба сервера HTTP

### Примечание

В этом окне можно добавить новую страницу кнопкой **+** или удалить текущую кнопкой **X**. Переключение между несколькими страницами осуществляется кнопками **<** **>**.

В окне html кода создаем текст первой страницы сайта **index.html**. Вариант 1 ( рис. 6.3).

```
<html>
<body>
<h1>Welcome to WEB-Server CISCO!</h1>
<p>Server working: <font color="red"><b>OK!</b></font></p>
</body>
</html>
```

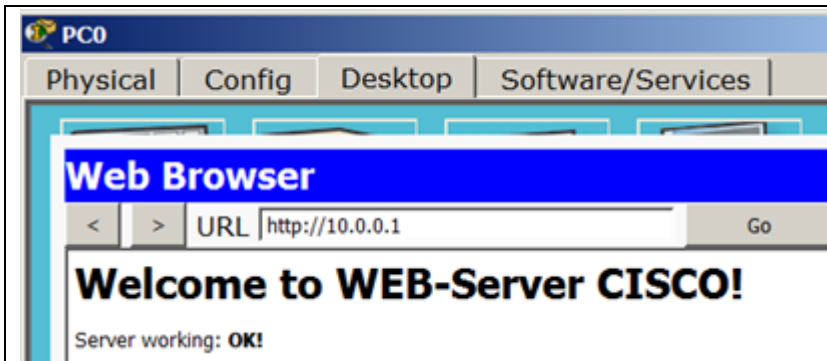


Рис. 6.3. Текст web-страницы, вариант 1

Либо вариант 2 ( рис. 6.4).

```

<html>
<center><font size='+2' color='blue'>Welcome to Cisco Packet Tracer HTML Server!
</font></center>
<body>
Hello!<br/>I am OK!
</body>
</html>

```

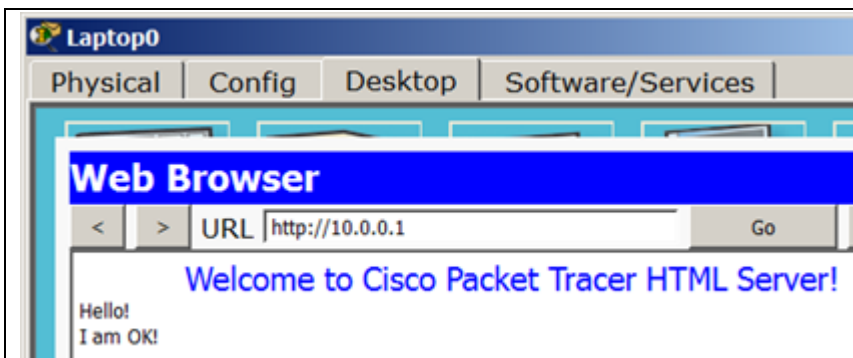


Рис. 6.4. Текст web-страницы, вариант 2

**Совет**

Текст можно переносить в это окно через буфер обмена. Он может быть только на английском языке

Для того, чтобы проверить работоспособность нашего сервера, открываем клиентскую машину (10.0.0.2 или 10.0.0.3) и на вкладке **Desktop** (Рабочий стол) запускаем приложение **Web Browser**. После чего набираем адрес нашего WEB-сервера 10.0.0.1 и нажимаем на кнопку **GO**. Убеждаемся, что наш веб-сервер работает.



## Задание 2. Настройка сетевых сервисов DNS, DHCP и Web

Создайте схему сети, представленную на рис. 6.5.

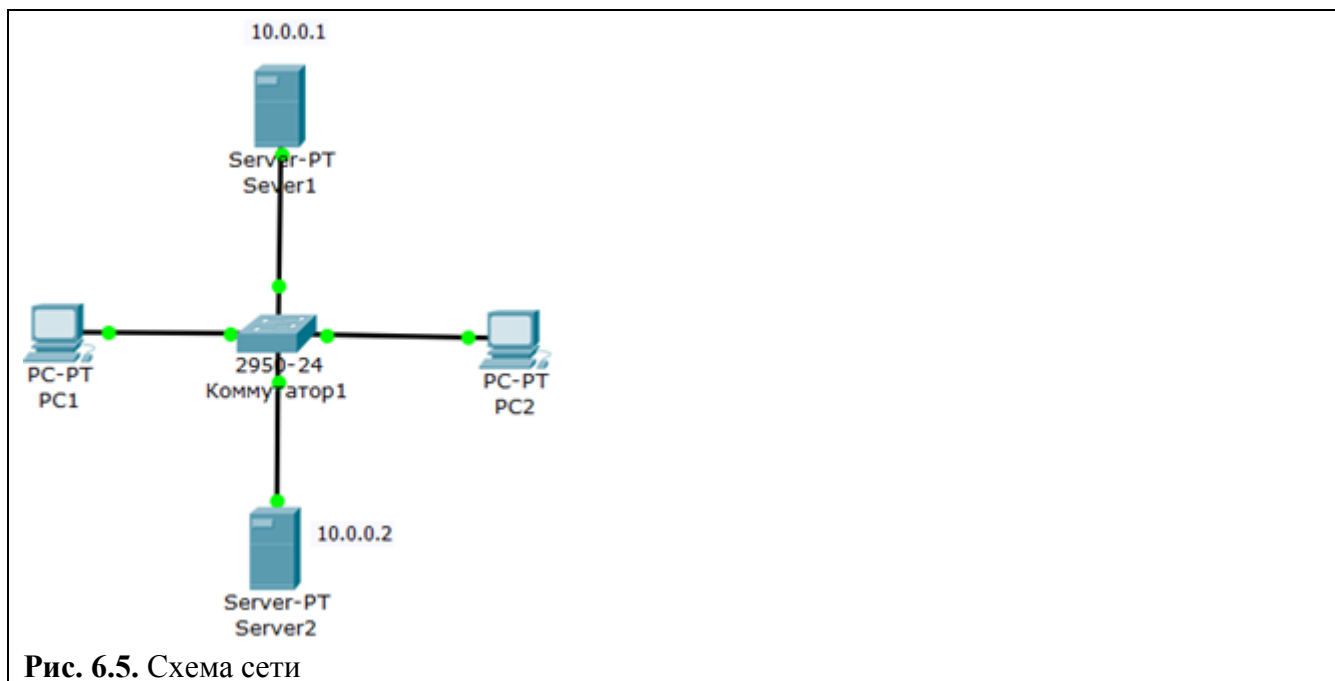


Рис. 6.5. Схема сети

Наша задача состоит в том, чтобы настроить Server1 как *DNS* и *Web-сервер*, а Server2 как *DHCP сервер*. Напомним, что работа *DNS*-сервера заключается в преобразовании доменных имен серверов в *IP*-адреса. *DHCP сервер* позволяет организовывать пулы для автоматического конфигурирования сетевых интерфейсов, то есть, обеспечивает автоматическое распределение *IP*-адресов между компьютерами в сети. Иначе говоря, в нашем случае компьютеры получают *IP*-адреса благодаря сервису *DHCP* Server2 и открывают, например, сайт на Server1.

### Настраиваем IP адреса серверов и DHCP на ПК

Войдите в конфигурацию PC1 и PC2 и установите настройку IP через DHCP сервер рис. 6.6.

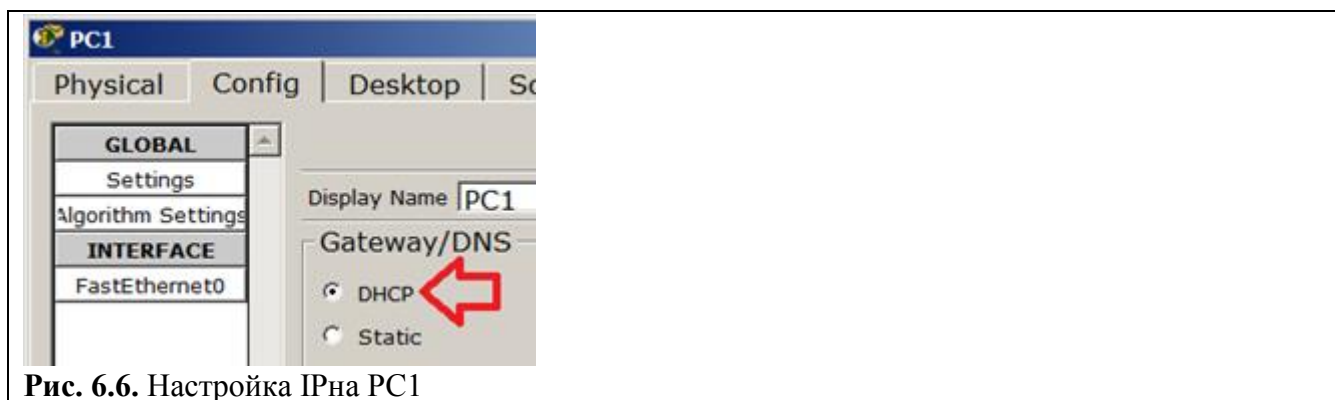


Рис. 6.6. Настройка IP на PC1

Задайте в конфигурации серверов настройки IP: Server1 – 10.0.0.1 (рис. 6.7), Server2 – 10.0.0.2 (рис. 6.8). Маска подсети установится автоматически как 255.0.0.0.



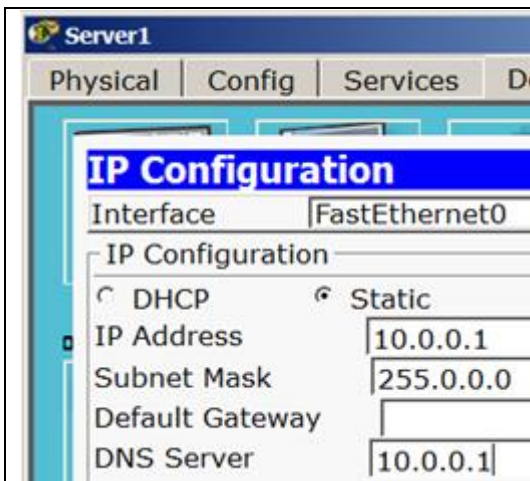


Рис. 6.7.

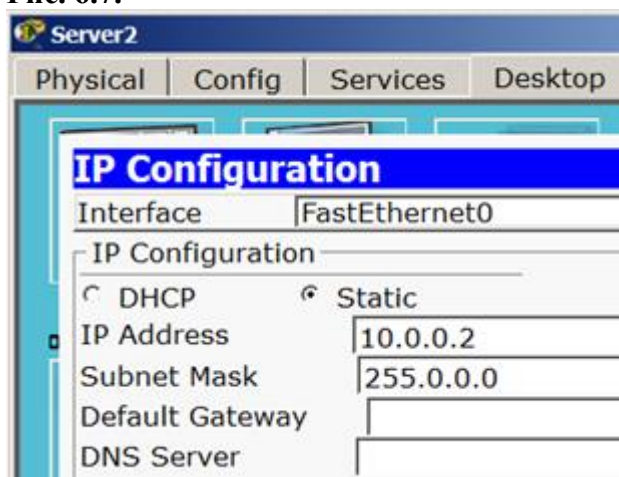


Рис. 6.8.

### Настройка служб DNS и HTTP на Server1

В конфигурации Server1 войдите на вкладку DNS и задайте две ресурсные записи (Resource Records) в прямой зоне DNS.

#### Новый термин

**Зона DNS** — часть дерева доменных имен (включая ресурсные записи), размещаемая как единое целое на сервере доменных имен (DNS-сервере). В зоне прямого просмотра на запрос доменного имени идет ответ в виде IP адреса. В зоне обратного просмотра по IP мы узнаем доменное имя ПК.

Сначала в ресурсной записи типа **A Record** свяжите доменное имя компьютера **server1.yandex.ru** с его **IP адресом 10.0.0.1** и нажмите на кнопку **Add** (добавить) и активируйте переключатель **On** – рис. 6.9.

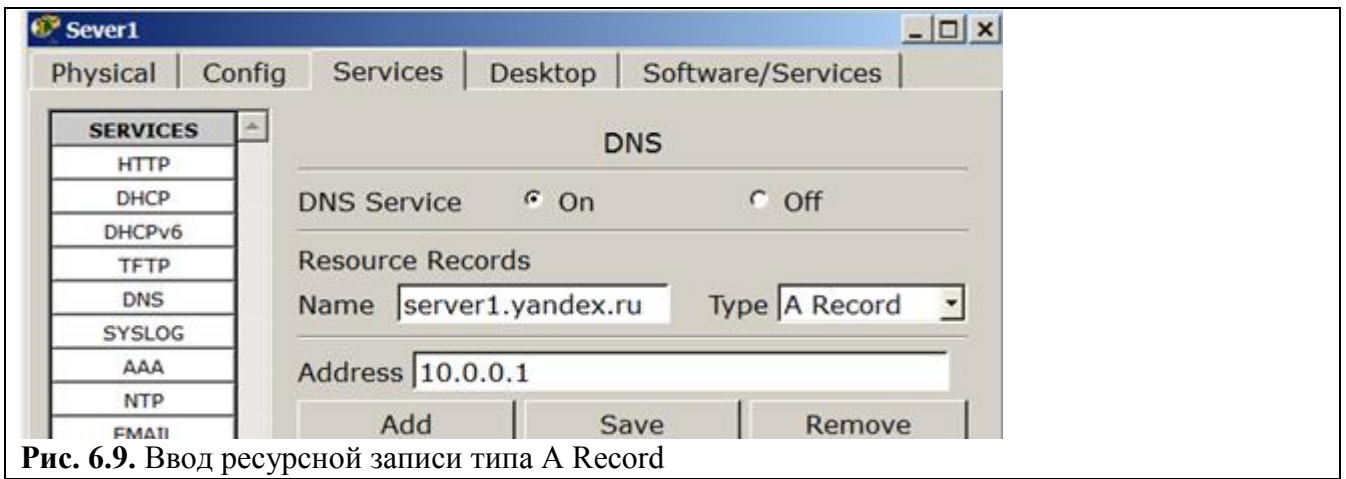


Рис. 6.9. Ввод ресурсной записи типа A Record

Далее в ресурсной записи типа **CNAME** свяжите название сайта с сервером и нажмите на кнопку **Add** (добавить) – рис. 6.10.

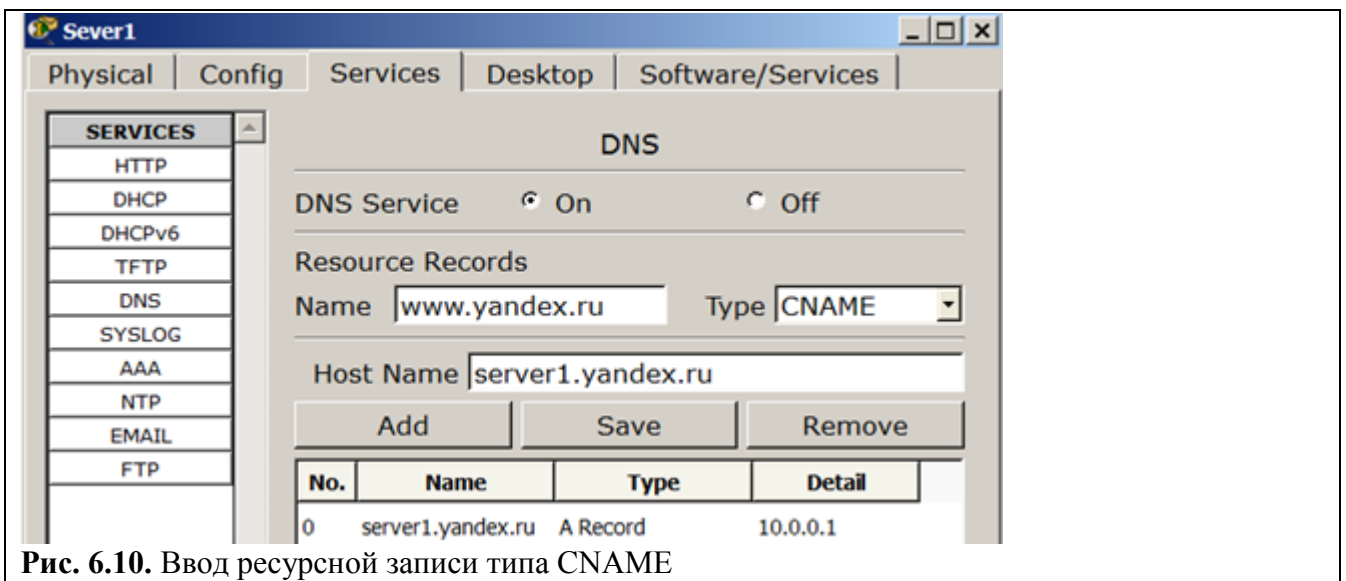


Рис. 6.10. Ввод ресурсной записи типа CNAME

В результате должно получиться следующее ( рис. 6.11).

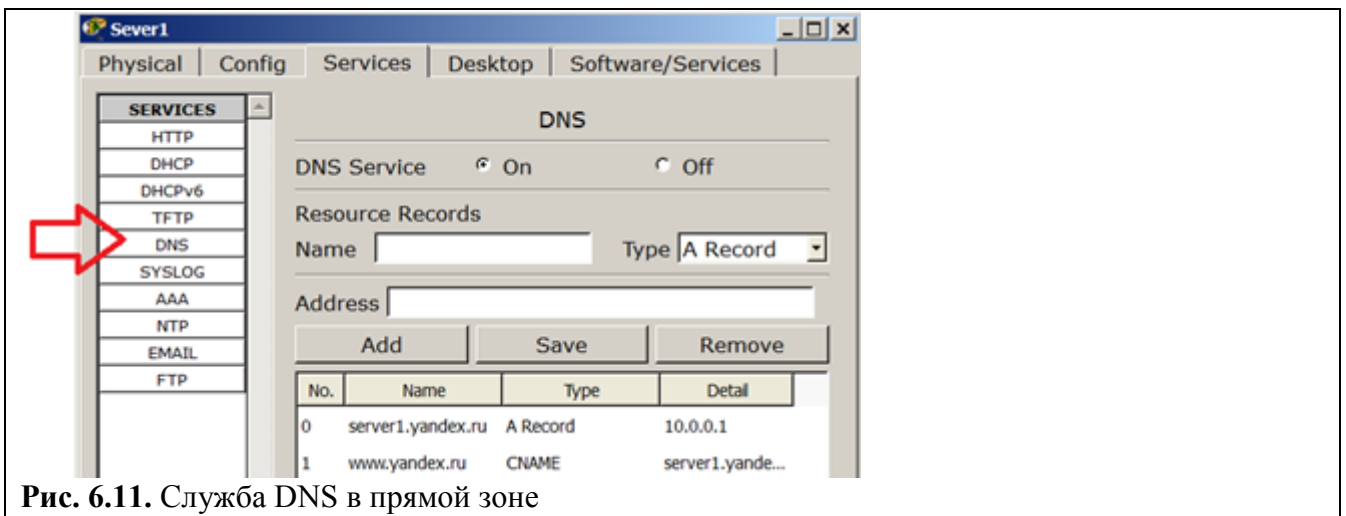


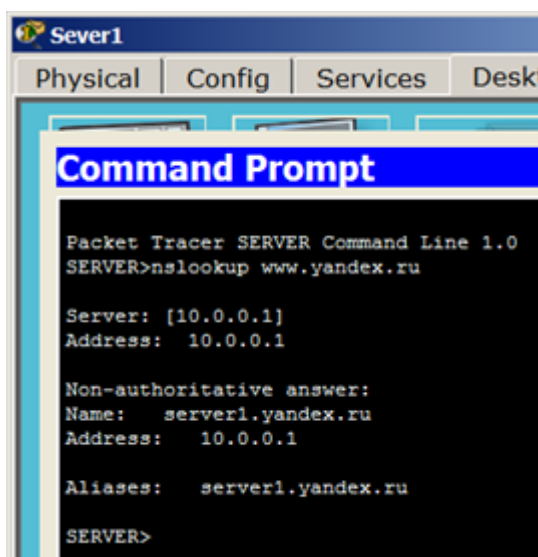
Рис. 6.11. Служба DNS в прямой зоне

Теперь настроим службу HTTP. В конфигурации Server1 войдите на вкладку HTTP и создайте стартовую страницу сайта ( рис. 6.12).

```
<html>
<center><font size='+2' color='green'>Web Server</font></center>
www.yandex.ru
<p>
Hello!<br/>I am Server1
</html>
```

Рис. 6.12. Стартовая страница сайта

Включите командную строку на Server1 и проверьте работу службы DNS. Для проверки правильности работы прямой зоны DNS сервера введите команду **SERVER>nslookup** . Если все правильно настроено, то вы получите отклик на запрос с указанием доменного имени DNS сервера в сети и его IP адреса ( рис. 6.13).



```
Sever1
Physical | Config | Services | Desk
Command Prompt
Packet Tracer SERVER Command Line 1.0
SERVER>nslookup www.yandex.ru

Server: [10.0.0.1]
Address: 10.0.0.1

Non-authoritative answer:
Name:   server1.yandex.ru
Address: 10.0.0.1

Aliases:  server1.yandex.ru
SERVER>
```

Рис. 6.13. Служба DNS в прямой зоне DNS на Server1 настроена правильно

### Примечание

Команда **nslookup** служит для определения ip-адреса по доменному имени (и наоборот).

### Настройка службы DHCP на Server2

Войдите в конфигурацию Server2 и на вкладке DHCP настройте службу DHCP. Для этого наберите новые значения пула, установите переключатель **On** и нажмите на кнопку **Save**(Сохранить) - рис. 6.14.

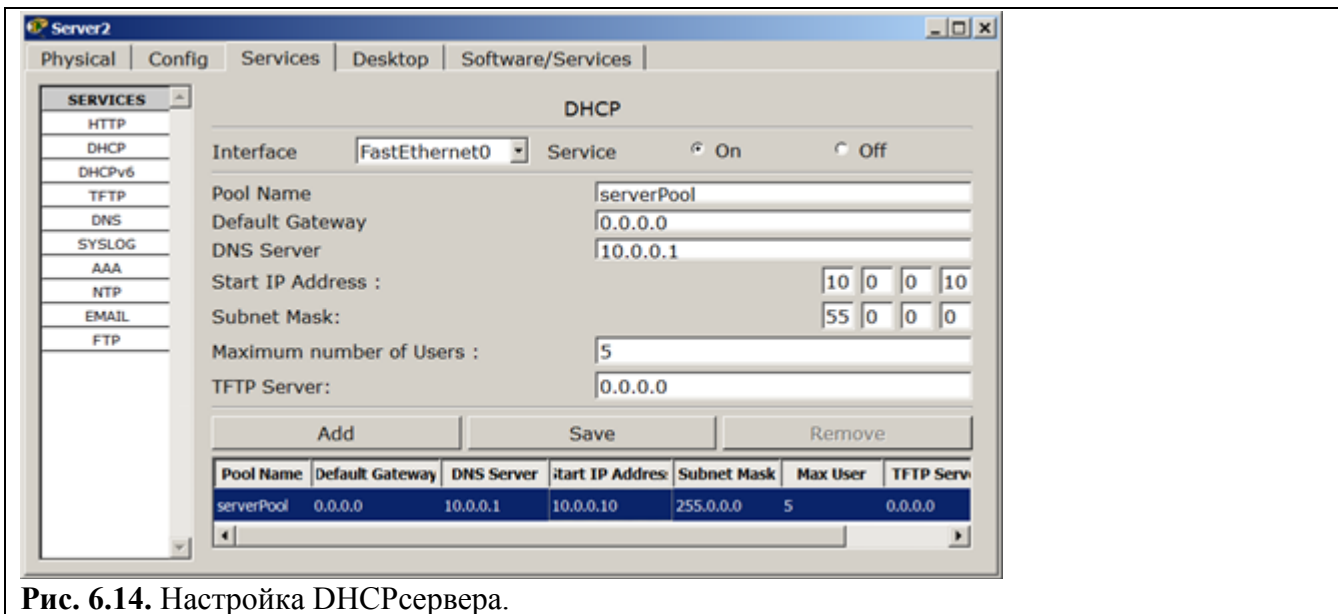


Рис. 6.14. Настройка DHCP сервера.

### Проверка работы клиентов

Войдите в конфигурации хоста PC1 и PC2 и в командной строке сконфигурируйте протокол TCP/IP. Для этого командой **PC> ipconfig /release** сбросьте (очистите) старые параметры IP адреса (рис. 6.15).



Рис. 6.15. Удаление конфигурации IP-адресов для всех адаптеров

### Примечание

Команда **ipconfig /release** отправляет сообщение **DHCP RELEASE** серверу DHCP для освобождения текущей конфигурации DHCP и удаления конфигурации IP-адресов для всех адаптеров (если адаптер не задан). Этот ключ отключает протокол TCP/IP для адаптеров, настроенных для автоматического получения IP-адресов.

Теперь командой **PC> ipconfig /renew** получите новые параметры от DHCP сервера (рис. 6.16).

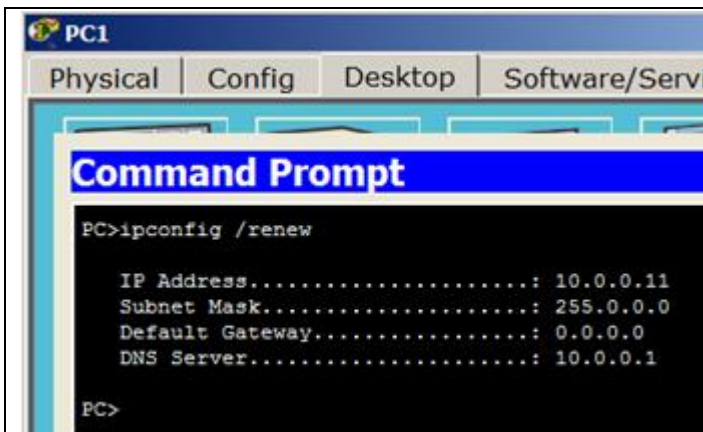


Рис. 6.16. Конфигурация протокол TCP/IP клиента от DHCP сервера

Аналогично поступите для PC2 ( рис. 6.17).

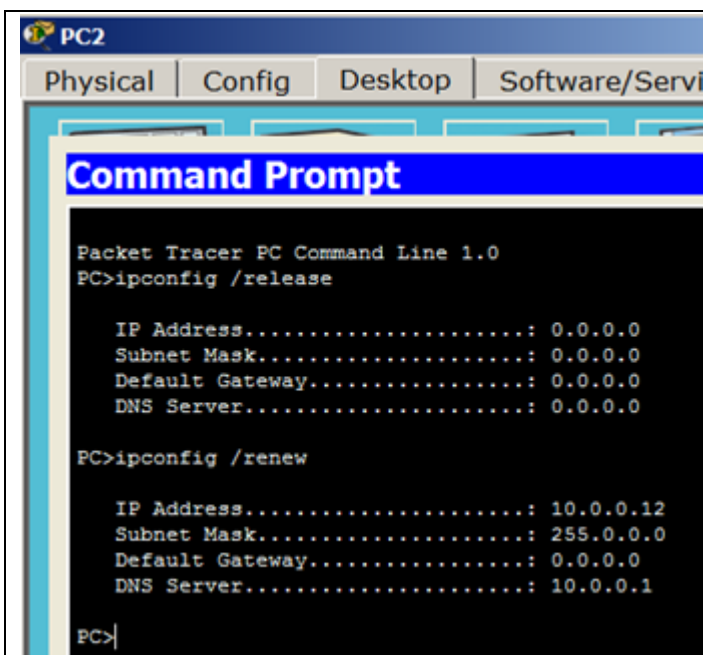


Рис. 6.17. PC2 получил IP адрес от DHCP сервера Server2

Осталось проверить работу WEB сервера Server1 и открыть сайт в браузере на PC1 или PC2 ( рис. 6.18).

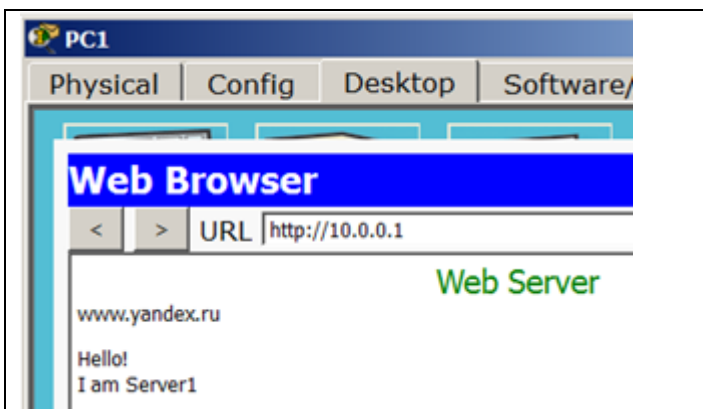


Рис. 6.18. Проверка работы службы HTTP на Server1

### Задание 3. Конфигурирование DHCP сервера на маршрутизаторе

Схема сети приведена на рис. 6.19. С помощью настроек ПК, представленных на рисунке, мы указываем хосту, что он должен получать IP *адрес*, *адрес* основного шлюза и *адрес DNS* сервера от *DHCP* сервера.

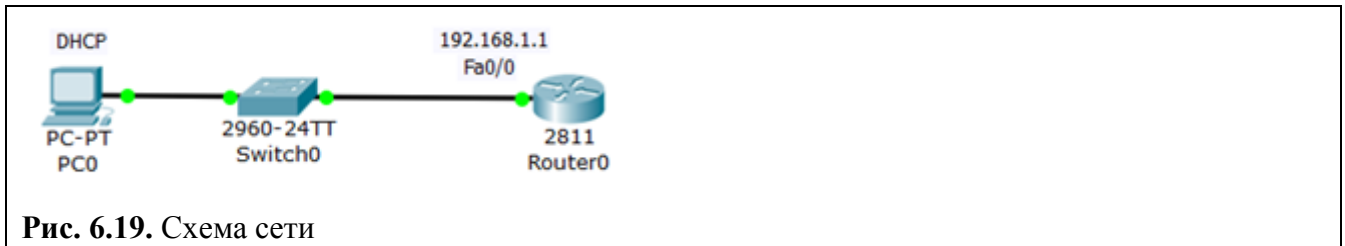


Рис. 6.19. Схема сети

Предварительно настройте устройства по схеме сети (рис 6.19)  
Только затем приступайте дальше

Произведем настройку R0:

**Router (config)#ip dhcp pool TST** создаем *пул IP* адресов для *DHCP* сервера с именем *TST*

**Router (dhcp-config)#network 192.168.1.0 255.255.255.0** указываем из какой сети мы будем раздавать *IP* адреса (первый *параметр* - *адрес* данной сети, а второй *параметр* ее *маска*)

**Router (dhcp-config)#default-router 192.168.1.1** указываем *адрес* основного шлюза, который будет рассылать в сообщениях *DHCP*

**Router (dhcp-config)#dns-server 5.5.5.5** указываем *адрес DNS* сервера, который так же будет рассылаться хостам в сообщениях *DHCP*

**Router (dhcp-config)#exit**

**Router (config)#ip dhcp excluded-address 192.168.1.1** этот *хост* исключен из пула, то есть, ни один из хостов сети не получит от *DHCP* сервера этот *адрес*.

Полный листинг этих команд приведен на рис. 6.20.

```
Router0
Physical | Config | CLI |
IOS Command Line Interface

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip dhcp pool tst
Router(dhcp-config)#network 192.168.1.1 255.255.255.0
Router(dhcp-config)#default-router 192.168.1.1
Router(dhcp-config)#dns-server 5.5.5.5
Router(dhcp-config)#exit

Router(config)#ip dhcp excluded-address 192.168.1.1
Router(config)#
```

Рис. 6.20. Команды для конфигурирования R0

Проверим результат получения динамических параметров для PC0 (рис. 6.21).



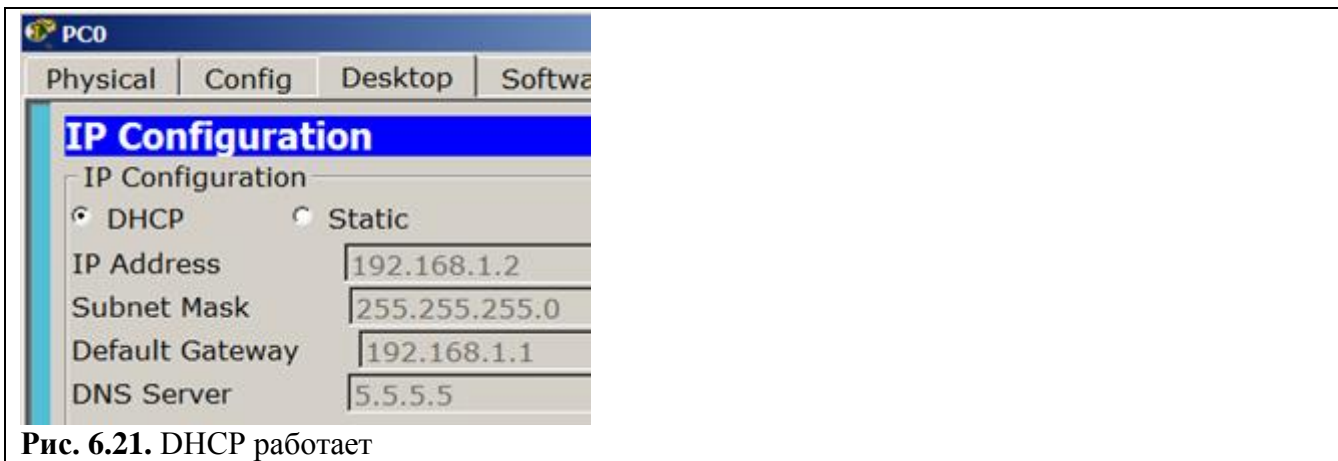


Рис. 6.21. DHCP работает

Проверим работоспособность *DHCP* сервера на хосте PC0 командой **ipconfig /all** ( рис. 6.22).

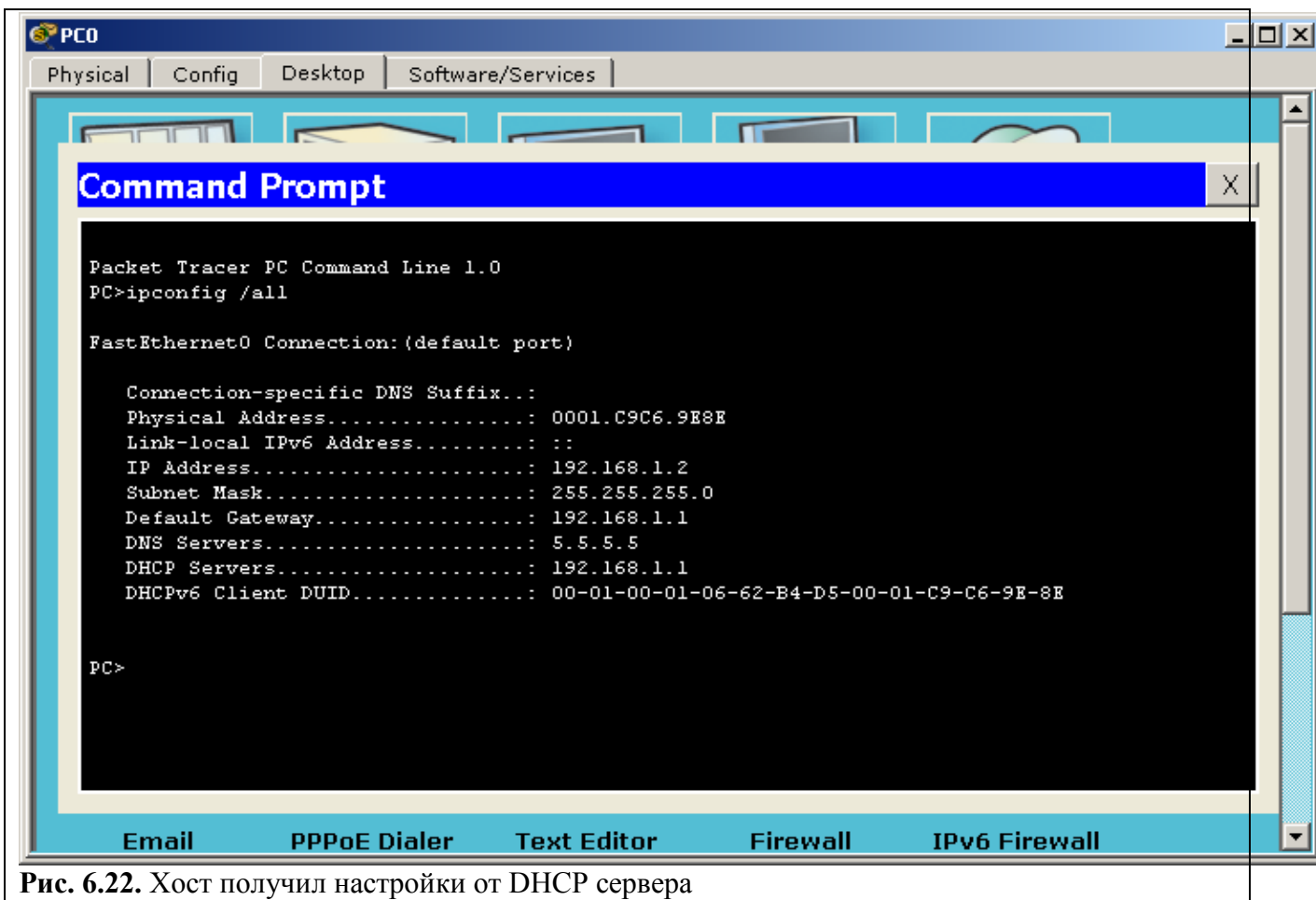


Рис. 6.22. Хост получил настройки от DHCP сервера

*Хост* успешно получил *IP* адрес, адрес шлюза и адрес *DNS* сервера от *DHCP* сервера R0.

## Задание 4. DHCP сервис на маршрутизаторе 2811

В этом примере мы будем конфигурировать *маршрутизатор* 2811, а именно, настраивать на нем *DHCP сервер*, который будет выдавать по *DHCP* адреса из сети 192.168.1.0 (рис. 6.26). PC1 и PC2 будут получать настройки динамически, а для сервера желательно иметь постоянный *адрес*, т.е., когда он задан статически.

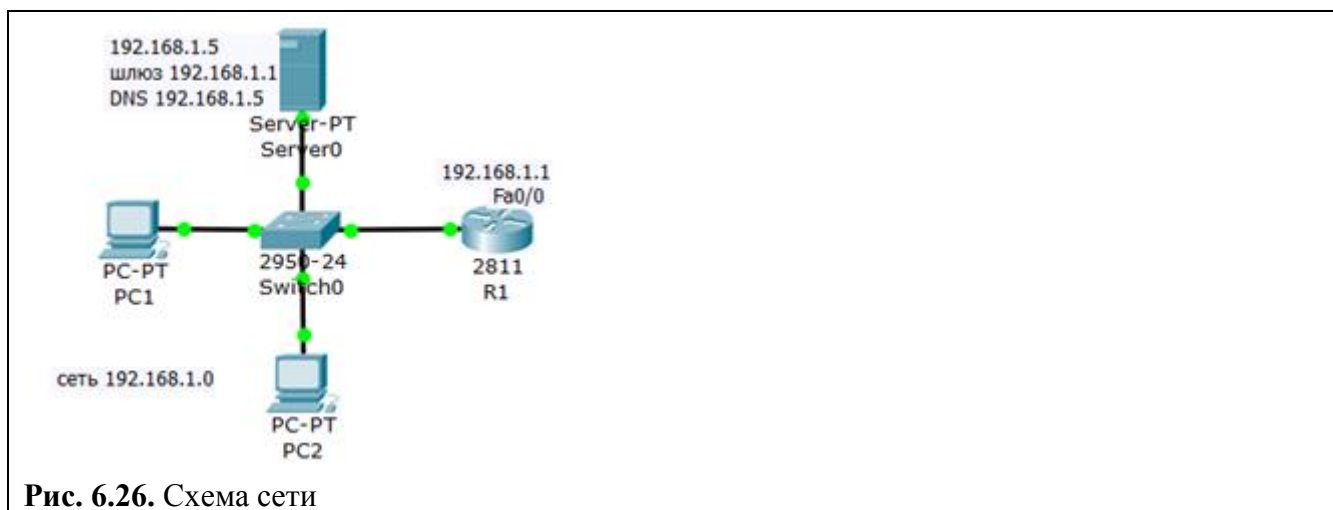


Рис. 6.26. Схема сети

### Примечание

Как устройство с постоянным адресом здесь можно включить еще и принтер.

### Резервируем 10 адресов

```
R1 (config)#ip dhcp excluded-address 192.168.1.1 192.168.1.10
```

### Примечание

Этой командой мы обязали маршрутизатор R1 не выдавать адреса с 192.168.1.1 по 192.168.1.10 потому, что адрес 192.168.1.1 будет использоваться самим маршрутизатором как шлюз, а остальные адреса мы резервируем под различные hosts этой сети.

Таким образом, первый *DHCP адрес*, который выдаст R1 равен **192.168.1.11**.

### Создаем пул адресов, которые будут выдаваться из сети 192.168.1.0

```
R1 (config)#ip dhcp pool POOL1  
R1 (dhcp-config)#network 192.168.1.0 255.255.255.0  
R1 (dhcp-config)#default-router 192.168.1.1  
R1 (dhcp-config)#dns-server 192.168.1.5
```

### Примечание

Согласно этим настройкам выдавать адреса из сети 192.168.1.0 (кроме тех, что мы исключили) будет маршрутизатор R1 через шлюз 192.168.1.1.

### Настраиваем интерфейс маршрутизатора

```
R1 (config)#interfacefa0/0  
R1 (config-if)#ip address 192.168.1.1 255.255.255.0  
R1 (config-if)#no shutdown  
R1 (config-if)#exit  
R1(config)#exit  
R1#
```

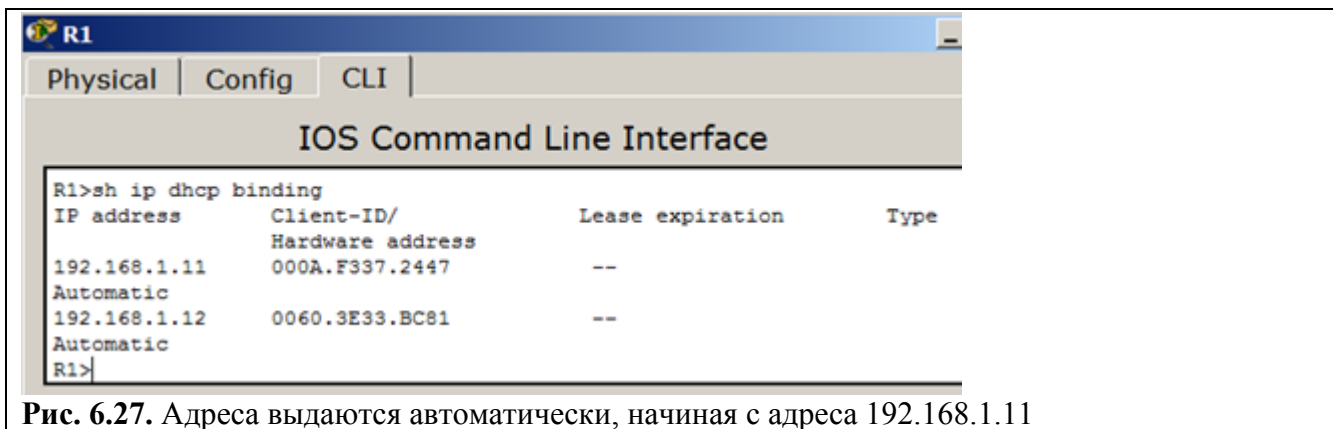


### Примечание

Команда **no shut** (сокращение от no shutdown) используется для того, чтобы бы интерфейс был активным. Обратная команда – shut, выключит интерфейс.

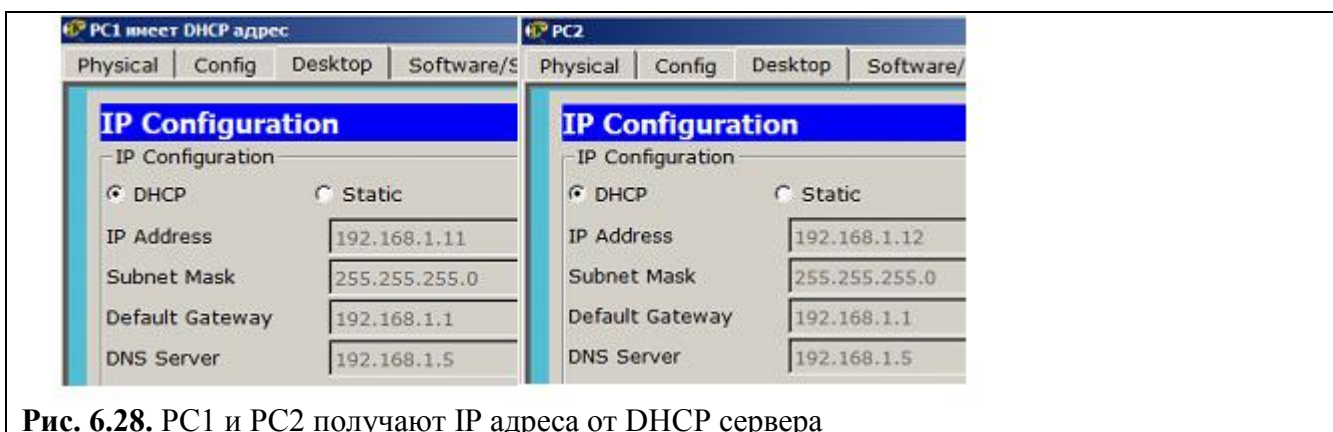
### Проверка результата

Теперь оба ПК получили настройки и командой **R1#show ip dhcp binding** можно посмотреть на *список* выданных роутером адресов ( рис. 6.27).



**Рис. 6.27.** Адреса выдаются автоматически, начиная с адреса 192.168.1.11

Итак, мы видим, что протокол *DHCP* позволяет производить автоматическую настройку сети на всех компьютерах ( рис. 6.28).



**Рис. 6.28.** PC1 и PC2 получают IP адреса от DHCP сервера

### Форма представления и содержание отчета

Отчет в формате DOC или PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.

### Содержание отчета

1. Заголовок согласно приложению.
2. Цель работы.
3. Ход работы:
  - a. Данный раздел состоит из последовательного описания значимых выполняемых шагов (с указанием их сути) Пояснения работы команды ping и **содержимго протоколов**.
  - b. копий экранов (должна быть видна набранная команда и реакция системы, если она есть).

4. По каждому пункту практической работы должны быть приведены **выводы**.

## Практическая работа №8. Статическая маршрутизация

**Целью** данной практической работы является получение навыков по настройке статической маршрутизации внутри различных сетей

### **Задачи:**

- Ознакомиться с понятием статической и динамической маршрутизации и их различиями
- Выполнить задания по настройке статической маршрутизации (задания 1-3)
- Составить отчет с последовательным выполнением заданий и скриншотами (задания 1-3)

### **Теоретическая часть**

*Таблица* маршрутизации может составляться двумя способами: статично и динамично.

В случае *статической маршрутизации* записи в таблице вводятся и изменяются вручную. Такой способ требует вмешательства администратора каждый раз, когда происходят изменения в топологии сети. С другой стороны, он является наиболее стабильным и требующим минимума аппаратных ресурсов маршрутизатора для обслуживания таблицы.

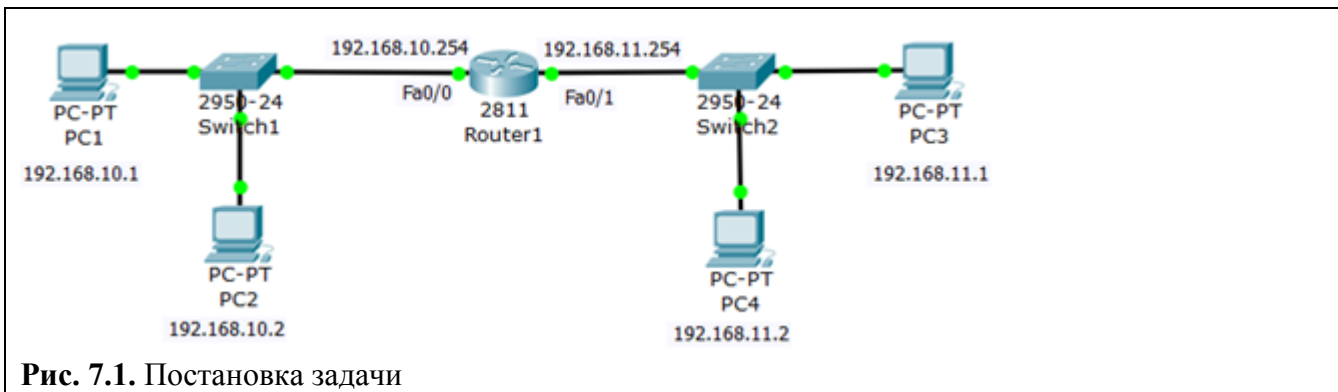
При *динамической маршрутизации* записи в таблице обновляются автоматически при помощи одного или нескольких протоколов маршрутизации — *RIP, OSPF, IGRP, EIGRP* и др. Кроме того, *маршрутизатор* строит таблицу оптимальных путей к сетям назначения на основе различных критериев (метрик), таких, как: количества промежуточных узлов, пропускной способности каналов, задержки передачи данных и т. п. *Динамическая маршрутизация* оказывает дополнительную нагрузку на устройства, а высокая нестабильность сети может приводить к ситуациям, когда маршрутизаторы не успевают синхронизировать свои таблицы, что приводит к противоречивым сведениям о топологии сети в различных её частях и потере передаваемых данных.

*Статическая маршрутизация* — вид маршрутизации, при котором информация о маршрутах заносится в таблицы маршрутизации каждого маршрутизатора вручную администратором сети. Отсюда сразу же вытекает ряд недостатков. Прежде всего это очень плохая масштабируемость сетей, так как при добавлении N+1 сети потребуется сделать  $2*(N+1)$  записей о маршрутах. Но, при использовании статических записей процессору маршрутизатора не требуется производить никаких расчетов, связанных с определением маршрутов – это плюс.

Статическая *маршрутизация* успешно используется при организации работы компьютерных сетей небольшого размера (1-2 маршрутизатора), в силу легкости конфигурации и отсутствии дополнительной нагрузки на *сеть* в виде широковещательного служебного трафика, характерного для динамических протоколов маршрутизации. Также статическая *маршрутизация* используется на компьютерах внутри сети. В таком случае обычно задается *маршрут* шлюза по умолчанию.

### **Задание 1. Настраиваем связь двух сетей через маршрутизатор**

Построим такую *сеть* ( рис. 7.1).

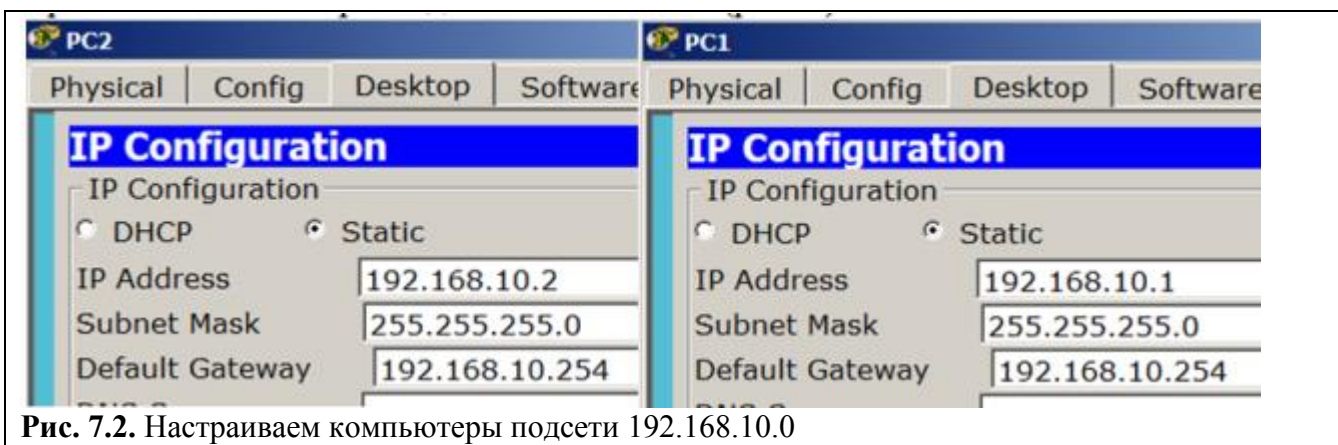


**Рис. 7.1.** Постановка задачи

Наша цель – настроить *связь* двух сетей через *маршрутизатор* (роутер).

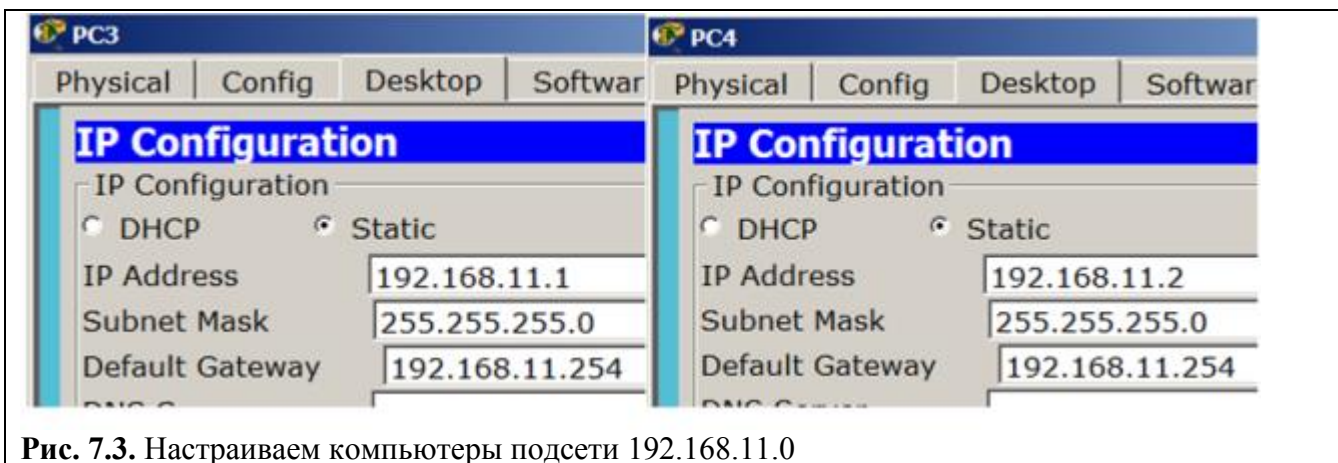
### Шаг 1. Настройка ПК

Настраиваем компьютеры подсети 192.168.10.0 ( рис. 7.2).



**Рис. 7.2.** Настраиваем компьютеры подсети 192.168.10.0

Настраиваем компьютеры подсети 192.168.11.0 ( рис. 7.3).



**Рис. 7.3.** Настраиваем компьютеры подсети 192.168.11.0

### Шаг 2. Настройка роутера (маршрутизатора)

Настраиваем роутер (маршрутизатор) как шлюз 192.168.10.254 для первой сети на интерфейсе Fa0/0 ( рис. 7.4).

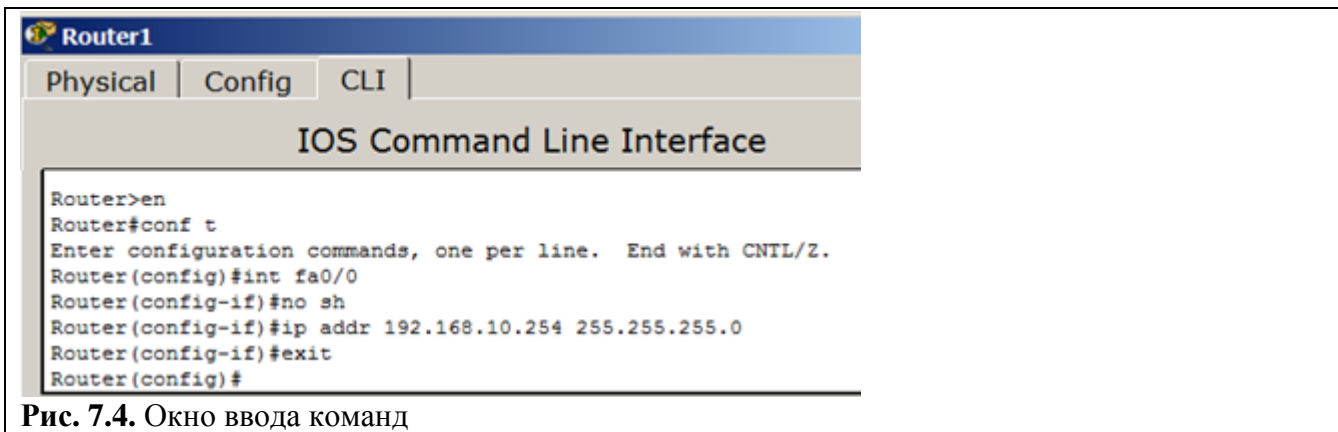


Рис. 7.4. Окно ввода команд

### Примечание

Здесь описаны следующие команды: привилегированный режим, режим конфигурирования, заходим на интерфейс, включаем этот интерфейс, задаем IP-адрес и маску порта, выходим.

Аналогично настраиваем роутер как шлюз 192.168.11.254 для второй сети на интерфейсе Fa0/1 (рис. 7.5).

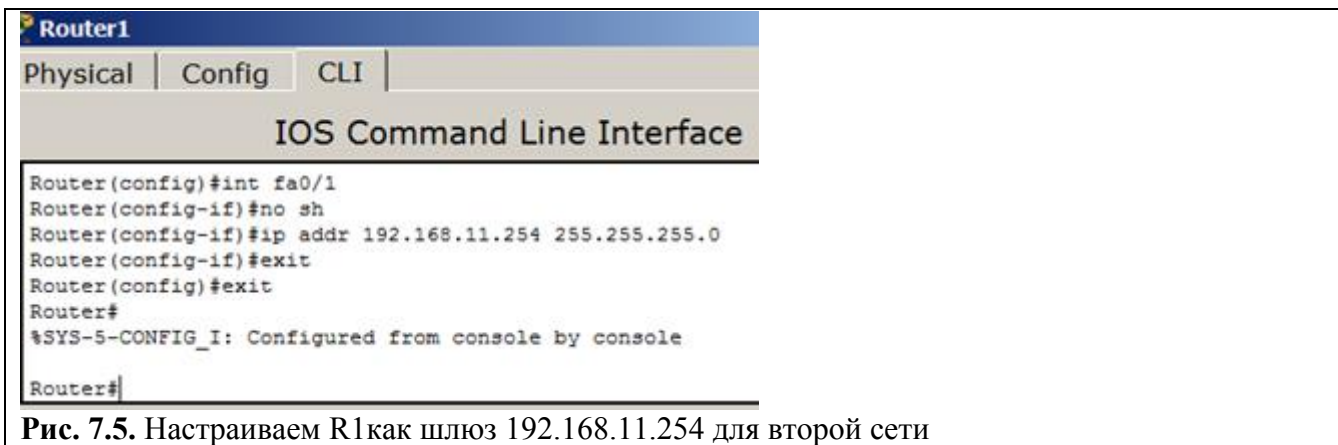


Рис. 7.5. Настраиваем R1 как шлюз 192.168.11.254 для второй сети

### Шаг 3. Проверка связи сетей

Проверяем таблицу маршрутизации командой **show ip route** (рис. 7.6).

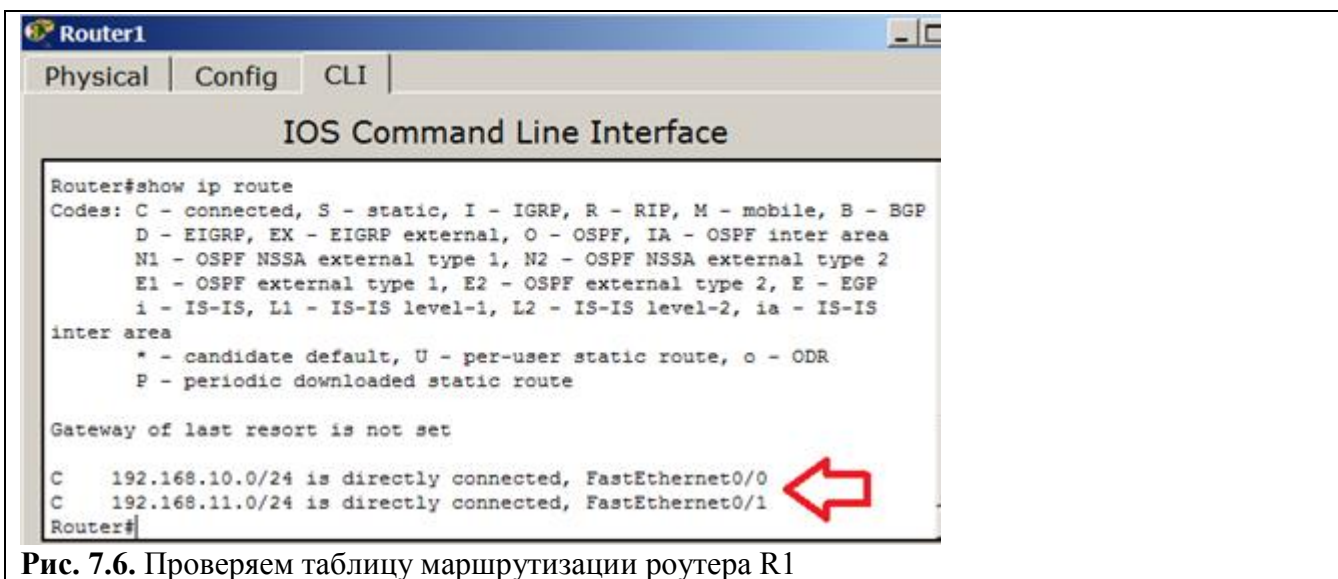


Рис. 7.6. Проверяем таблицу маршрутизации роутера R1

У нас роутер обслуживает две сети. Проверяем связь роутера и ПК (рис. 7.7).

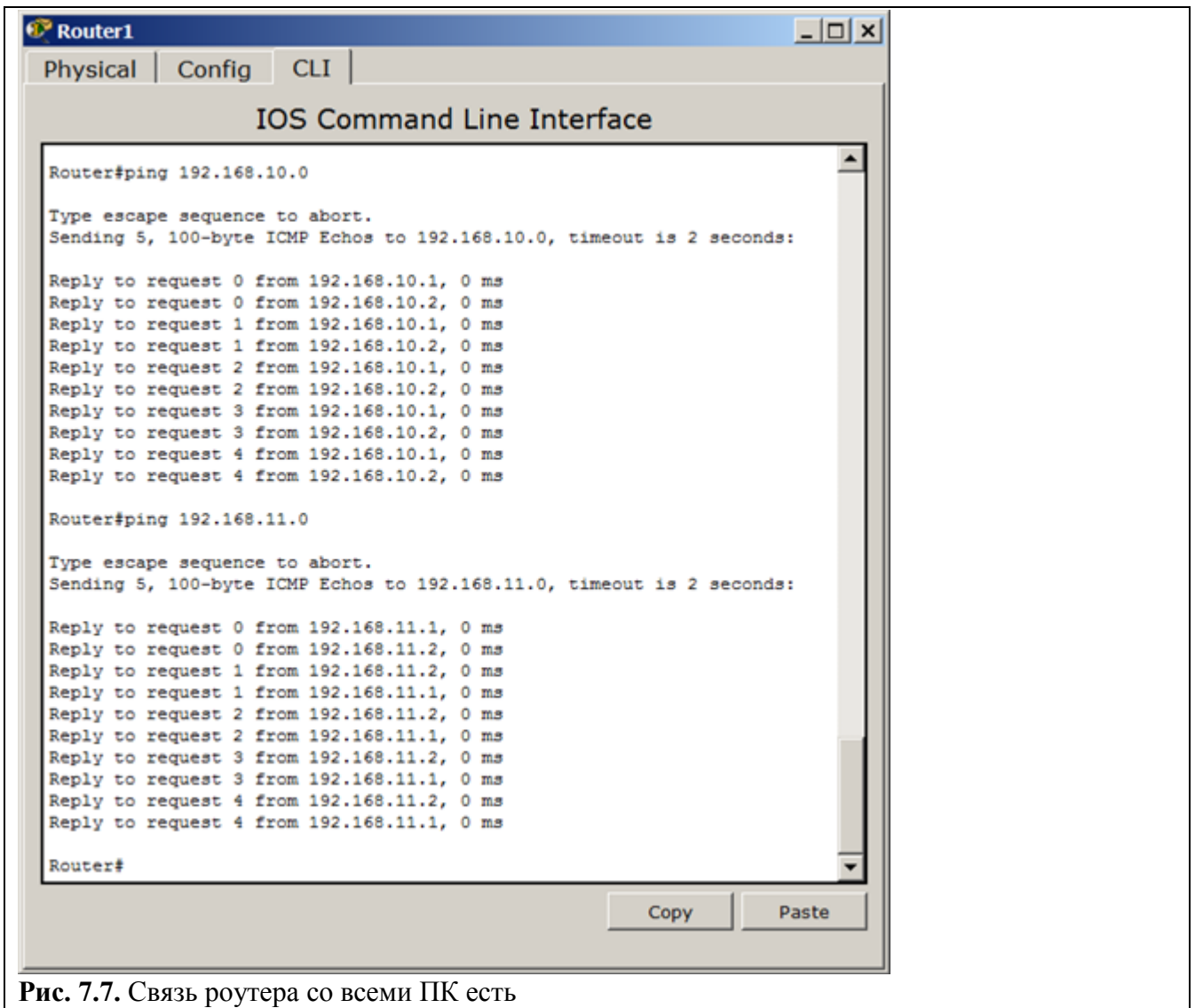


Рис. 7.7. Связь роутера со всеми ПК есть

Проверяем связь роутера с подсетями ( рис. 7.8).

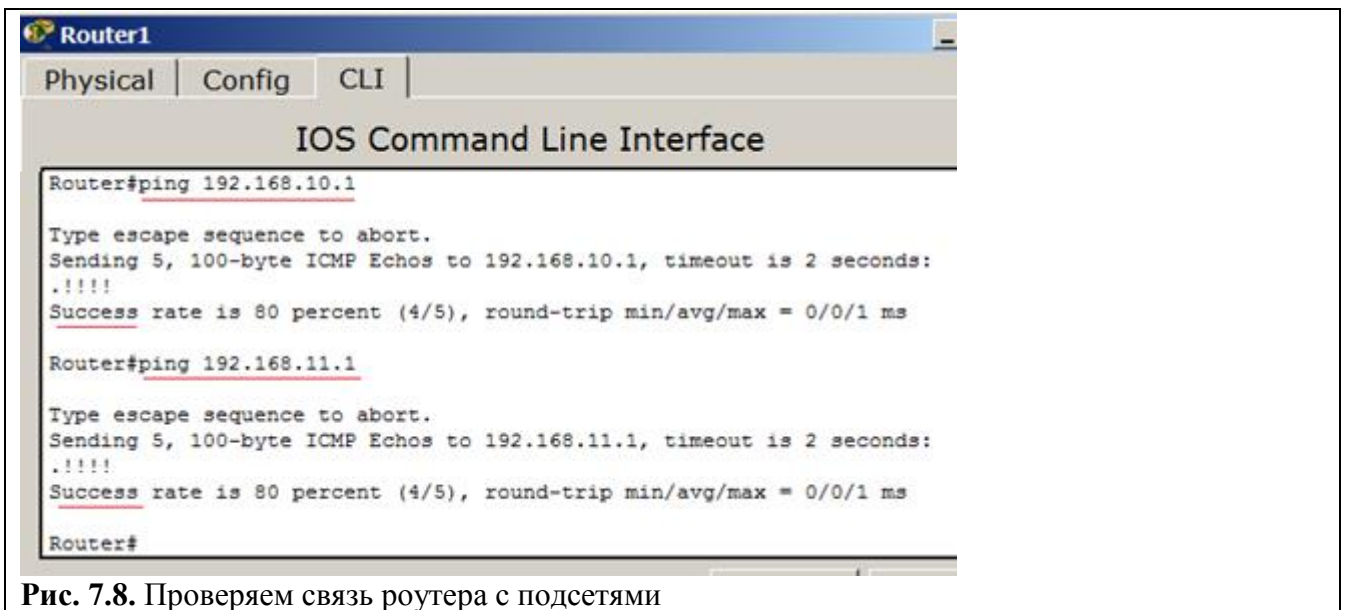


Рис. 7.8. Проверяем связь роутера с подсетями



## Примечание

Команда ping посылает ICMP эхо-пакеты для верификации соединения. В приведённом выше примере время прохождения одного эхо-пакета превысило заданное, о чём свидетельствует точка (.) в выведенной информации, а четыре пакета прошли успешно, о чём говорит восклицательный знак (!).

Проверим также связь ПК из разных сетей между собой ( рис. 7.9).

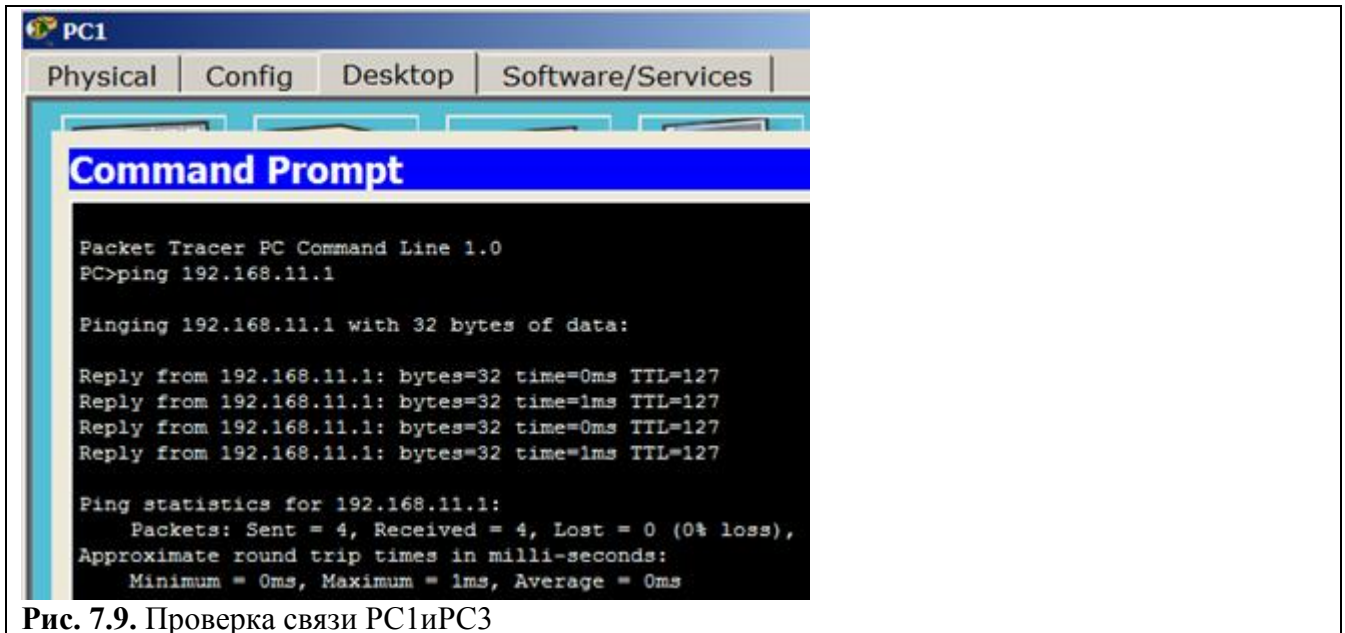


Рис. 7.9. Проверка связи PC1иPC3

## Специальные термины и понятия

**Маршрутизатором (иллюзом)**, называется узел сети с несколькими IP-интерфейсами (содержащими свой MAC-адрес и IP-адрес), подключенными к разным IP-сетям, осуществляющий на основе решения задачи маршрутизации перенаправление дейтаграмм из одной сети в другую для доставки от отправителя к получателю. Как уже отмечалось, **динамическая маршрутизация** — это процесс протокола маршрутизации, определяющий взаимодействие устройства с соседними маршрутизаторами. **Маршрутизатор** будет обновлять сведения о каждой подключенной к нему сети. Если в сети произойдет изменение, протокол динамической маршрутизации автоматически информирует об изменении все маршрутизаторы. Если же используется **статическая маршрутизация**, обновить таблицы маршрутизации на всех устройствах придется системному администратору. Статическая **маршрутизация** позволяет сократить объем таблиц маршрутизации в конечных узлах и маршрутизаторах за счет использования в качестве номера сети назначения так называемого **маршрута по умолчанию** – *default* (0.0.0.0), который обычно занимает в таблице маршрутизации последнюю строку. Если в таблице маршрутизации есть такая **запись**, то все пакеты с номерами сетей, которые отсутствуют в таблице маршрутизации, передаются маршрутизатору, указанному в строке *default*.

**Шлюз по умолчанию** (defaultgateway) - адрес маршрутизатора, на который отправляется трафик для которого не нашлось отдельных записей в таблице маршрутизации. Для устройств, подключенных к одному маршрутизатору (как правило, это рабочие станции) использование шлюза по умолчанию — единственная форма маршрутизации.

Доступность компьютера проверяется при помощи посылки контрольного диагностического сообщения по протоколу **ICMP** (*Internet Control Message Protocol*), по которому любая оконечная станция должна выдать эхо-ответ узлу, отправившему такое сообщение. В сетях на основе **TCP/IP** для проверки соединений обычно используется **утилита ping**. Эта программа отправляет запросы (**ICMP Echo-Request**) протокола **ICMP** узлу сети с

указанным IP-адресом. Получив этот *запрос*, исследуемый узел должен послать пакет с ответом (*ICMP Echo-Reply*). Первый узел фиксирует поступающие ответы. **Время между отправкой запроса и получением ответа (RTT**, от англ. Round Trip Time) позволяет определять двусторонние задержки (*RTT*) по маршруту и частоту потери пакетов, то есть косвенно определить загруженность каналов передачи данных и промежуточных устройств. **Метрика** — числовой коэффициент, влияющий на выбор маршрута в компьютерных сетях. Как правило, определяется количеством "хопов" (ретрансляционных переходов) до сети назначения или параметрами канала связи. Чем *метрика* меньше, тем *маршрут* приоритетнее. **Петля маршрутизации** — явление, возникающее, когда *маршрутизатор* отправляет пакет на неверный *адрес* назначения. Получивший такой пакет *маршрутизатор* возвращает его обратно. Таким образом получается петля. Для борьбы с подобными петлями в *TCP/IP* предусмотрен механизм *TTL*. Протоколы маршрутизации так же предлагают свои способы борьбы с петлями.

## Задание 2. Настройка трех сетей с WEB сервером. Понятие маршрута по умолчанию

Схема у нас будет следующая: два коммутатора 2950-24, два ПК в сети 192.168.10.0 с маской 255.255.255.0. *Сервер* и *компьютер* в сети 192.168.20.0 с маской 255.255.255.0. *Сеть* между маршрутизаторами (марки 1841) 192.168.1.0 с маской 255.255.255.252. Компьютеры из сети 192.168.10.0 должны достигать к DNS серверу в сети 192.168.20.0 ( рис. 7.12).

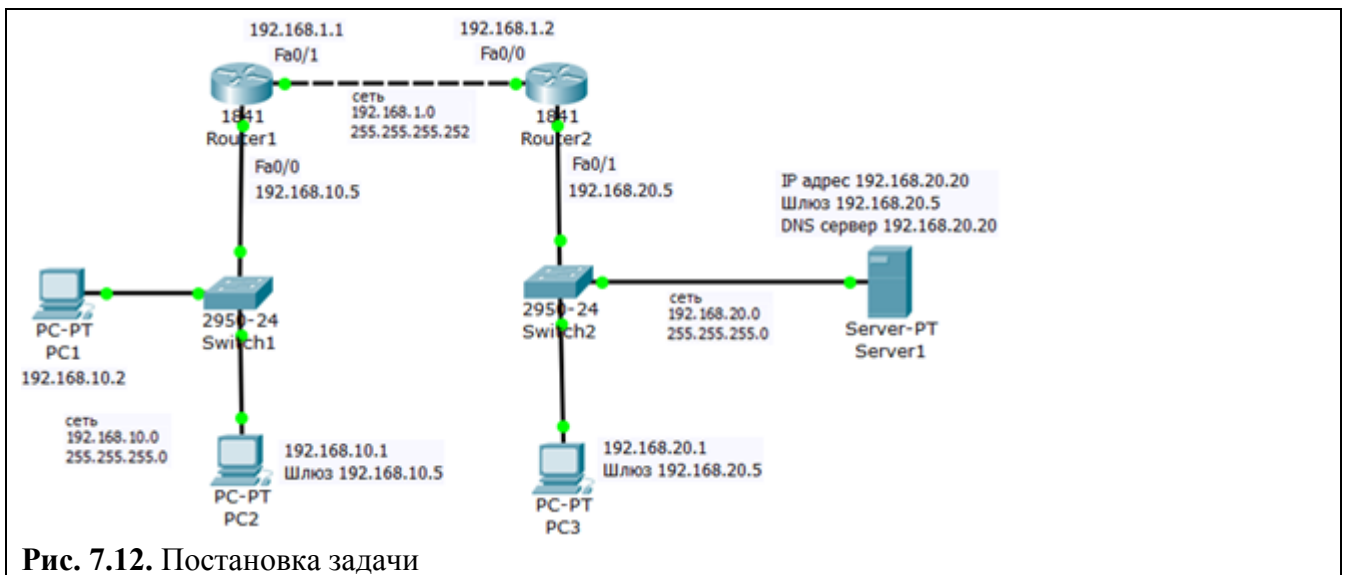


Рис. 7.12. Постановка задачи

*Сеть* у нас не сложная, ПК в ней немного, поэтому будем использовать не динамическую, а статическую маршрутизацию.

## Настройки сетевых интерфейсов роутеров

Будем настраивать связь роутеров через порты Fa0/1 для R1 и Fa0/0 для R2. Настраиваем Router1 исходя из постановки задачи о том, что сеть между маршрутизаторами 192.168.1.0 с маской 255.255.255.252. Поэтому порту Fa0/1 присвоим IP адрес 192.168.1.1 ( рис. 7.13).



Рис. 7.13. Настраиваем порт 0/1 для маршрутизатора R1



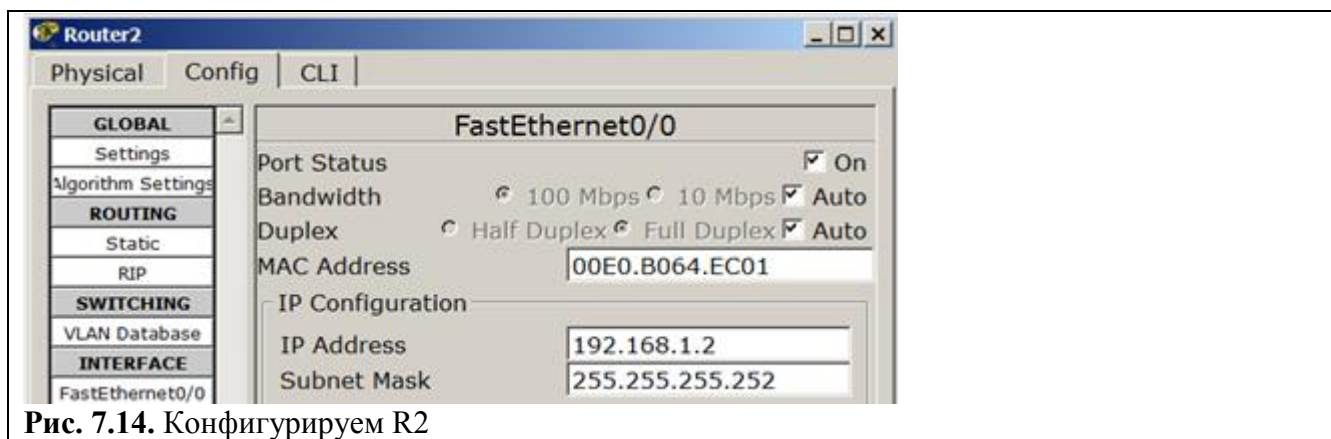
## Важно

При конфигурировании через web интерфейс обязательно установите флажок **On** (Вкл.), что эквивалентно команде **nosh**.

## Примечание

Как вариант, все параметры маршрутизатор можно настроить из командной строки на вкладке CLI следующими командами: **enable** (включаем привилегированный режим), **config terminal** (входим в режим конфигурации), **interface fastethernet0/1** (настраиваем интерфейс 100 мб Ethernet 0/1), **ip address 192.168.1.1 255.255.255.252** (прописываем ip адрес интерфейса и маску сети маршрутизатора), **no shutdown** (включаем интерфейс - по умолчанию все выключено), **exit** (выходим из режима конфигурирования интерфейса), **end** (закончили редактирование), **write** (сохранили конфигурацию).

Аналогично настраиваем Router2 исходя из постановки задачи о том, что сеть между маршрутизаторами 192.168.1.0 с маской 255.255.255.252. Порту Fa0/0 присвоим IP адрес 192.168.1.2 ( рис. 7.14).



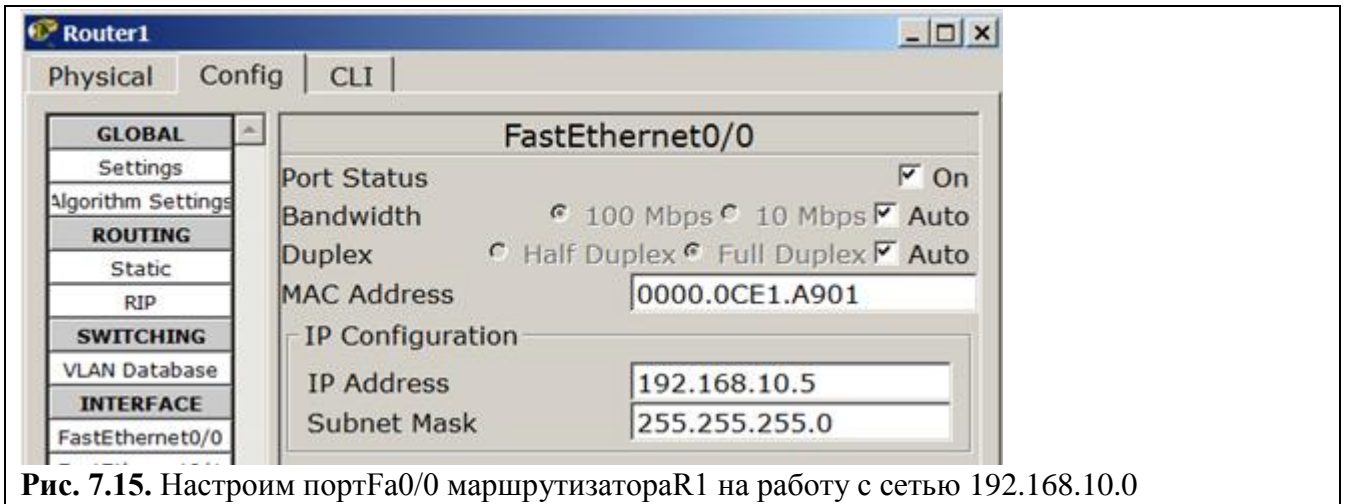
## Примечание

При конфигурировании роутера из командной строки можно использовать сокращенную форму записи команд: **en** (включаем расширенный режим). **conf t** (входим в режим конфигурации). **int fa0/0** (настраиваем интерфейс 100 мб. Ethernet 0/0). **Ip addr 192.168.1.2 255.255.255.252** (прописываем ip адрес интерфейса и маску сети). **No shut** (включаем интерфейс - по умолчанию он выключен). **exit** (выходим из режима конфигурирования интерфейса). **end** (заканчиваем редактирование). **wr** (сохраняем конфигурацию).

В итоге после настройки маршрутизаторов на портах загорятся зеленые маркеры, то есть, связь между ними есть. Сеть между маршрутизаторами работает, но маршрутизации пока нет, то есть, из одной сети в другую попасть нельзя.

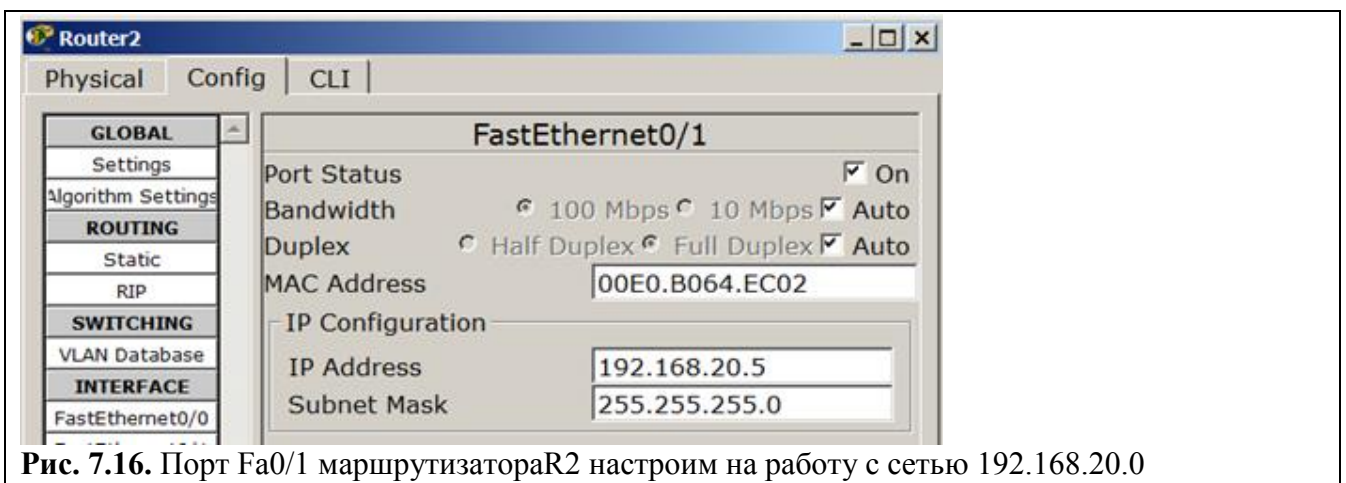
## Настройка связи маршрутизаторов с подсетями (настройка шлюзов)

Настроим порт Fa0/0 маршрутизатора R1 на работу с сетью 192.168.10.0 ( рис. 7.15).



**Рис. 7.15.** Настроим порт Fa0/0 маршрутизатора R1 на работу с сетью 192.168.10.0

Аналогично порт Fa0/1 маршрутизатора R2 настроим на работу с сетью 192.168.20.0 ( рис. 7.16).

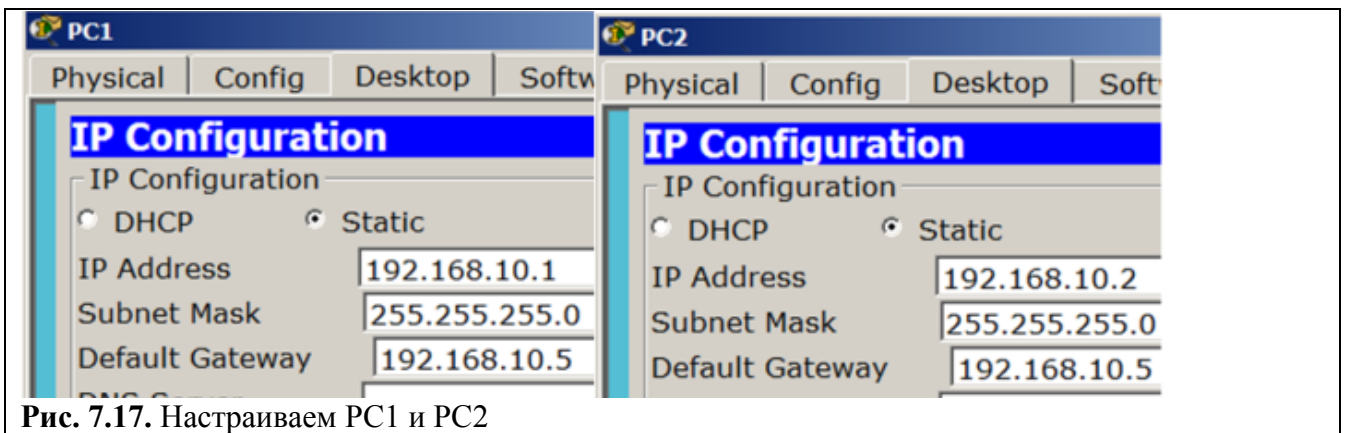


**Рис. 7.16.** Порт Fa0/1 маршрутизатора R2 настроим на работу с сетью 192.168.20.0

Как теперь видно по маркерам – сеть поднялась (Up), то есть все индикаторы горят зеленым цветом.

### Настройка PC1 и PC2

Продолжим работу и настроим компьютеры в сети 192.168.10.0, то есть, нужно задать IP компьютеров, маску сети и основной шлюз. По исходным условиям задачи у нас слева пара компьютеров в сети 192.168.10.0 с маской 255.255.255.0 ( рис. 7.17).



**Рис. 7.17.** Настраиваем PC1 и PC2

**Основной шлюз (Default Gateway)** – это адрес, куда компьютер отправляет пакет, если не знает, куда его отправить. Например, при попытке узла Б отправить данные узлу А. в

отсутствие конкретного адреса к узлу А, узел Б направляет трафик TCP/IP, предназначенный для узла А, своему основному шлюзу.

### Настройка сервера и PC3

Далее нужно настроить PC3 и сервер в сети 192.168.20.0 ( рис. 7.18 и рис. 7.19).

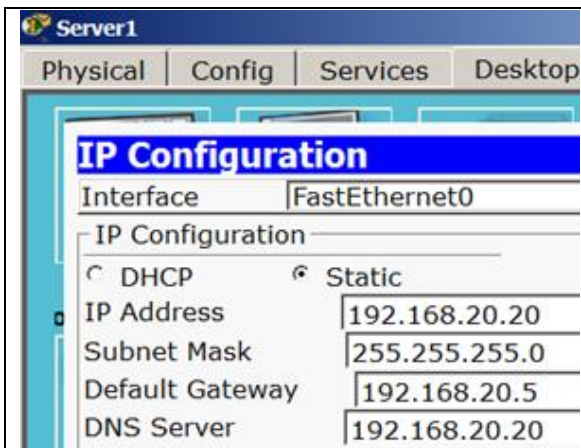


Рис. 7.18. Настройка сервера

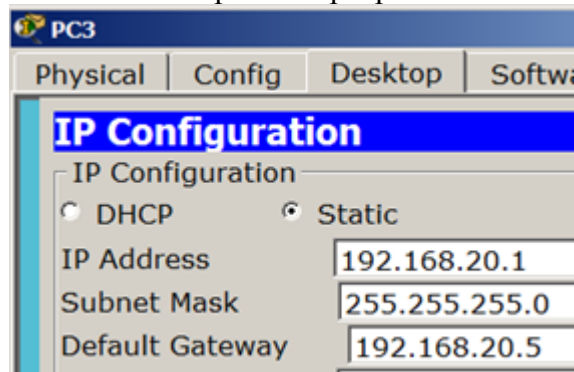


Рис. 7.19. Настраиваем PC2

### Настройка маршрутизации на маршрутизаторах (маршрута по умолчанию)

Можете пропинговать сети и убедиться в том, что ситуация такая: запросы из сети ...10.0 в сеть...20.0 проходят, а ответов – нет. Поэтому надо прописать на маршрутизаторах маршруты по умолчанию. Вспомним, что порту Fa0/1 мы присвоили IP адрес 192.168.1.1, а порту Fa0/0 – адрес 192.168.1.2. Поэтому на маршрутизаторе R1 для порта Fa0/1 с IP адресом 192.168.1.1 следует выполнить такие команды ( рис. 7.20).

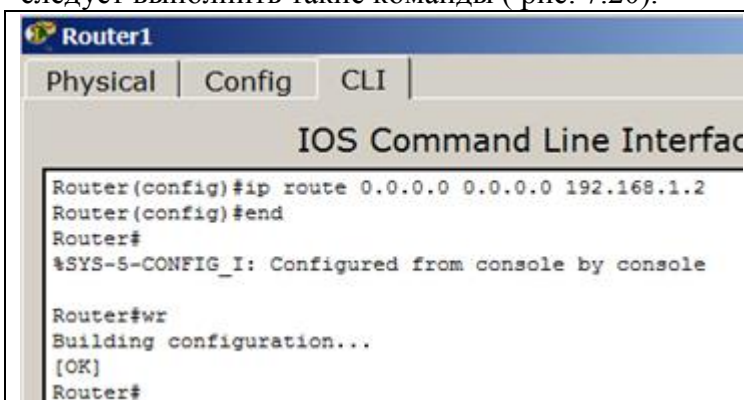


Рис. 7.20. Прописываем маршрут по умолчанию на R1

### Примечание

Запись означает, что все запросы, для которых не прописаны маршруты, R1 посылает на 192.168.1.2, то есть, на R2.

Для R2 поступаем аналогично ( рис. 7.21).

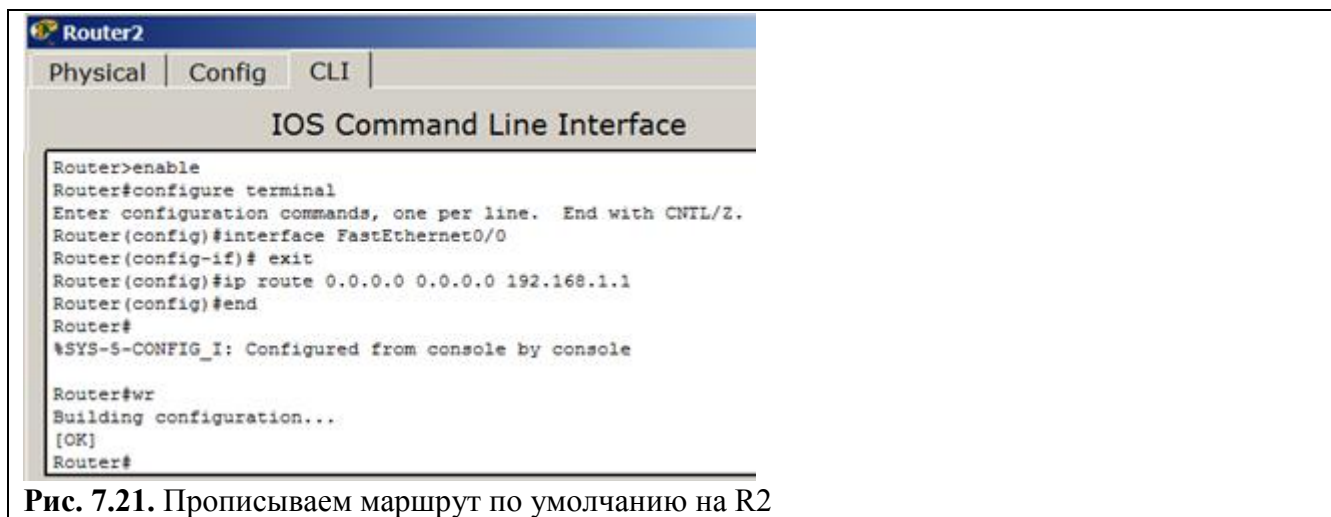


Рис. 7.21. Прописываем маршрут по умолчанию на R2

### Примечание

Запись означает, что все запросы, для которых не прописаны маршруты, R2 отправляет на 192.168.1.1, то есть, на R1.

### Проверяем работу сети

После настройки роутеров можно протестировать сеть, для этого нужно пропинговать компьютерами из одной сети — компьютеры из другой сети ( рис. 7.22).

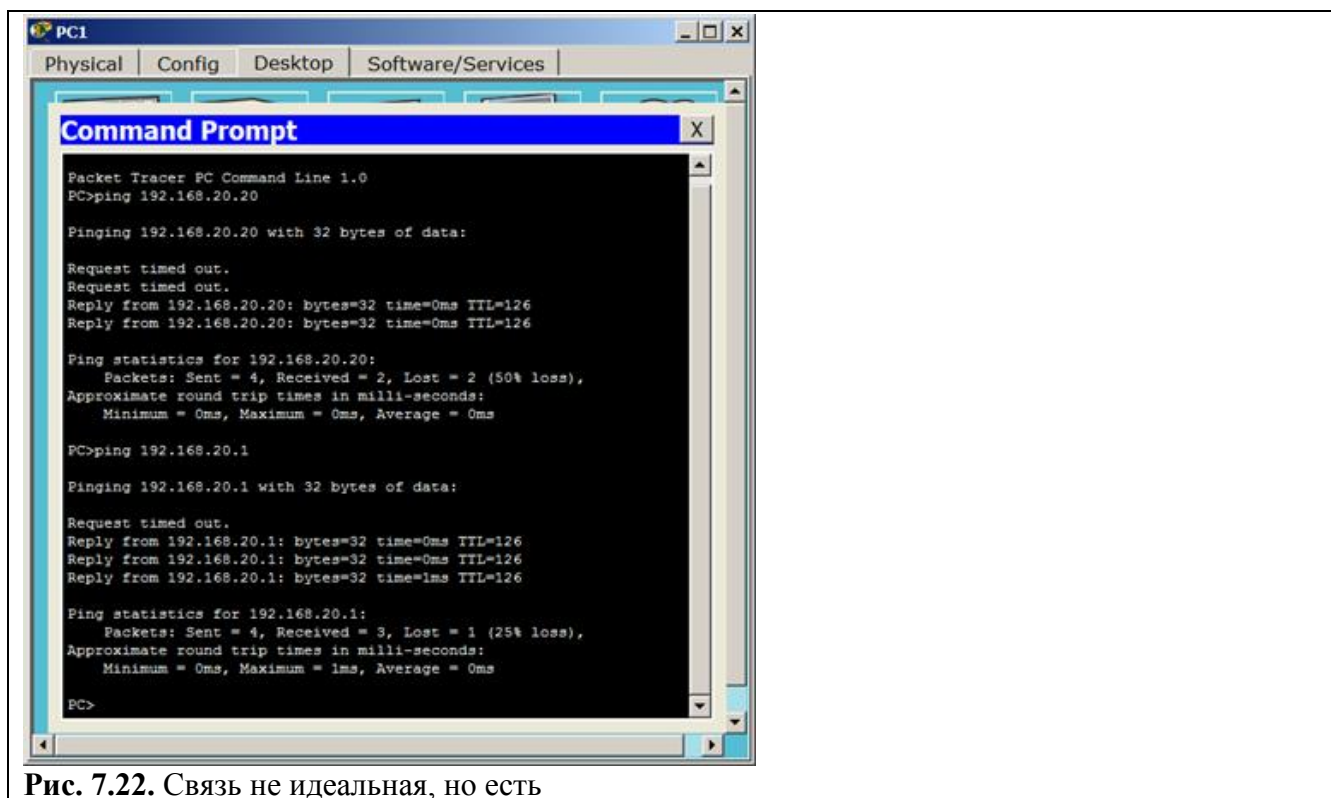


Рис. 7.22. Связь не идеальная, но есть

Чтобы убедиться наверняка, давайте посмотрим, как идут пакеты по узлам сети и для этого воспользуемся командой **tracert 192.168.20.20** ( рис. 7.23).

## Примечание

Tracert — команда, предназначенная для определения маршрутов следования данных в сетях TCP/IP.

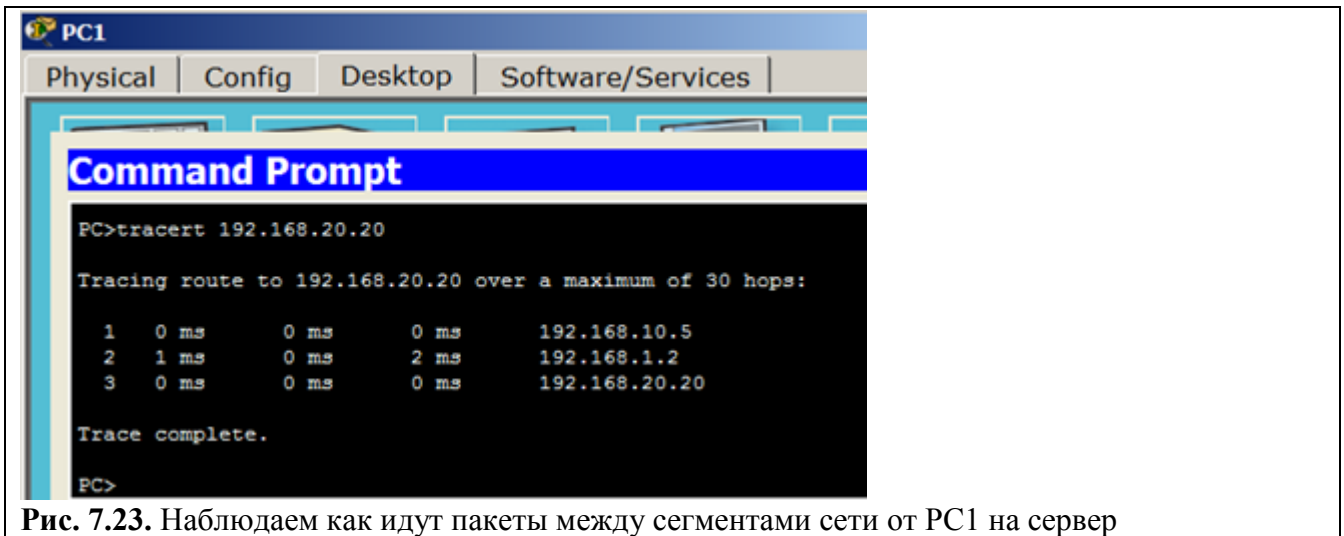


Рис. 7.23. Наблюдаем как идут пакеты между сегментами сети от PC1 на сервер

Как видно из скриншота пакеты сначала уходят на адрес 192.168.10.5 (R1 – порт Fa0/0), далее на адрес 192.168.1.2 (R2 – порт Fa0/0), а дальше приходит на сервер 192.168.20.20 — все верно!

## Примечание

Web страниц на сервере мы не создавали, но они там существуют изначально, по умолчанию. Запустите Web Browser и убедитесь в этом самостоятельно ( рис. 7.24).

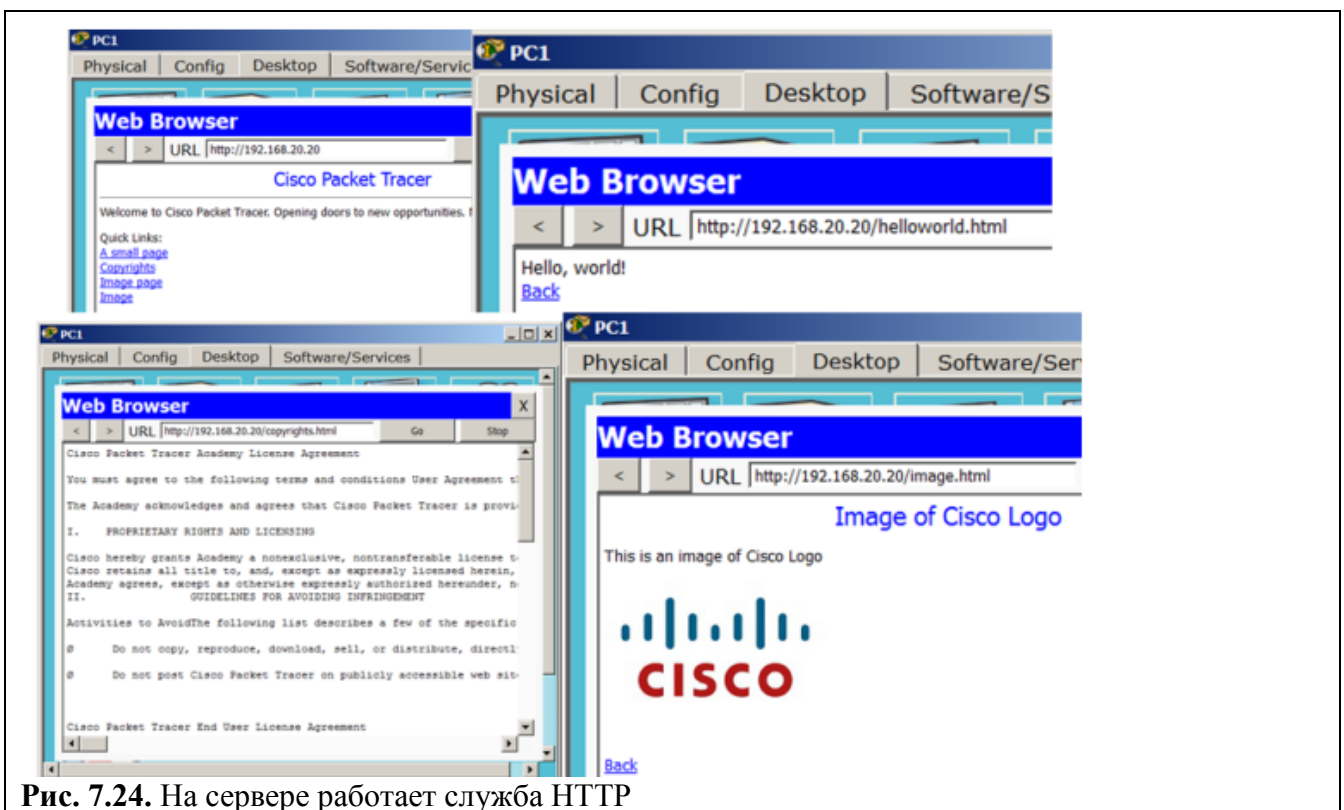
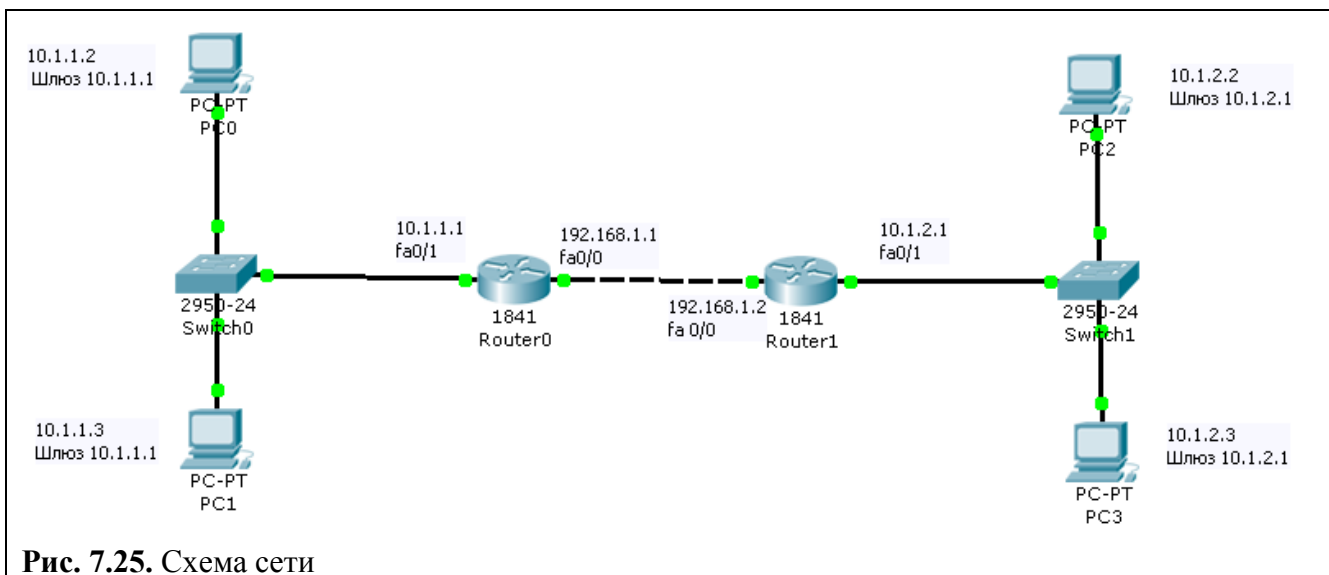


Рис. 7.24. На сервере работает служба HTTP

## Задание 3. Локальная сеть на двух маршрутизаторах

Схема сети для настройки статической маршрутизации приведена на рис. 7.25.





**Рис. 7.25.** Схема сети

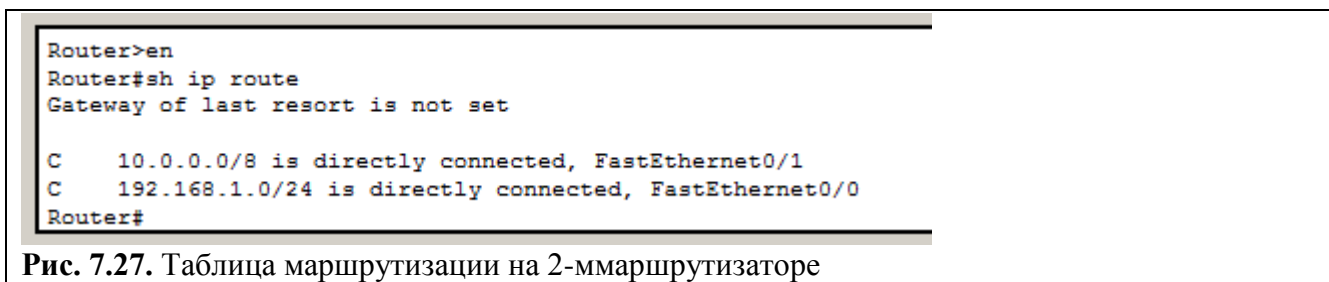
Предварительно настройте устройства по схеме сети (рис 7.25)

Только затем приступайте дальше

Если сейчас командой `show ip route` посмотреть таблицу маршрутизации на R0 и R1, то увидим следующее (рис. 7.26 и рис. 7.27).



**Рис. 7.26.** Таблица маршрутизации на 1-м маршрутизаторе



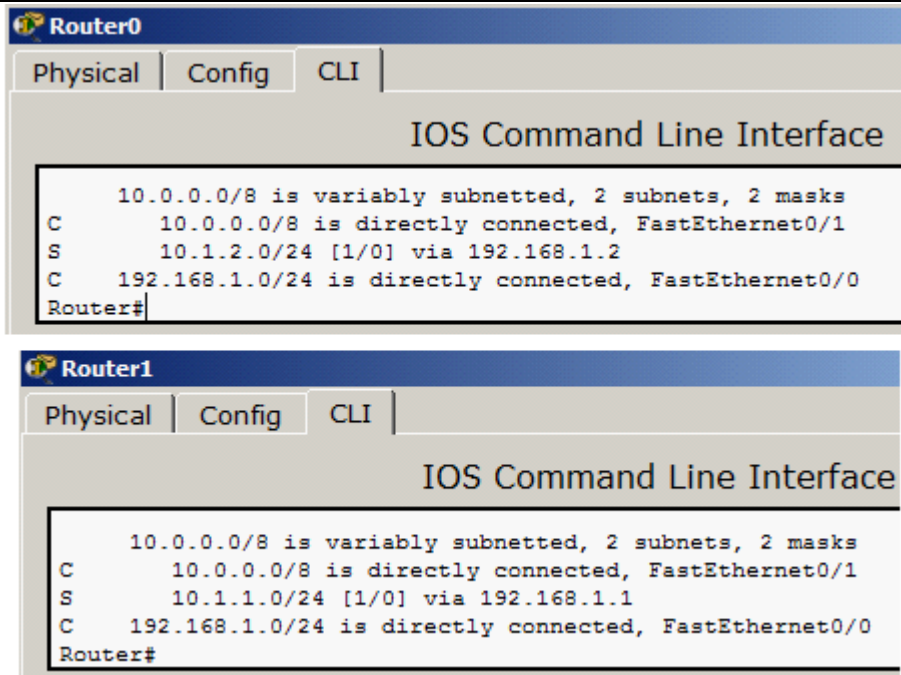
**Рис. 7.27.** Таблица маршрутизации на 2-м маршрутизаторе

Мы видим, что в данный момент в нашей таблице есть только сети, подключенные напрямую. R0 не знает сеть 10.1.2.0, а R1 не знает сеть 10.1.1.0. Поэтому, чтобы настроить маршрутизацию, следует добавить эти маршруты в таблицы маршрутизаторов:

R0 (config)#ip route 10.1.2.0 255.255.255.0 192.168.1.2

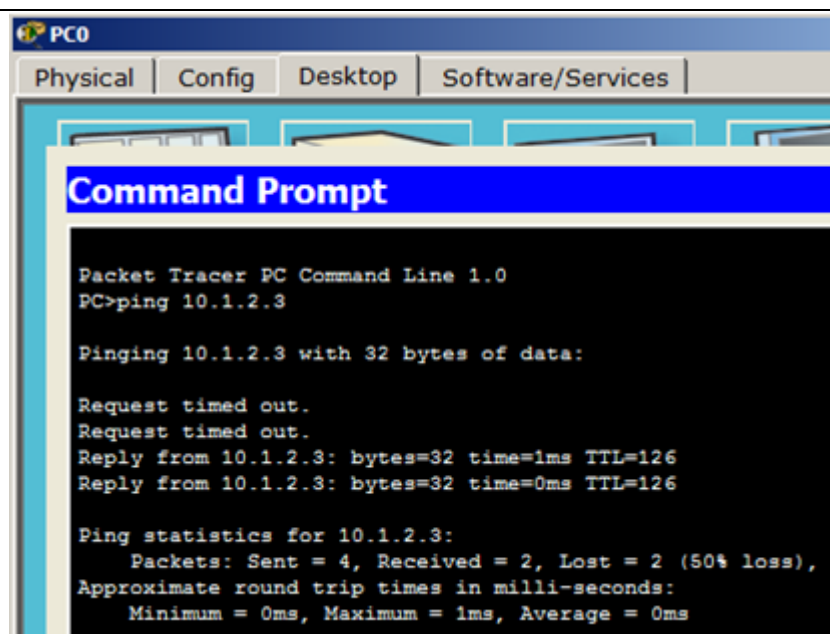
R1 (config)#ip route 10.1.1.0 255.255.255.0 192.168.1.1

Теперь снова выведем таблицы маршрутизации наших устройств (рис. 7.28).



**Рис. 7.28.** Маршрутизация настроена

Теперь 1-й маршрутизатор знает, что пакеты, направляемые в подсеть 10.1.2.0 можно переслать маршрутизатору с ip адресом 192.168.1.2, а 2-й маршрутизатор знает, что пакеты, направляемые в подсеть 10.1.1.0 можно переслать маршрутизатору с ip адресом 192.168.1.1. Проверяем связь ПК из разных сетей ( рис. 7.29).



**Рис. 7.29.** Статическая маршрутизация настроена – PC0 может общаться с PC3

Форма представления и содержание отчета

Отчет в формате DOC или PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.

*Содержание отчета*

1. Заголовок согласно приложению.

2. Цель работы.

3. Ход работы:

- c. Данный раздел состоит из последовательного описания значимых выполняемых шагов (с указанием их сути) Пояснения работы команды ping и **содержимго протоколов**.
- d. копий экранов (должна быть видна набранная команда и реакция системы, если она есть).

4. По каждому пункту практической работы должны быть приведены **выводы**.

### **Практическая работа №9. Динамическая маршрутизация на протоколах RIP, EIGRP и OSPF**

**Целью** данной практической работы является получение навыков по настройке динамической маршрутизации внутри различных сетей

#### **Задачи:**

- Ознакомиться с динамической маршрутизацией и ее протоколами RIP, EIGRP, OSPF
- Выполнить задания по настройке сетей с динамической маршрутизацией с различными протоколами (задания 1-4)
- Составить отчет с последовательным выполнением заданий и скриншотами (задания 1-4)

#### **Теоретическая часть**

**Маршрутизация** - процесс определения в сети наилучшего пути, по которому пакет может достигнуть адресата. *Динамическая маршрутизация* может быть осуществлена с использованием одного и более протоколов (*RIP v2, OSPF* и др.).

**Динамическая маршрутизация** — вид маршрутизации, при котором таблица маршрутизации заполняется и обновляется автоматически при помощи одного или нескольких протоколов маршрутизации (RIP, OSPF, EIGRP, BGP).

Каждый *протокол маршрутизации* использует свою систему оценки маршрутов (*метрику*). *Маршрут* к сетям назначения строится на основе таких критериев как

- количество ретрансляционных переходов
- пропускная способность канала связи
- задержки передачи данных
- и др.

Маршрутизаторы обмениваются друг с другом информацией о маршрутах с помощью служебных пакетов по протоколу *UDP*. Такой обмен информации увеличивает наличие дополнительного трафика в сети и нагрузку на эту *сеть*. Возможна также ситуация, при которой таблицы маршрутизации на роутерах не успевают согласоваться между собой, что может повлечь появление ошибочных маршрутов и потерю данных.

Протоколы маршрутизации делятся на три типа:



- Дистанционно векторные протоколы (RIP)
- Протоколы с отслеживанием состояния каналов (OSPF)
- Смешанные протоколы (EIGRP)
- И др.

## Протокол RIP

RIP — протокол дистанционно-векторной маршрутизации, использующий для нахождения оптимального пути алгоритм Беллмана-Форда. Алгоритм маршрутизации RIP- один из самых простых протоколов маршрутизации. Каждые 30 секунд он передает в сеть свою таблицу маршрутизации. Основное отличие протоколов в том, что RIPv2 (в отличие от RIPv1) может работать по мультикасту, то есть, рассылаясь на мультикаст адрес. Максимальное количество "хопов" (шагов до места назначения), разрешенное в RIP1, равно 15 (метрика 15). Ограничение в 15 хопов не дает применять RIP в больших сетях, поэтому протокол наиболее распространен в небольших компьютерных сетях. Вторая версия протокола — протокол RIP2 была разработана в 1994 году и является улучшенной версией первого. В этом протоколе повышена безопасность за счет введения дополнительной маршрутной информации. Принцип дистанционно-векторного протокола: каждый маршрутизатор, использующий протокол RIP периодически широкоэвещательно рассылает своим соседям специальный пакет-вектор, содержащий расстояния (измеряются в метрике) от данного маршрутизатора до всех известных ему сетей. Маршрутизатор получивший такой вектор, наращивает компоненты вектора на величину расстояния от себя до данного соседа и дополняет вектор информацией об известных непосредственно ему самому сетях или сетях, о которых ему сообщили другие маршрутизаторы. Дополненный вектор маршрутизатор рассылает всем своим соседям. Маршрутизатор выбирает из нескольких альтернативных маршрутов маршрут с наименьшим значением метрики, а маршрутизатор, передавший информацию о таком маршруте помечается как следующий (nexthop). Протокол непригоден для работы в больших сетях, так как засоряет сеть интенсивным трафиком, а узлы сети оперируют только векторами-расстояний, не имея точной информации о состоянии каналов и топологии сети. Сегодня даже в небольших сетях протокол вытесняется превосходящими его по возможностям протоколами EIGRP и OSPF.

### Задание 1. Настройка протокола RIP версии 2 для сети из шести устройств

Настройте маршрутизацию на схеме, представленной на рис. 8.1.

Только затем приступайте дальше

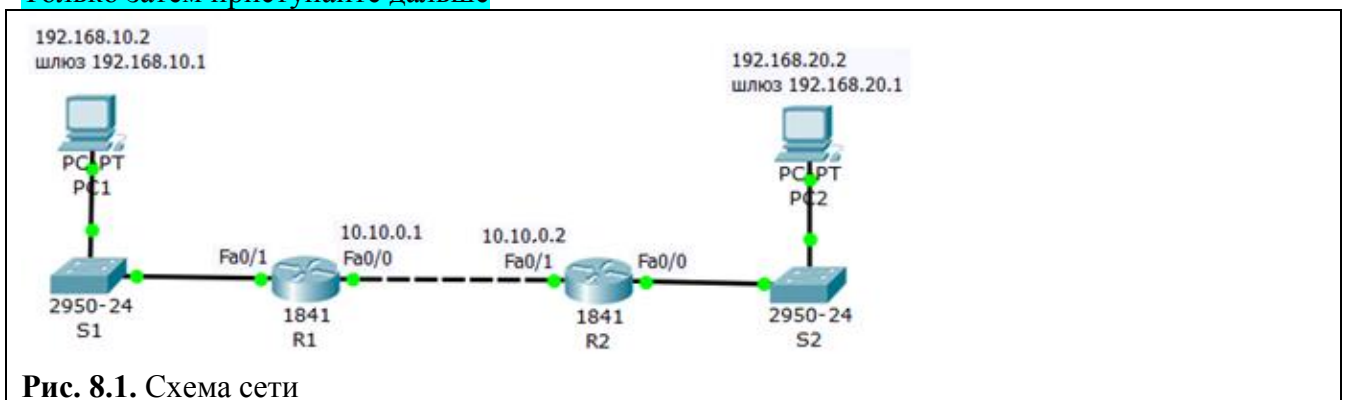


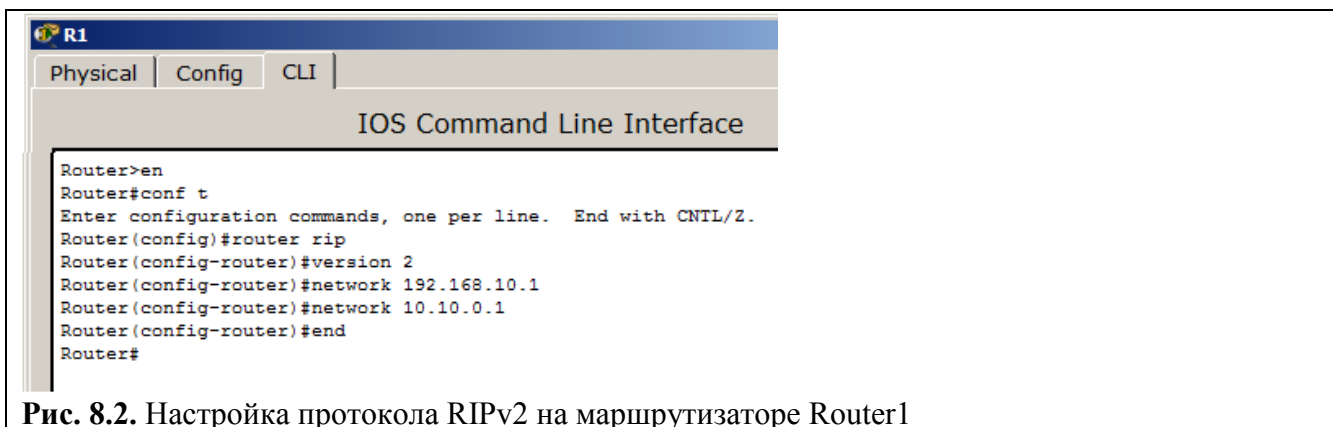
Рис. 8.1. Схема сети

#### Примечание

При настройке сети не забывайте включать порты.

#### Настройка протокола RIP на маршрутизаторе R1

Войдите в конфигурации в консоль роутера и выполните следующие настройки (рис. 8.2).



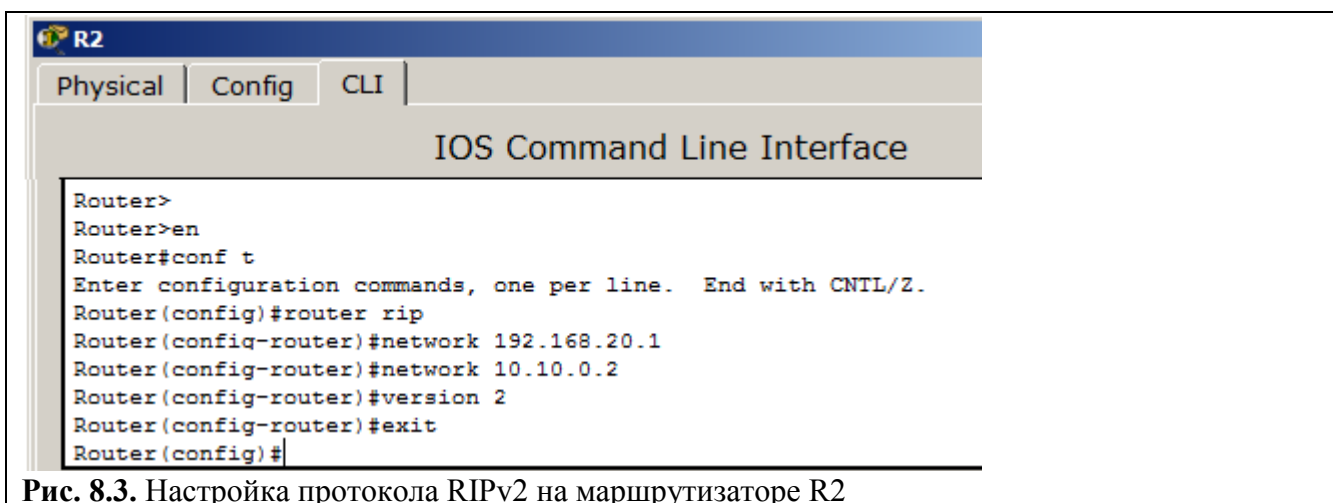
**Рис. 8.2.** Настройка протокола RIPv2 на маршрутизаторе Router1

### Примечание

**Router(config)#router rip** (Вход в режим конфигурирования протокола RIP). **Router(config-router)#network 192.168.10.1** (Подключение клиентской сети к роутеру со стороны коммутатора S1). **Router(config-router)#network 192.168.20.1** (Подключение второй сети, то есть сети между роутерами). **Router(config-router)#version 2** (Задание использования второй версии протокол RIP).

### Настройка протокола RIP на маршрутизаторе R2

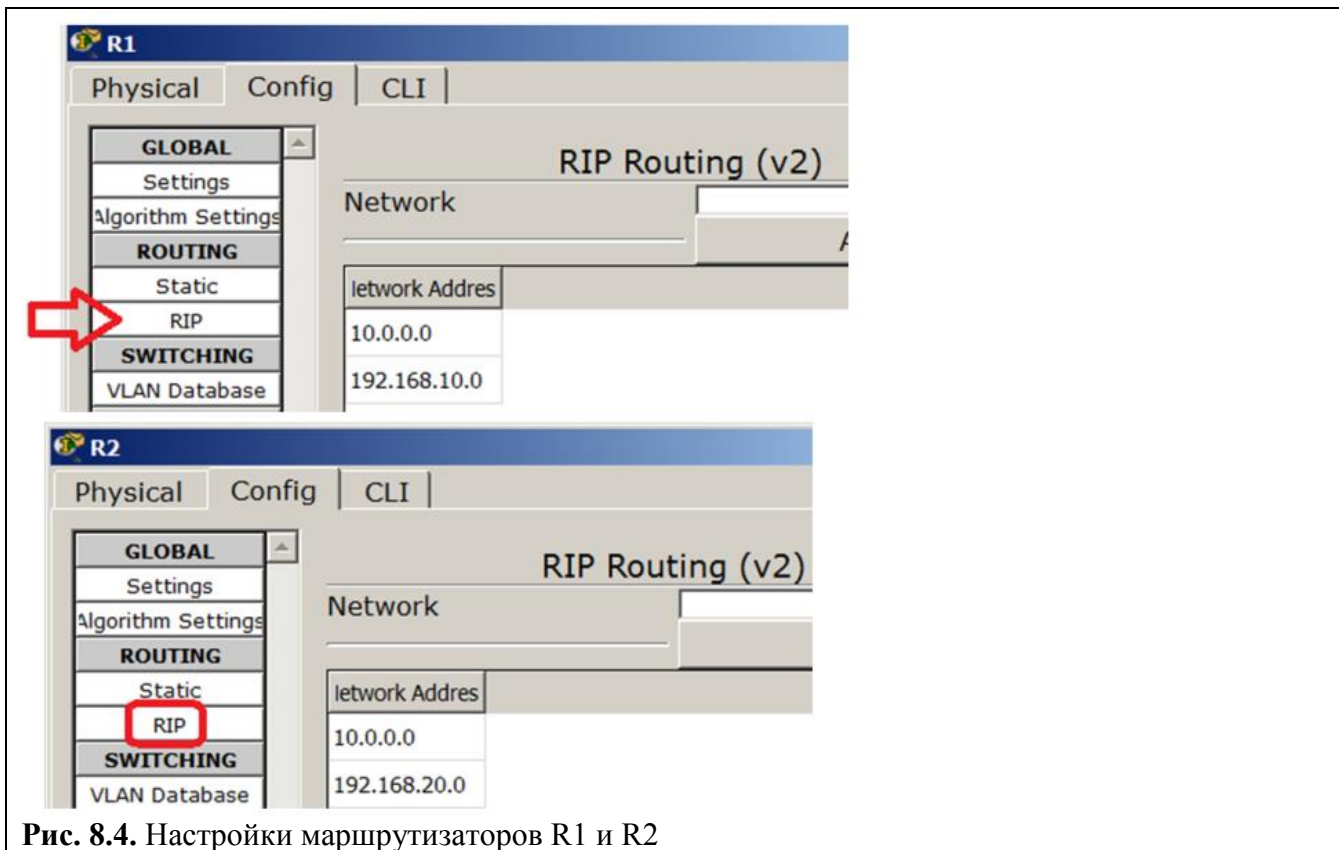
Войдите в конфигурации роутера 2 и выполните следующие настройки ( рис. 8.3).



**Рис. 8.3.** Настройка протокола RIPv2 на маршрутизаторе R2

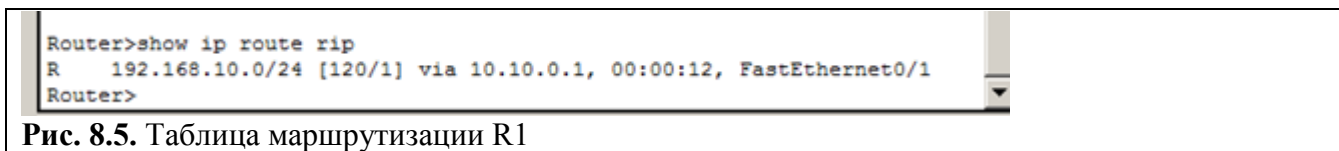
### Проверяем настройки коммутаторов и протокола RIP

Давайте посмотрим настройки протокола RIPv2 на маршрутизаторах R1 и R2 ( рис. 8.4).



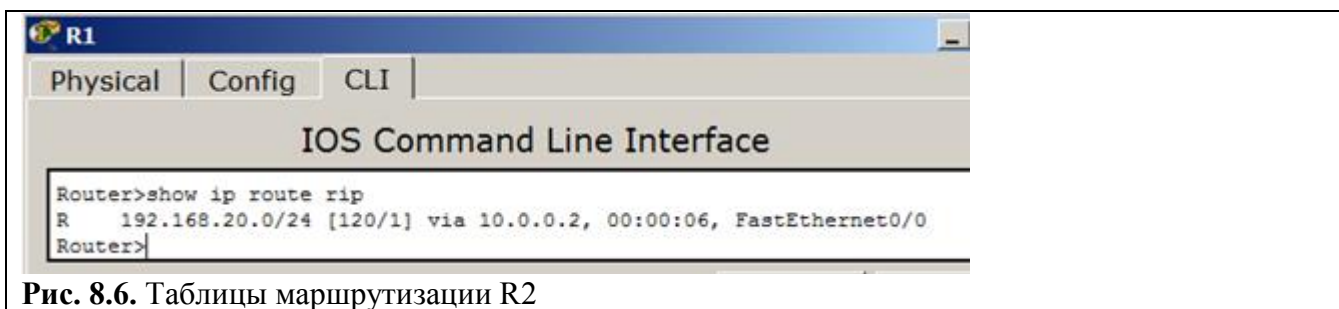
**Рис. 8.4.** Настройки маршрутизаторов R1 и R2

Чтобы убедиться в том, что маршрутизаторы действительно правильно сконфигурированы и работают корректно, просмотрите таблицу RIP роутеров, используя команду: **Router#show ip route rip** ( рис. 8.5 и рис. 8.6).



**Рис. 8.5.** Таблица маршрутизации R1

Данная таблица показывает, что к сети 192.168.10.0 есть только один маршрут: через R1(сеть 10.10.0.1).



**Рис. 8.6.** Таблицы маршрутизации R2

Данная таблица показывает, что к сети 192.168.20.0 есть только один маршрут: через R2 (сеть 10.10.0.2).

### Проверка связи между PC1 и PC2

Проверим, что маршрутизация производится верно ( рис. 8.7).

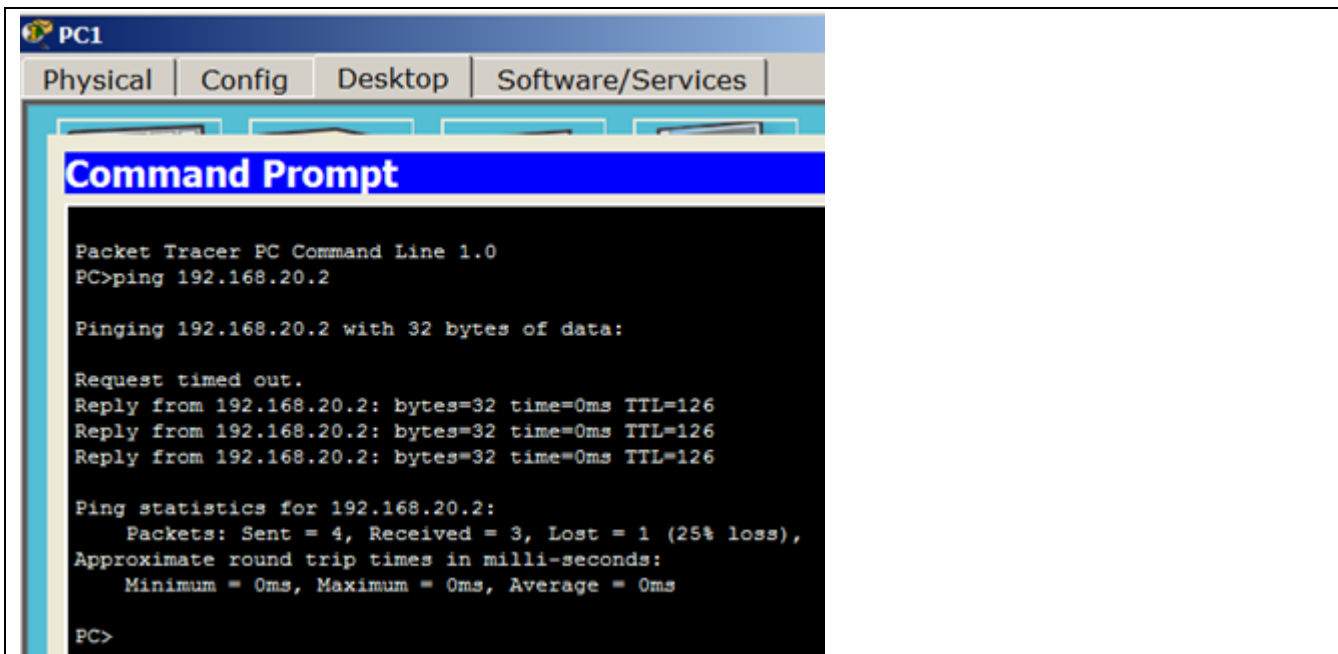


Рис. 8.7. Пинг с PC1 на PC2

## Задание 2. Конфигурирование протокола RIP версии 2 для сети из четырех устройств

На рис. 8.8 представлена *сеть*, на примере которой мы сконфигурируем *протокол маршрутизации RIP v2*.



Рис. 8.8. Сеть для конфигурации протоколов маршрутизации

Настройте маршрутизацию на схеме, представленной на рис. 8.8. Только затем приступайте дальше

Сначала сконфигурируем R1 (рис. 8.9).

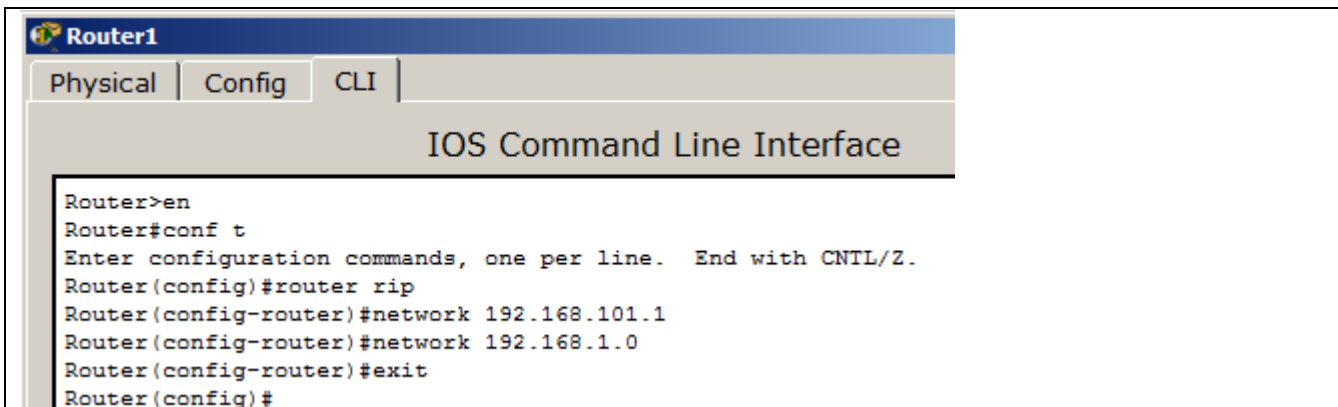


Рис. 8.9. Настройка RIP на R1

Смотрим результат на вкладке **Config** (рис. 8.10).

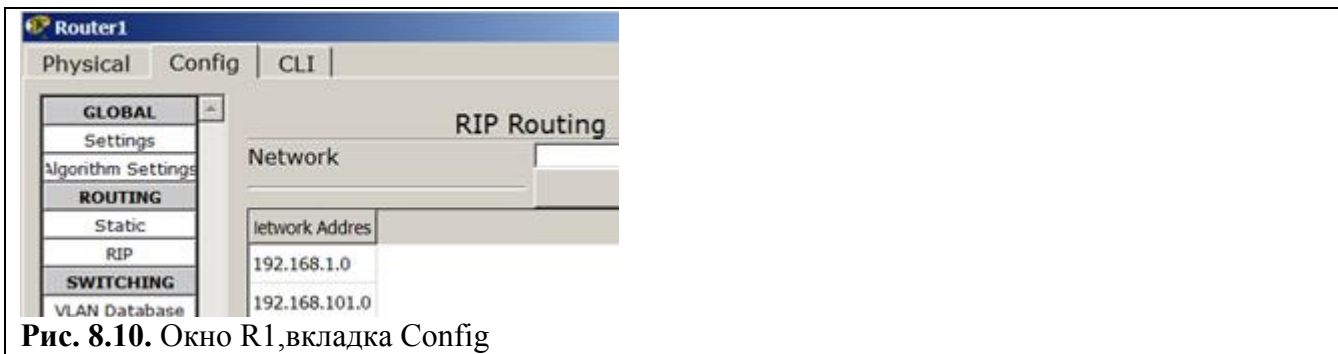


Рис. 8.10. Окно R1, вкладка Config

Конфигурируем R2 ( рис. 8.11).

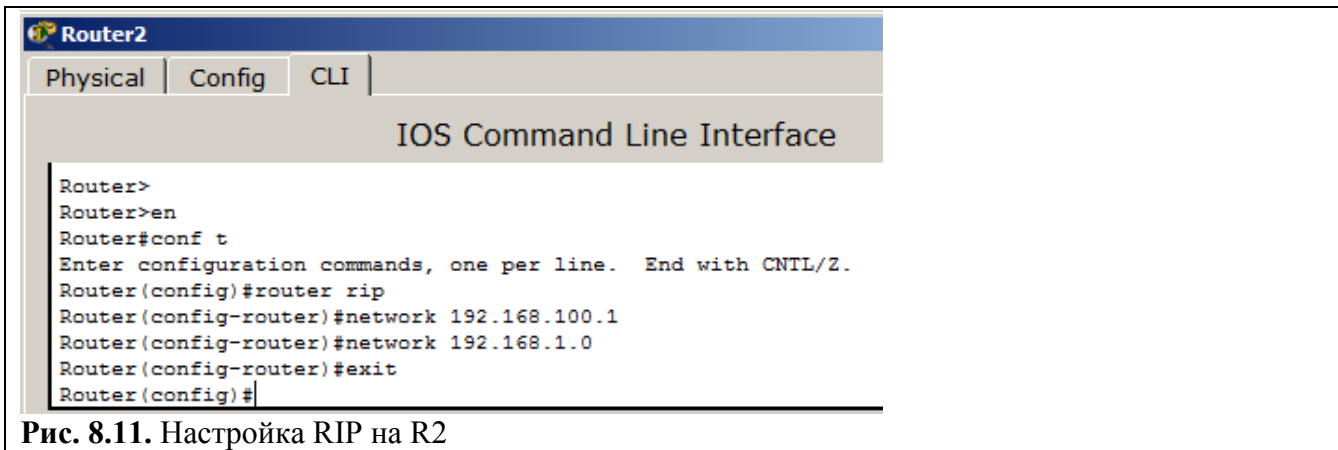


Рис. 8.11. Настройка RIP на R2

Наблюдаем результат ( рис. 8.12).

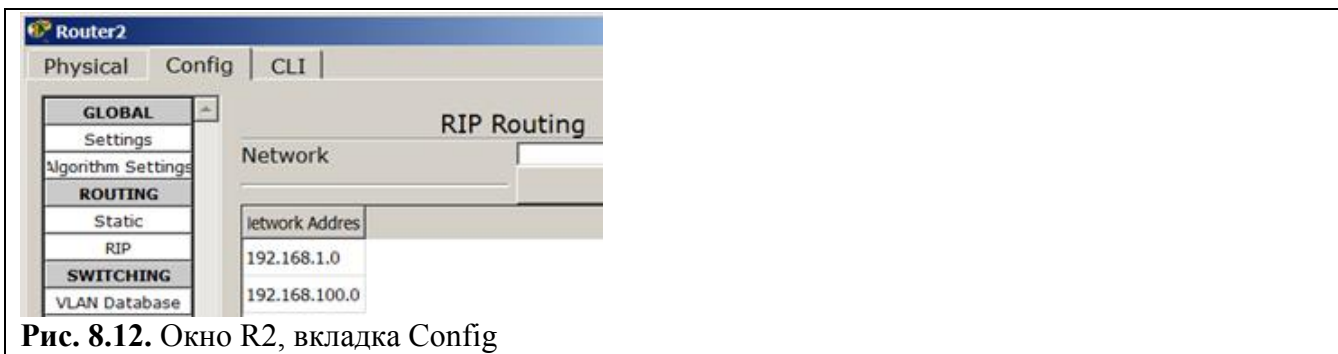


Рис. 8.12. Окно R2, вкладка Config

Проверяем доступность ПК из разных сетей ( рис. 8.13).

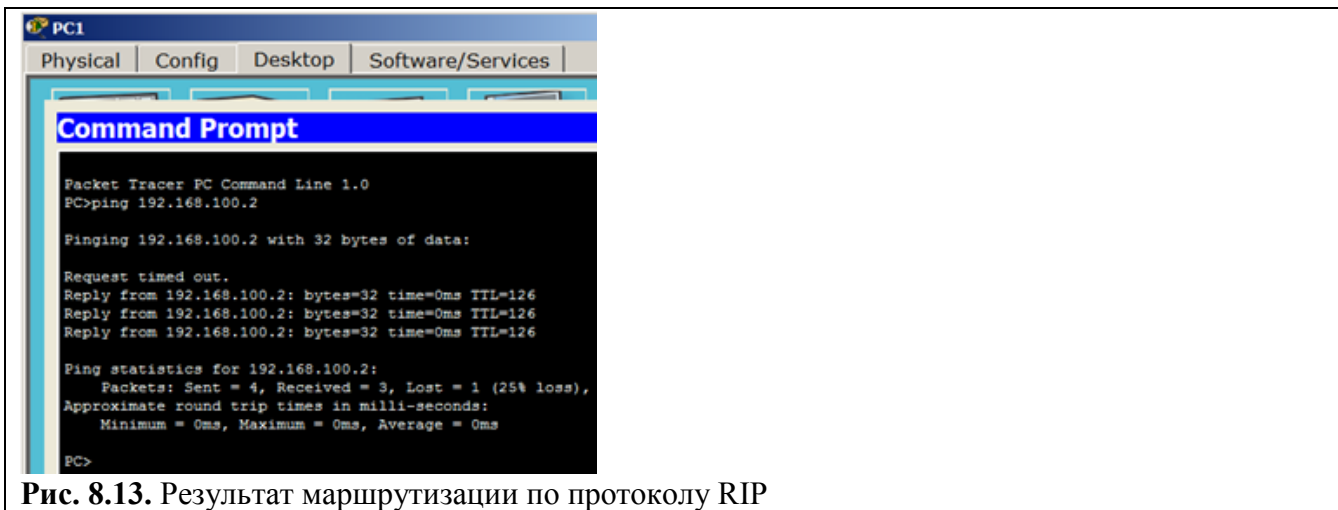


Рис. 8.13. Результат маршрутизации по протоколу RIP

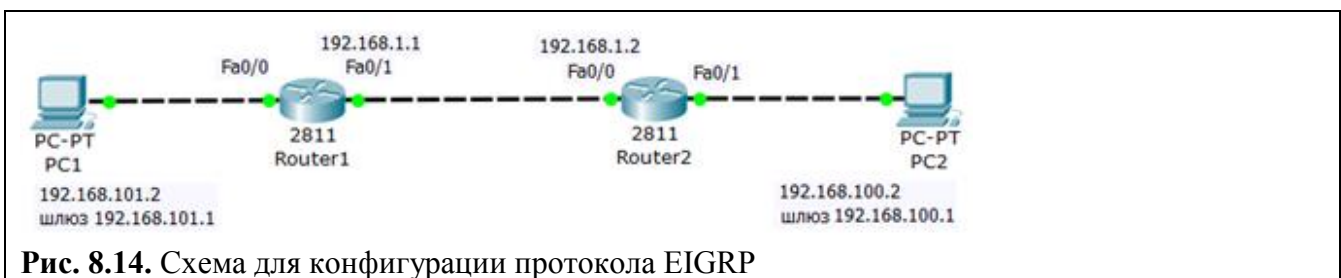
### Задание 3. Конфигурирование протокола EIGRP

#### Протокол маршрутизации EIGRP

Протокол *EIGRP* более прост в реализации и менее требователен к вычислительным ресурсам маршрутизатора, чем протокол *OSPF*. Также *EIGRP* имеет более продвинутый *алгоритм* вычисления метрики. В формуле вычисления метрики есть возможность учитывать загруженность и *надежность* интерфейсов на пути пакета. Недостатком протокола *EIGRP* является его ограниченность в его использовании только на оборудовании компании Cisco.

Схема сети изображена на рис. 8.14.

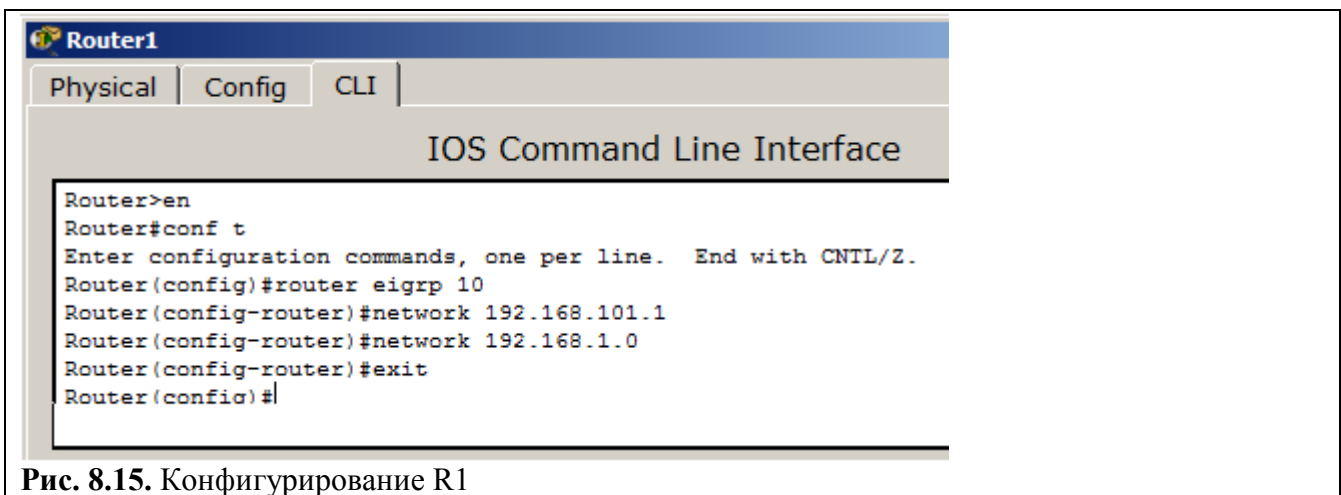
Настройте маршрутизацию на схеме, представленной на рис. 8.14  
Только затем приступайте дальше



Настройка протокола *EIGRP* очень похожа на настройку протокола *RIP*.

#### Программирование R1

Конфигурируем R1 ( рис. 8.15).



#### Программирование R2

Конфигурируем R2 ( рис. 8.16).



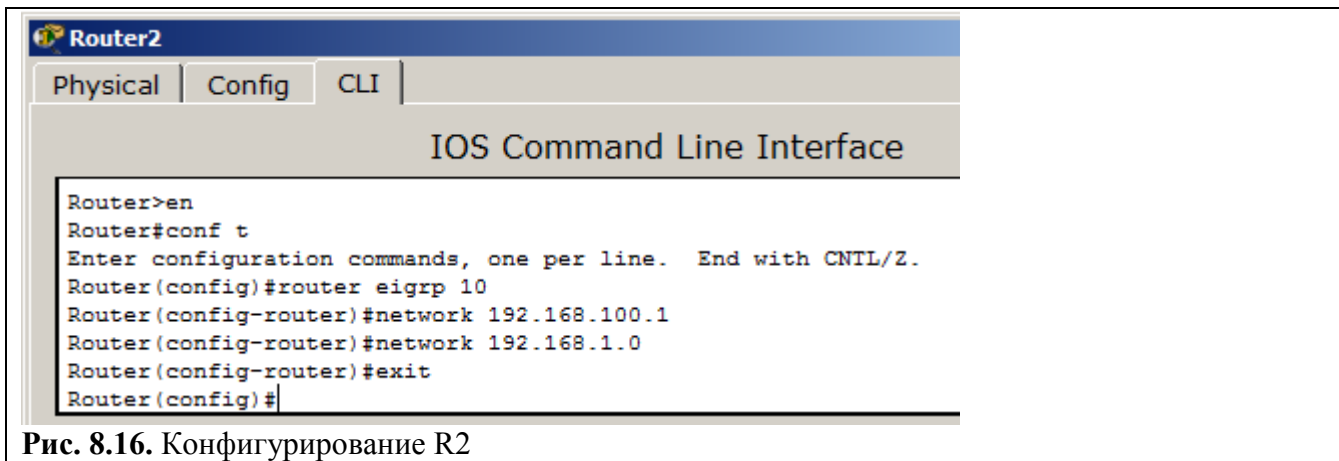


Рис. 8.16. Конфигурирование R2

### Проверка работы сети

Проверяем работу маршрутизаторов ( рис. 8.17).

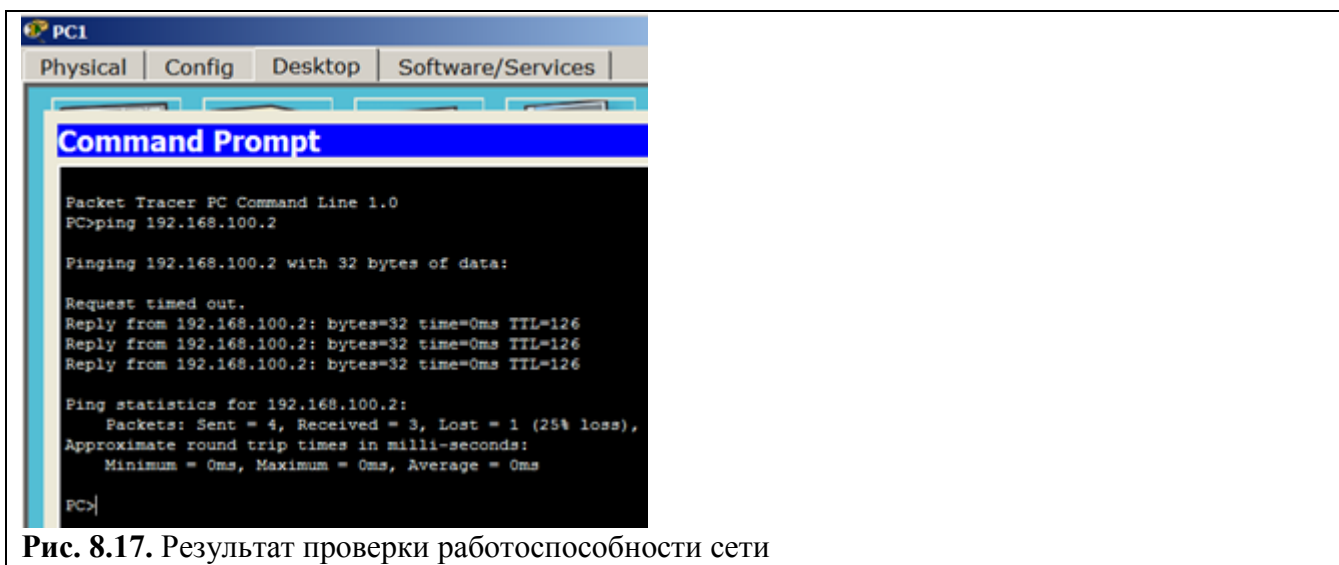


Рис. 8.17. Результат проверки работоспособности сети

## Задание 4. Конфигурирование протокола OSPF для 4-х устройств

### Протокол OSPF

Алгоритм работы протокола динамической маршрутизации *OSPF* основан на использовании всеми маршрутизаторами единой *базы данных*, описывающей, с какими сетями связан каждый *маршрутизатор*. Описывая каждую *связь*, маршрутизаторы связывают с ней метрику – *значение*, характеризующее "качество" канала связи. Это позволяет маршрутизаторам *OSPF* (в отличие от *RIP*, где все каналы равнозначны) учитывать реальную пропускную способность канала и выявлять наилучшие маршруты. Важной особенностью протокола *OSPF* является то, что используется групповая, а не широковещательная рассылка (как в *RIP*), то есть, нагрузка каналов меньше.

*OSPF (Open Shortest Path First)* — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала *link-state (LSA)*. Основан на алгоритме для поиска кратчайшего пути. Отслеживание состояния канала требует отправки объявлений о состоянии канала (*LSA*) на активные интерфейсы всех доступных маршрутизаторов зоны. В этих объявлениях содержится описание всех каналов маршрутизатора и *стоимость* каждого канала. *LSA* сообщения отправляются, только если произошли какие-либо изменения в сети, но раз в 30 минут *LSA* сообщения отправляются в принудительном порядке. Протокол

реализует *деление* автономной системы на зоны (areas). Использование зон позволяет снизить нагрузку на *сеть* и процессоры маршрутизаторов и уменьшить размер таблиц маршрутизации.

### Описание работы протокола:

Все маршрутизаторы обмениваются специальными Hello-пакетами через все интерфейсы, на которых активирован протокол *OSPF*. Таким образом, определяются маршрутизаторы-соседи, разделяющие общий *канал передачи данных*. В дальнейшем hello-пакеты посылаются с интервалом раз в 30 секунд. Маршрутизаторы пытаются перейти в состояние соседства со своими соседями. Переход в данное состояние определяется типом маршрутизаторов и типом сети, по которой происходит обмен hello-пакетами, по зонному признаку. Пара маршрутизаторов в состоянии соседства синхронизирует между собой базу данных состояния каналов. Каждый *маршрутизатор* посылает объявление о состоянии канала своим соседям, а каждый получивший такое объявление записывает информацию в базу данных состояния каналов и рассылает копию объявления другим своим соседям. При рассылке объявлений по зоне, все маршрутизаторы строят идентичную базу данных состояния каналов. Каждый *маршрутизатор* использует *алгоритм SPF* для вычисления графа (дерева кратчайшего пути) без петель. Каждый *маршрутизатор* строит собственную маршрутизацию, основываясь на построенном дереве кратчайшего пути.

### Прямая и обратная маска

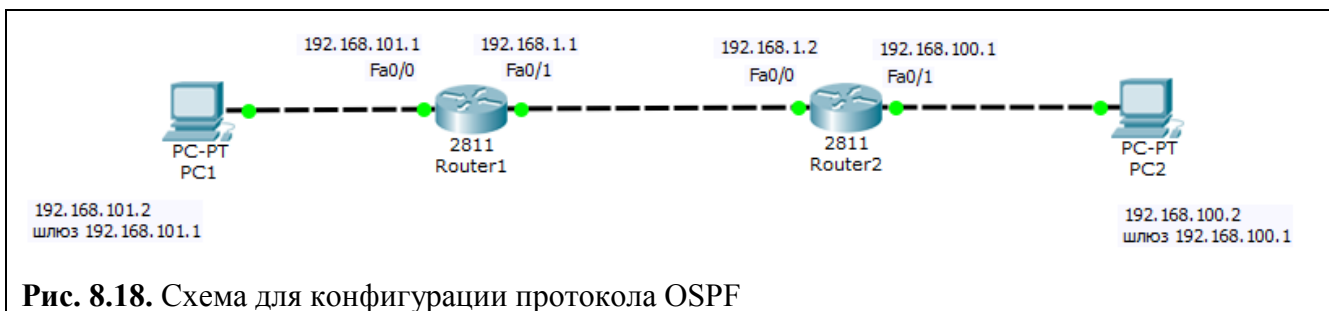
В оборудовании **Cisco** иногда приходится использовать обратную маску, то есть не привычную нам **255.255.255.0** (Subnet mask — прямая *маска*), а **0.0.0.255** (Wildcard mask — обратная *маска*). Обратная *маска* используется в листах допуска (*access list*) и при описании сетей в протоколе **OSPF**. Прямая *маска* используется во всех остальных случаях. Отличие масок заключается также в том, что прямая *маска* оперирует сетями, а обратная — хостами. С помощью обратной маски вы можете, например, выделить во всех подсетях хосты с конкретным адресом и разрешить им *доступ в Интернет*. Так, как чаще всего в локальных сетях используют адреса типа 192.168.1.0 с маской 255.255.255.0, то самая распространенная Wildcard mask (шаблонная *маска* или обратная *маска*, или инверсная *маска*) - *маска* 0.0.0.255.

**Шаблонная маска (wildcard mask)** — маска, указывающая на количество хостов сети. Является дополнением для маски подсети. Вычисляется по формуле для каждого из октетов маски подсети как 255-маска\_подсети. Например, для сети 192.168.1.0 и маской подсети 255.255.255.242 шаблонная маска будет выглядеть как 0.0.0.13. Шаблонная маска используется в настройке некоторых протоколов маршрутизации, а также является удобным параметром ограничений в списках доступа.

### Расчёт Wildcard mask

Существует *связь*, между обратной и *прямой* маской: в сумме эти маски по каждому разряду должны составлять 255. Пусть наша *сеть* 192.168.32.0 /28. Рассчитает wildcard mask: *префикс* /28 это 255.255.255.240 или 11111111.11111111.11111111.11110000. Для wildcard mask нам нужны только нули, то есть, 11110000 переводим в десятичное число и считаем: 128/64/32/16/8/4/2/1 это будет 8+4+2+1=15, т.е. наша wildcard mask будет равна 0.0.0.15.



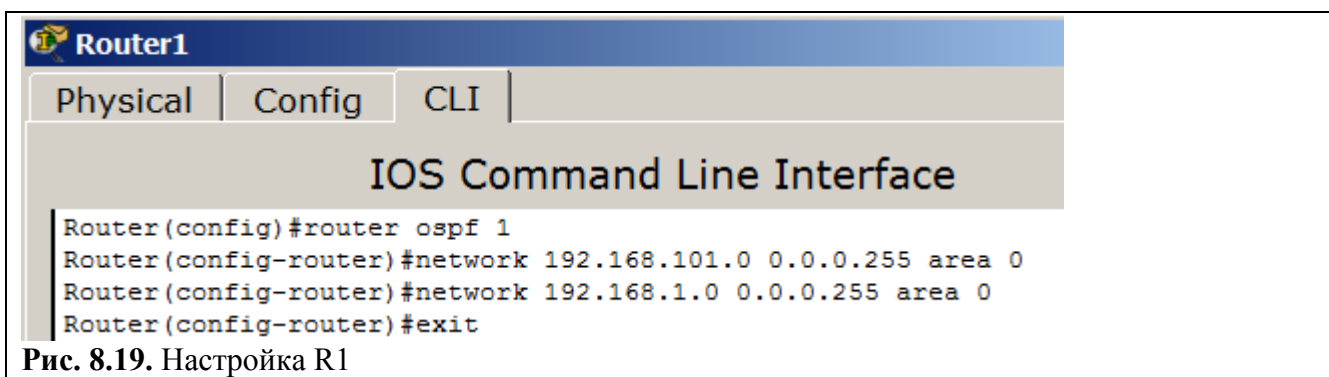


**Рис. 8.18.** Схема для конфигурации протокола OSPF

Настройте маршрутизацию на схеме, представленной на рис. 8.18  
Только потом приступайте дальше

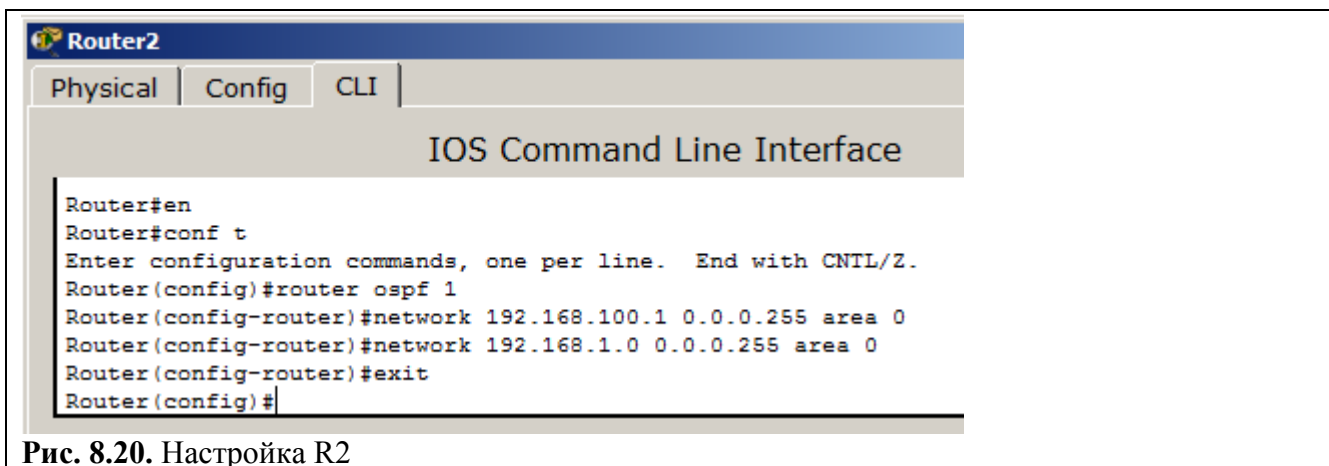
## Настройка роутеров

Выполним конфигурирование R1 ( рис. 8.19).



**Рис. 8.19.** Настройка R1

Теперь выполним настройки R2 ( рис. 8.20).



**Рис. 8.20.** Настройка R2

### Совет

Если вам потребуется в СРТ сбросить настройки роутера, то следует выключить его тумблер питания, а затем снова включить.

### Проверка результата

Для проверки маршрутизации пропиnguем ПК из разных сетей ( рис. 8.21).

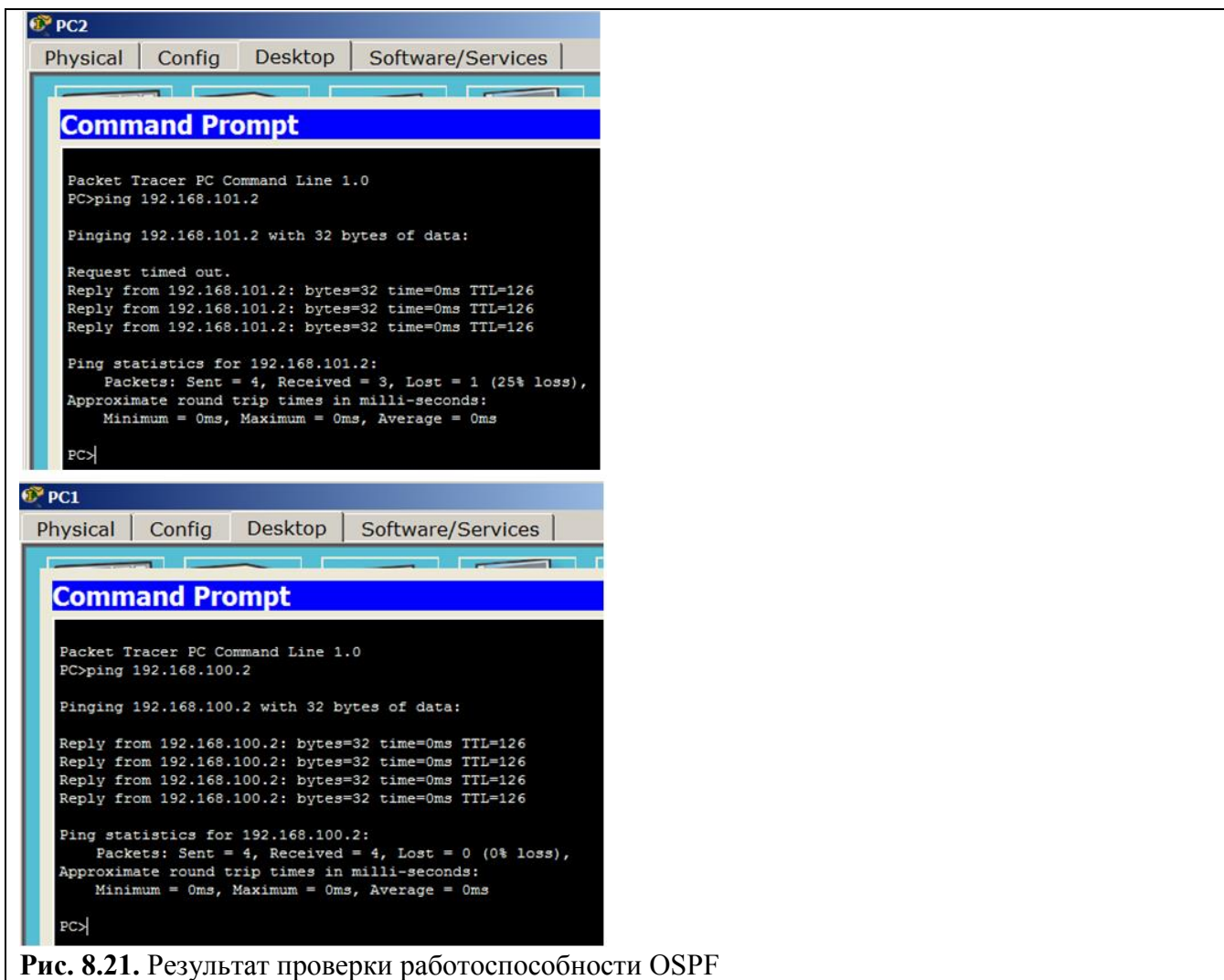


Рис. 8.21. Результат проверки работоспособности OSPF

### Форма представления и содержание отчета

Отчет в формате DOC или PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.

### *Содержание отчета*

1. Заголовок согласно приложению.
2. Цель работы.
3. Ход работы:
  - e. Данный раздел состоит из последовательного описания значимых выполняемых шагов (с указанием их сути) Пояснения работы команды ping и **содержимго протоколов**.
  - f. копий экранов (должна быть видна набранная команда и реакция системы, если она есть).
4. По каждому пункту практической работы должны быть приведены **выводы**.

## Практическая работа №10. Списки доступа ACL. Настройка технологии NAT

**Целью** данной практической работы является получение навыков по настройке списка доступа ACL и статического NAT

### **Задачи:**

- Ознакомиться с механизмом списков доступа ACL
- Выполнить задания по настройке списка доступа ACL (задания 1-2)
- Ознакомиться с технологией NAT
- Выполнить задание по настройке статической трансляции адресов NAT (задание 3)
- Составить отчет с последовательным выполнением заданий и скриншотами (задания 1-3)

### **Теоретическая часть**

Списки доступа (access-lists) используются в целом ряде случаев и являются механизмом задания условий, которые роутер проверяет перед выполнением каких-либо действий. *Маршрутизатор* проверяет каждый пакет и на основании вышеперечисленных критериев, указанных в **ACL** определяет, что нужно сделать с пакетом, пропустить или отбросить. Типичными критериями являются адреса отправителя и получателя пакета, тип протокола. Каждый критерий в списке доступа записывается отдельной строкой. *Список* доступа в целом представляет собой набор строк с критериями, имеющих один и тот же номер (или имя). Порядок задания критериев в списке существенен. Проверка пакета на соответствие списку производится последовательным применением критериев из данного списка (в том порядке, в котором они были введены). Пакет, который не соответствует ни одному из введенных критериев будет отвергнут. Для каждого протокола на *интерфейс* может быть назначен только один *список* доступа. Как пример ниже приведена *таблица* списка управления доступом по умолчанию:

№ правила	Подсеть	Конечная точка	Разрешить или запретить
100	0.0.0.0/0	3389	Разрешить

**Без ACL** - по умолчанию при создании конечной точки ей все разрешено.

**Разрешить** - при добавлении одного или нескольких диапазонов "разрешения" все остальные диапазоны по умолчанию запрещаются. Только пакеты из разрешенного диапазона *IP*-адресов смогут достичь конечной точки виртуальной машины.

**Запретить** - при добавлении одного или нескольких диапазонов "запретить" все другие диапазоны трафика по умолчанию разрешаются.

**Сочетание разрешения и запрета** - можно использовать сочетание правил "разрешить" и "запретить", чтобы указать вложенный разрешенный или запрещенный *диапазон IP*-адресов.

Рассмотрим два примера стандартных списков:

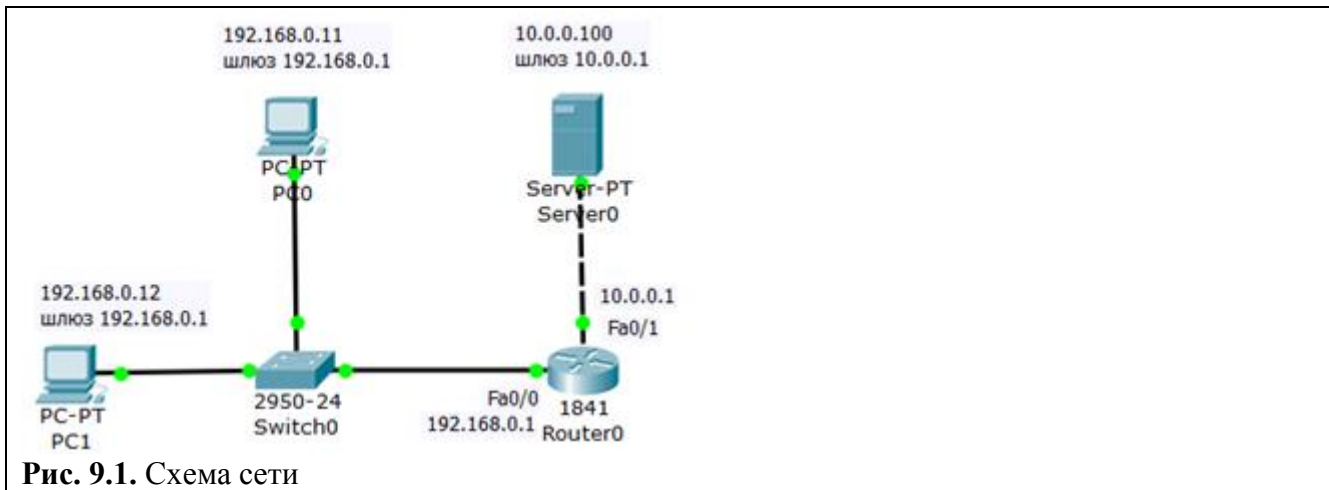
**# access-list 1 permit host 10.0.0.10** - разрешаем прохождение трафика от узла 10.0.0.10.

**# access-list 2 deny 10.0.1.0 0.0.0.255** - запрещаем прохождение пакетов из подсети 10.0.1.0/24.

## Задание 1. Создание стандартного списка доступа

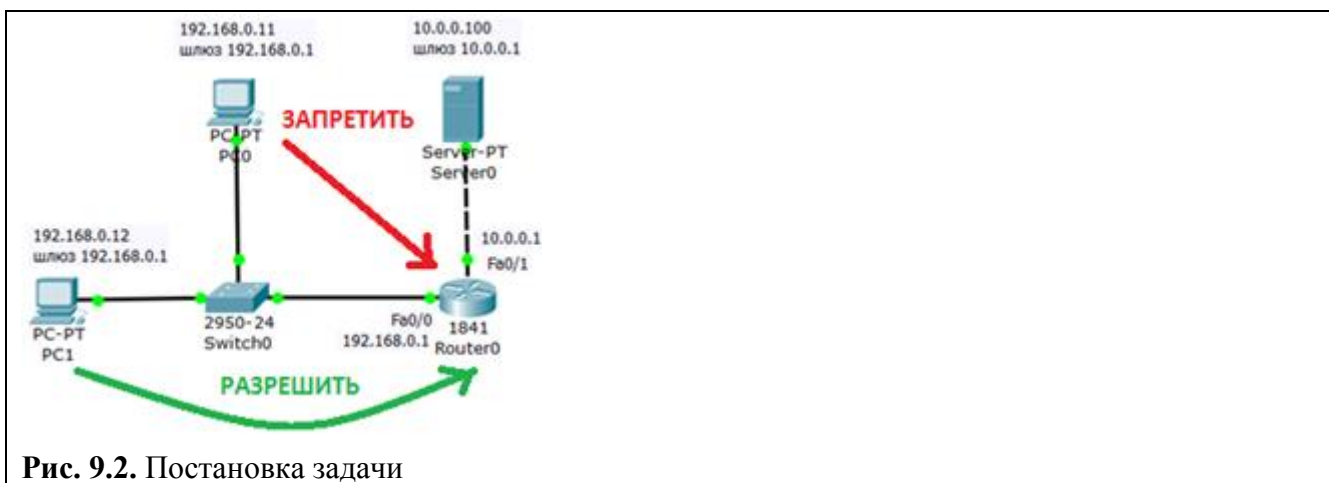
Списки доступа бывают нескольких видов: стандартные, расширенные, динамические и другие. В стандартных *ACL* есть возможность задать только *IP адрес* источника пакетов для их запретов или разрешений.

На рис. 9.1 показаны две подсети: 192.168.0.0 и 10.0.0.0.



### Постановка задачи

Требуется разрешить доступ на сервер PC1 с адресом 192.168.0.12, а PC0 с адресом 192.168.0.11 – запретить (рис. 9.2).



Соберем данную схему и настроим ее. Настройку PC1 и PC2 выполните самостоятельно.

### Настройка R0

Интерфейс 0/0 маршрутизатора 1841 настроим на адрес 192.168.0.1 и включим следующими командами:

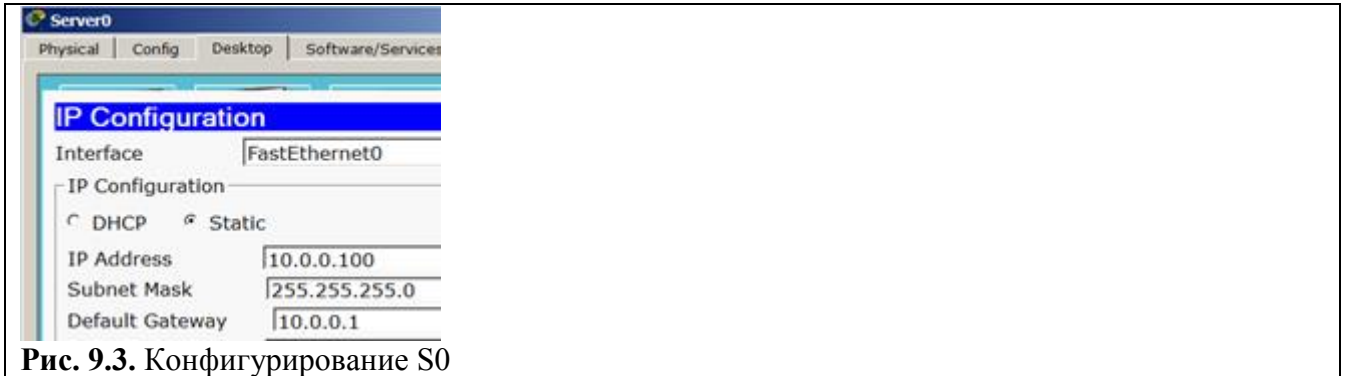
```
Router>en
Router#conf t
Router (config)#int fa0/0
Router (config-if)#ip addr 192.168.0.1 255.255.255.0
Router (config-if)#no shut
Router (config-if)#exit
```

Второй интерфейс маршрутизатора (порт 0/1) настроим на адресом 10.0.0.1 и так же включим:

```
Router (config)#intfa0/1
Router (config-if)#ip addr 10.0.0.1 255.255.255.0
Router (config-if)#no shut
```

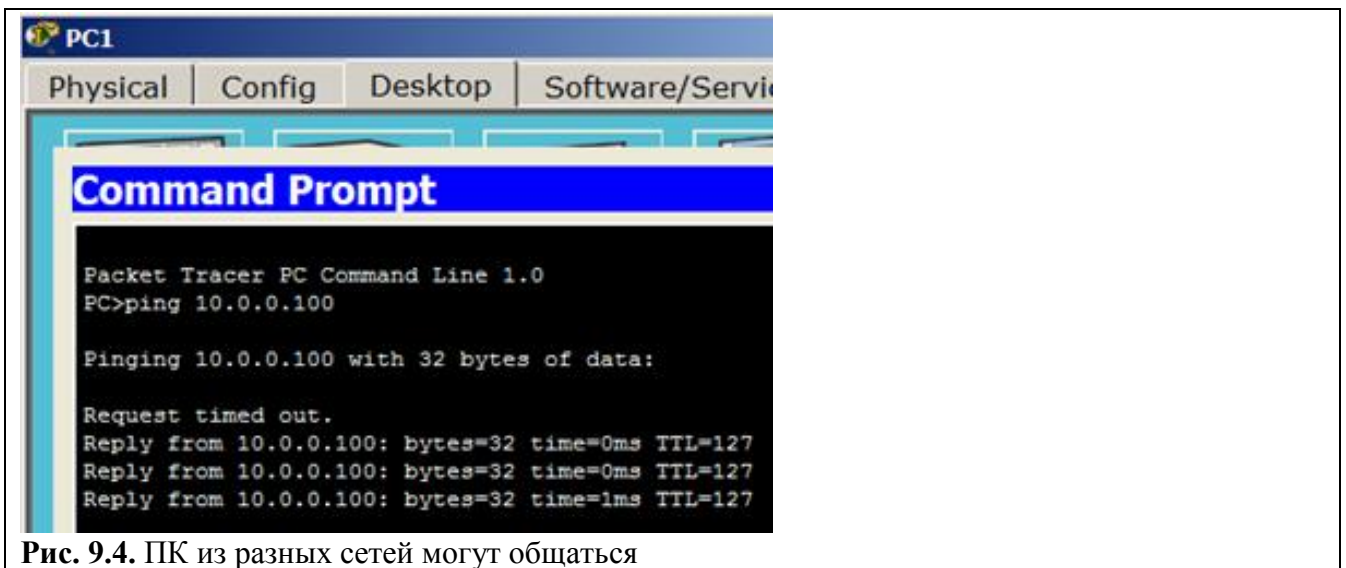
### Настройка сервера

Настройки сервера приведены на рис. 9.3.



### Диагностика сети

Проверяем связь ПК из разных сетей ( рис. 9.4).



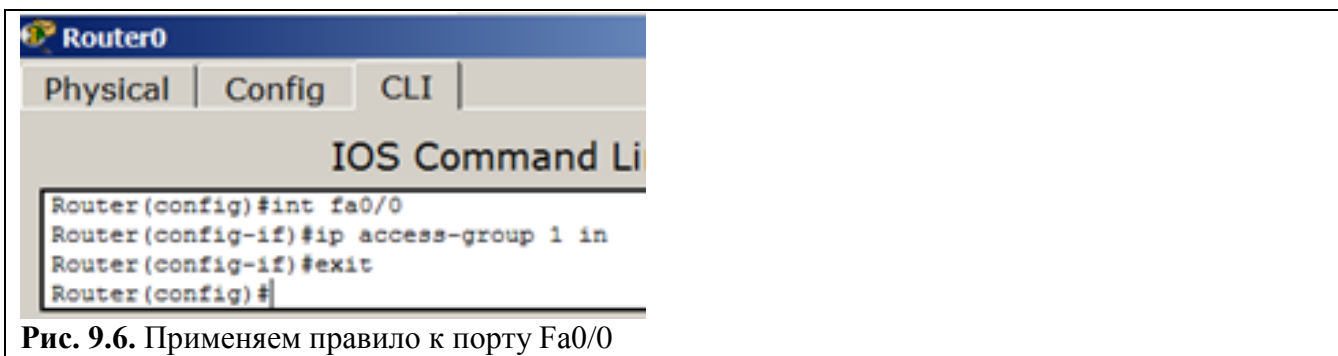
### Приступаем к решению задачи

Правило запрета и разрешения доступа будем составлять с использованием стандартных списков доступа (ACL). Пока не задан список доступа на интерфейсе всё разрешено (**permit**). Но, стоит создать список, сразу действует механизм "Всё, что не разрешено, то запрещено". Поэтому нет необходимости что-то запрещать (**deny**) – указываем что разрешено, а "остальным – запретить" подразумевается автоматически. По условиям задачи нам нужно на R0 пропустить пакеты с узла 192.168.0.12 на сервер ( рис. 9.5).



**Рис. 9.5.** Создаем на R0 разрешающий ACL

Применяется данное правило на интерфейс в зависимости от направления (PC1 расположен со стороны порта Fa0/0) – рис. 9.6. Эта настройка означает, что список доступа (правило с номером 1) будет действовать на интерфейсе fa0/0 на входящем (in) от PC1 направлении.

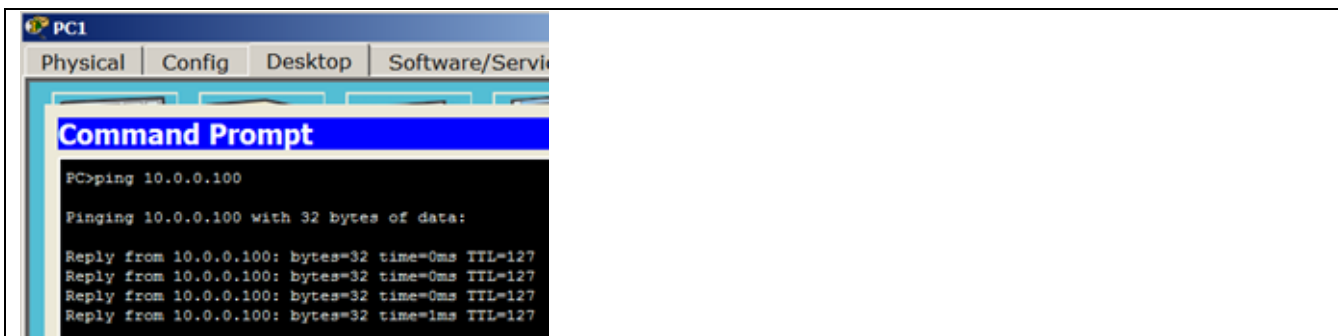


**Рис. 9.6.** Применяем правило к порту Fa0/0

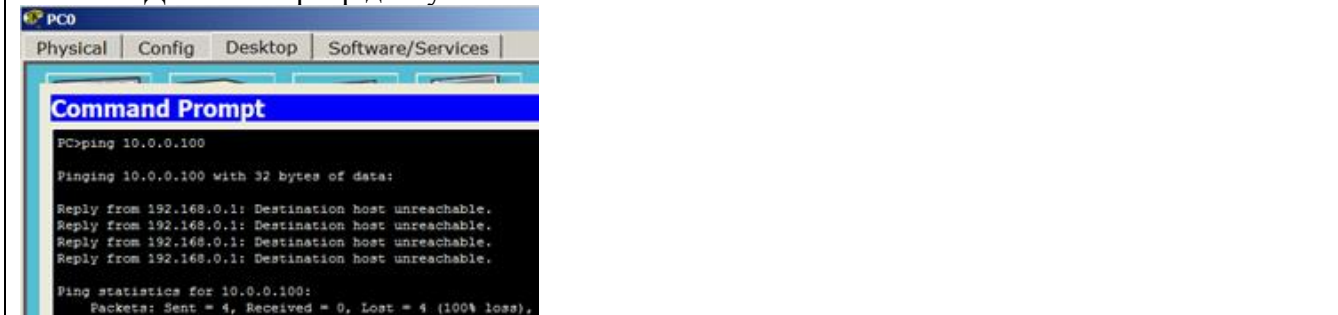
**Примечание**

Входящий трафик (in) — это тот, который приходит на интерфейс извне. Исходящий (out) — тот, который отправляется с интерфейса вовне. Список доступа вы можете применить либо на входящий трафик, тогда неудобные пакеты не будут даже попадать на маршрутизатор и соответственно, дальше в сеть, либо на исходящий, тогда пакеты приходят на маршрутизатор, обрабатываются им, доходят до целевого интерфейса и только на нём обрабатываются. Как правило, списки применяют на входящий трафик (in).

Проверяем связь ПК с сервером ( рис. 9.7 и рис. 9.8).



**Рис. 9.7.** Для PC1 сервер доступен



**Рис. 9.8.** Для PC0 сервер не доступен

Давайте посмотрим ACL ( рис. 9.9).



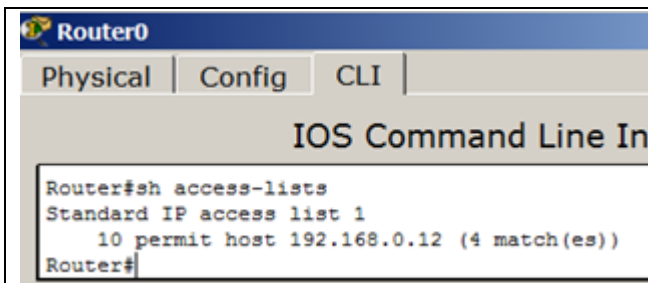


Рис. 9.9. Узел 192.168.0.12 разрешен

**Примечание**

Теперь, предположим, нужно добавить новый узел, например, PC2 с адресом 192.168.0.13 в раздел "разрешённых". Пишем команду Router (config)#access-list 1 permit host 192.168.0.13. Теперь адреса 192.168.0.12 и 192.168.0.13 могут общаться с сервером, в 192.168.0.11 – нет. А для отмены какого-либо правила – повторяем его с приставкой "no". Тогда это правило исключается из конфигурации. Например, если выполнить команду Router (config-if)#no ip access-group 1 in, то ACL будет отменен и снова все ПК могут пинговать сервер.

**Задание 2. Расширенные списки доступа ACL**

**Расширенные списки доступа ACL**

Стандартные *права* не так гибки, как хотелось бы. В отличие от стандартных списков, расширенные списки фильтруют трафик более "тонко". При создании расширенных списков в правилах доступа можно включать фильтрацию трафика по протоколам и портам. Для указания портов в правиле доступа указываются следующие обозначения ():

Таблица 9.1. Обозначение портов в ACL

обозначение	действие
lt n	Все номера портов, меньше n.
gt n	Все номера портов, больше n.
eq n	Порт n
neq n	Все порты, за исключением n.
range n m	Все порты от n до m включительно.

Соберите схему сети, показанную на рис. 9.10.

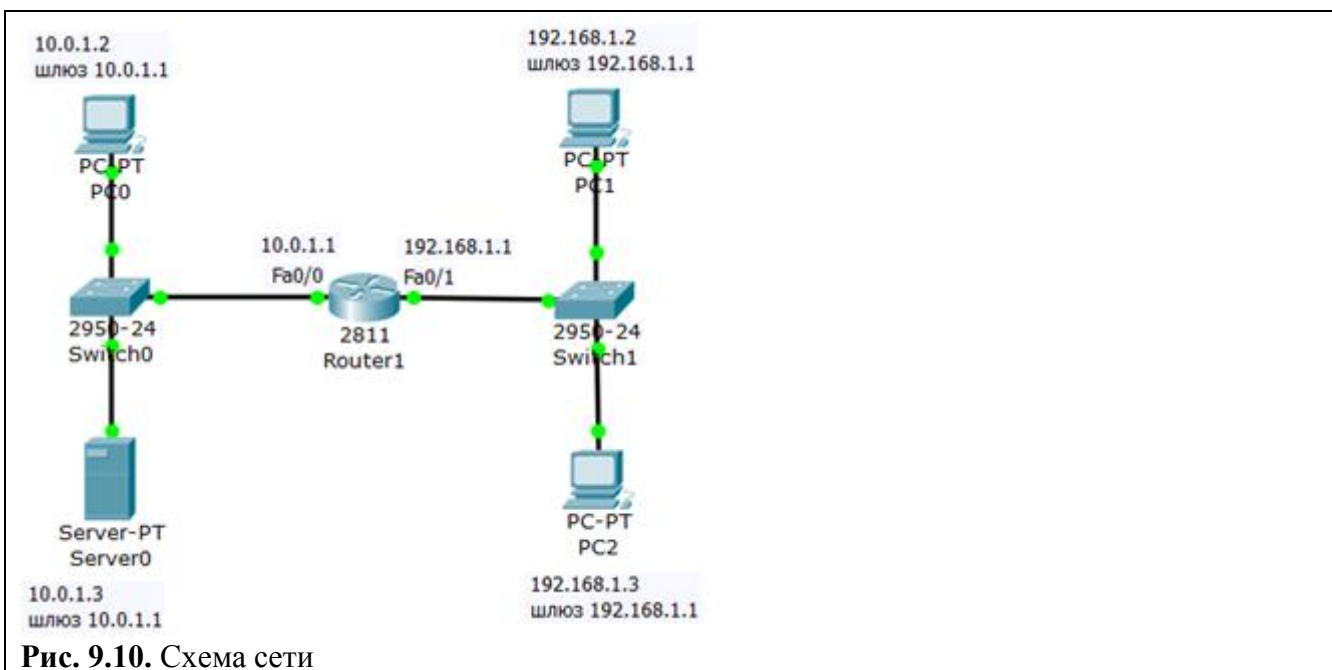


Рис. 9.10. Схема сети

Задача: разрешить *доступ* к FTP серверу 10.0.1.3 для узла 192.168.1.2 и запретить для узла 192.168.1.3.

### Создаем расширенные списки доступа и запрещаем FTP трафик

Постановка задачи графически изображена на рис. 9.11.

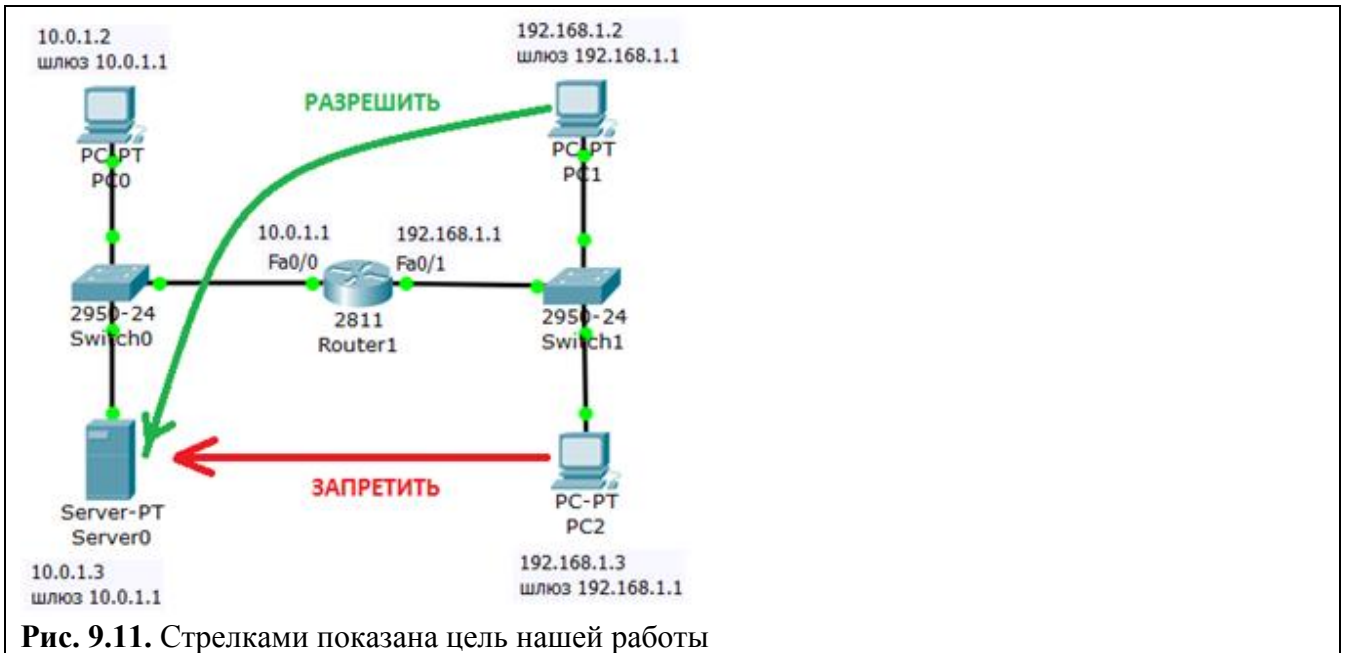


Рис. 9.11. Стрелками показана цель нашей работы

Изначально на сервере 10.0.1.3 FTP сервис поднят по умолчанию со значениями имя пользователя Cisco, пароль Cisco. Убедимся, что узел S0 доступен и FTP работает, для этого заходим на PC1 и связываемся с сервером (рис. 9.12). Выполняем какие-либо команды, например, DIR – чтение директории.

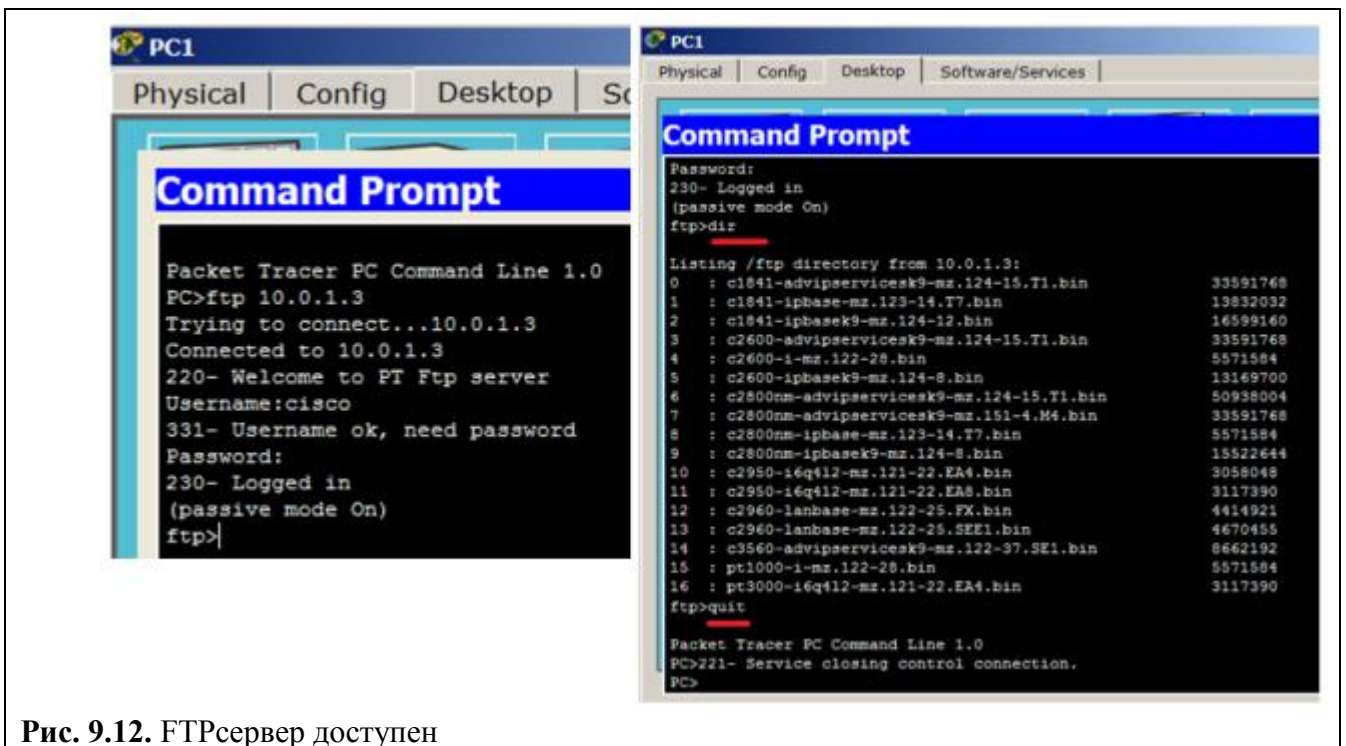


Рис. 9.12. FTP сервер доступен

### Примечание

При наборе пароля на экране ничего не отображается.



Теперь создадим список правил с номером 101 в котором укажем 2 разрешающих и по 2 запрещающих правила для портов сервера 21 и 20 (Эти порты служат для FTP - передачи команд и данных) – рис. 9.13.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip access-list extended 101
Router(config-ext-nacl)#permit tcp 192.168.1.2 0.0.0.0 10.0.1.3 0.0.0.0 eq 21
Router(config-ext-nacl)#permit tcp 192.168.1.2 0.0.0.0 10.0.1.3 0.0.0.0 eq 20
Router(config-ext-nacl)#deny tcp 192.168.1.3 0.0.0.0 10.0.1.3 0.0.0.0 eq 21
Router(config-ext-nacl)#deny tcp 192.168.1.3 0.0.0.0 10.0.1.3 0.0.0.0 eq 20
Router(config-ext-nacl)#deny tcp 192.168.1.3 0.0.0.0 10.0.1.3 0.0.0.0 eq 20
Router(config-ext-nacl)#
```

**Рис. 9.13.** Составляем расширенные списки доступа

#### Совет

Набирайте команды аккуратно и внимательно: даже один лишний пробел может привести к ошибке при выполнении команды.

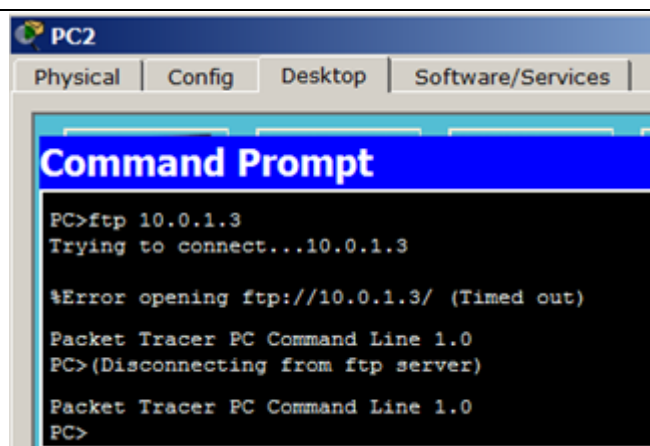
А теперь применяем наш список с номером 101 на вход (in) Fa0/1 потому, что трафик входит на этот порт роутера со стороны сети 192.168.1.0 ( рис. 9.14).

```
Router(config-ext-nacl)#int fa0/1
Router(config-if)#ip access-group 101 in
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#wr mem
Building configuration...
[OK]
Router#
```

**Рис. 9.14.** Применяем правило с номером 101 к порту 0/1 роутера

Проверяем связь сервера с PC2 ( рис. 9.15).



**Рис. 9.15.** Для PC2 FTP сервер не доступен

Проверяем связь сервера с PC1 ( рис. 9.16).



Рис. 9.16. Для PC1 FTP сервер доступен

## Настройка статического NAT

*NAT (Network Address Translation) — трансляция сетевых адресов, технология, которая позволяет преобразовывать (изменять) IP адреса и порты в сетевых пакетах. NAT используется чаще всего для осуществления доступа устройств из локальной сети предприятия в Интернет, либо наоборот для доступа из Интернет на какой-либо ресурс внутри сети. Локальная сеть предприятия строится на частных IP адресах:*

- 10.0.0.0 — 10.255.255.255 (10.0.0.0/255.0.0.0 (/8))
- 172.16.0.0 — 172.31.255.255 (172.16.0.0/255.240.0.0 (/12))
- 192.168.0.0 — 192.168.255.255 (192.168.0.0/255.255.0.0 (/16))

Эти адреса не маршрутизируются в Интернете, и провайдеры должны отбрасывать пакеты с такими IP адресами отправителей или получателей. Для преобразования частных адресов в Глобальные (маршрутизируемые в Интернете) применяют *NAT*.

**NAT** — технология трансляции сетевых адресов, т.е. подмены адресов (или портов) в заголовке IP-пакета. Другими словами, пакет, проходя через маршрутизатор, может поменять свой адрес источника и/или назначения. Подобный механизм служит для обеспечения доступа из LAN, где используются частные IP-адреса, в Internet, где используются глобальные IP-адреса.

Существует три вида трансляции *Static NAT*, *Dynamic NAT*, *Overloading (PAT)*.

- **Static NAT (статический NAT)** осуществляет преобразование IP адреса один к одному, то есть сопоставляется один адрес из внутренней сети с одним адресом из внешней сети. Иными словами, при прохождении через маршрутизатор, адрес(а) меняются на строго заданный адрес, один-к-одному (Например, 10.1.1.5 всегда заменяется на 11.1.1.5 и обратно). Запись о такой трансляции хранится неограниченно долго, пока есть соответствующая строка в конфигурации роутера.

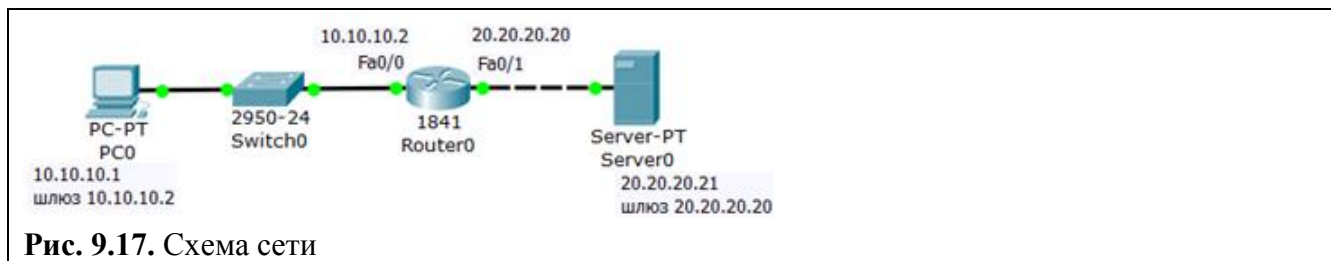
- **Dynamic NAT (динамический NAT)** производит преобразование внутреннего адреса/ов в один из группы внешних адресов. То есть, перед использованием динамической трансляции, нужно задать *nat*-пул внешних адресов. В этом случае при прохождении через маршрутизатор, новый адрес выбирается динамически из некоторого диапазона адресов, называемого пулом (*pool*). Запись о трансляции хранится некоторое время, чтобы ответные пакеты могли быть доставлены адресату. Если в течение некоторого времени трафик по этой трансляции отсутствует, трансляция удаляется и адрес возвращается в пул. Если требуется создать трансляцию, а свободных адресов в пуле нет, то пакет отбрасывается. Иными словами, хорошо бы, чтобы число внутренних адресов было ненамного больше числа адресов в пуле, иначе высока вероятность проблем с выходом в WAN.

- **Overloading (или PAT)** позволяет преобразовывать несколько внутренних адресов в один внешний. Для осуществления такой трансляции используются порты, поэтому такой NAT называют **PAT (Port Address Translation)**. С помощью **PAT** можно преобразовывать

внутренние адреса во внешний адрес, заданный через пул или через адрес на внешнем интерфейсе.

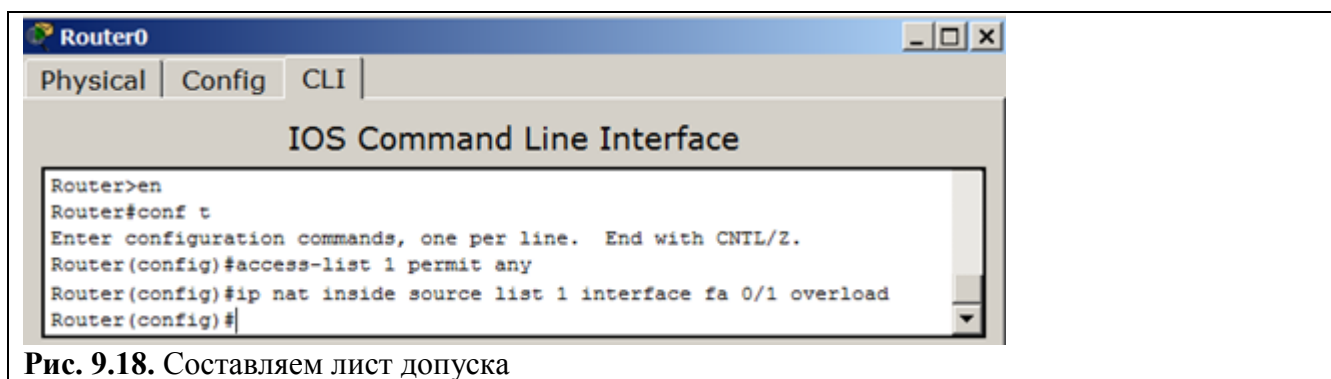
### Задание 3. Статическая трансляция адресов NAT

На рис. 9.17 имеется внешний адрес 20.20.20.20 (внешний интерфейс fa0/1) и внутренняя сеть 10.10.10.0 (внутренний интерфейс fa0/0). Нужно настроить NAT. Предполагается, что адреса уже прописаны, и сеть поднята (рабочая).



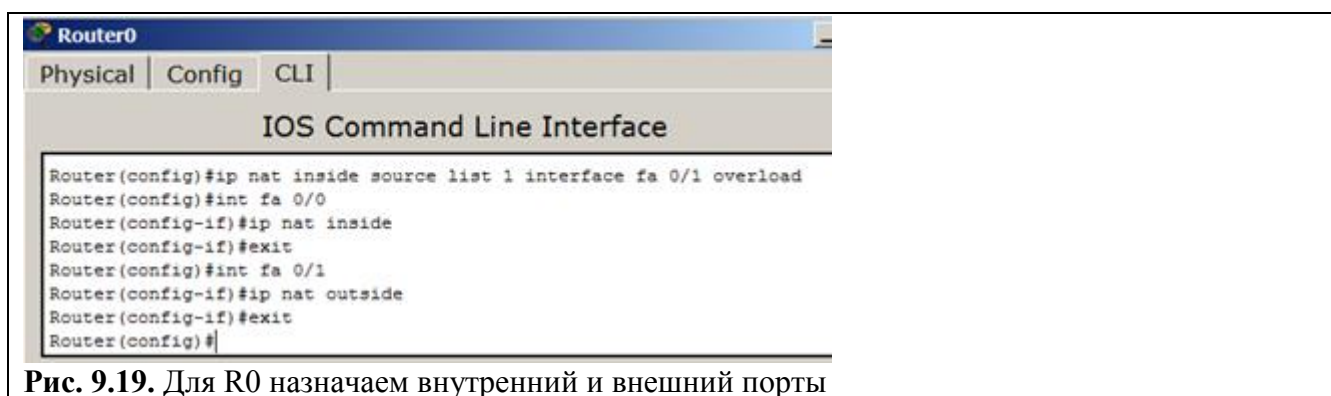
На R0 добавляем access-list, разрешаем всё (any)

Разрешаем весь трафик, то есть, любой IP адрес (рис. 9.18).



Создаём правило трансляции

Теперь настроим трансляцию на интерфейсах (на внутреннем inside, на внешнем – outside), то есть, для R0 указываем внутренний и внешний порты (рис. 9.19).



Выходим из режима глобального конфигурирования и записываем настройки роутера в микросхему памяти (рис. 9.20).

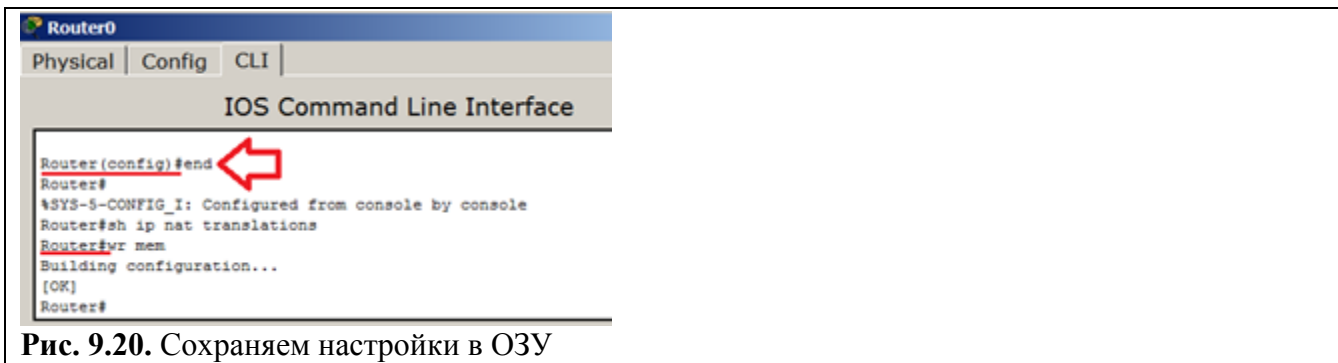


Рис. 9.20. Сохраняем настройки в ОЗУ

Проверяем работу сети (просмотр состояния таблицы NAT)

С PC0 пингуем провайдера и убеждаемся, что PC1 и сервер могут общаться ( рис. 9.21).

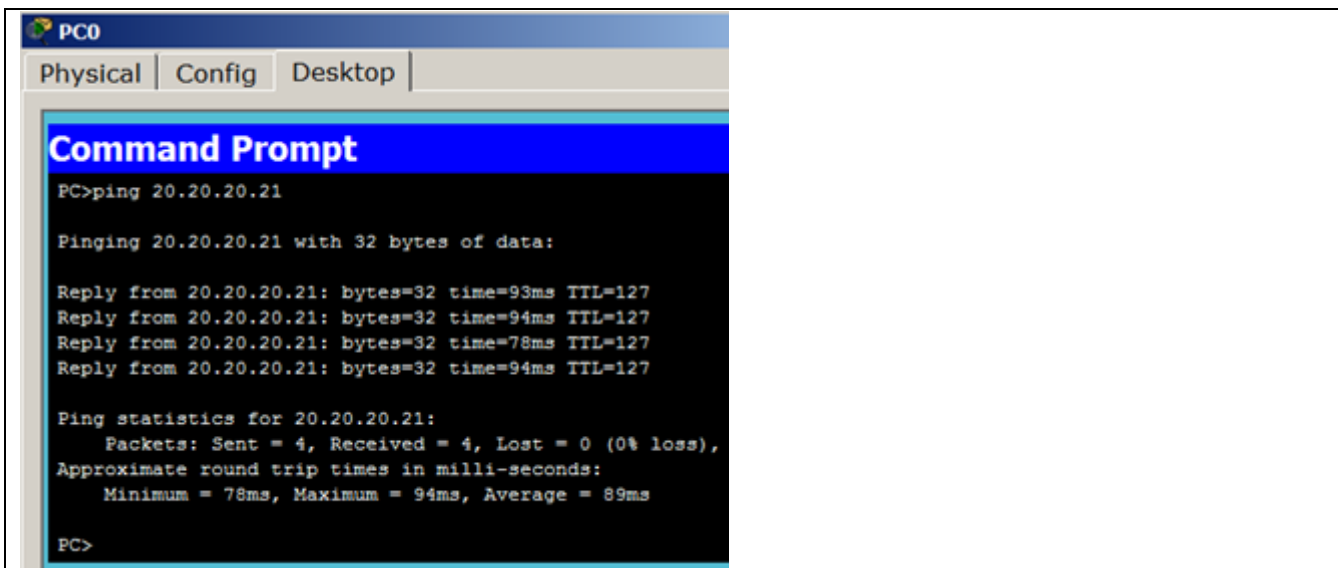


Рис. 9.21. Из внутренней сети пингуем внешнюю сеть

Для просмотра состояния таблицы NAT, одновременно с пингом используйте команду **Router#sh ip nat translations** (я запустил пинг с машины 10.10.10.1, т.е., с PC1 на адрес 20.20.20.21, т.е., на S0) – рис. 9.22.

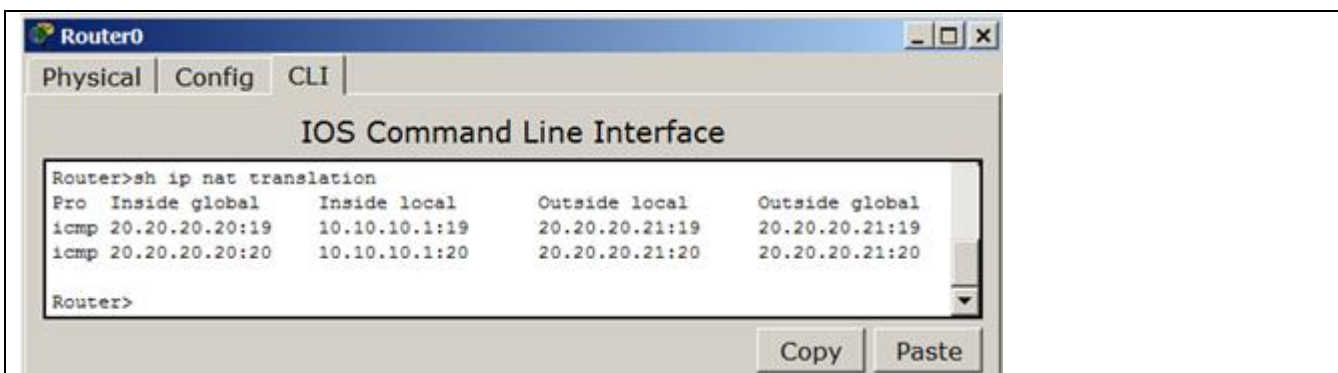


Рис. 9.22. Вовремя пинга просматриваем состояние таблицы NAT

Убеждаемся в успешной маршрутизации в режиме симуляции ( рис. 9.23).

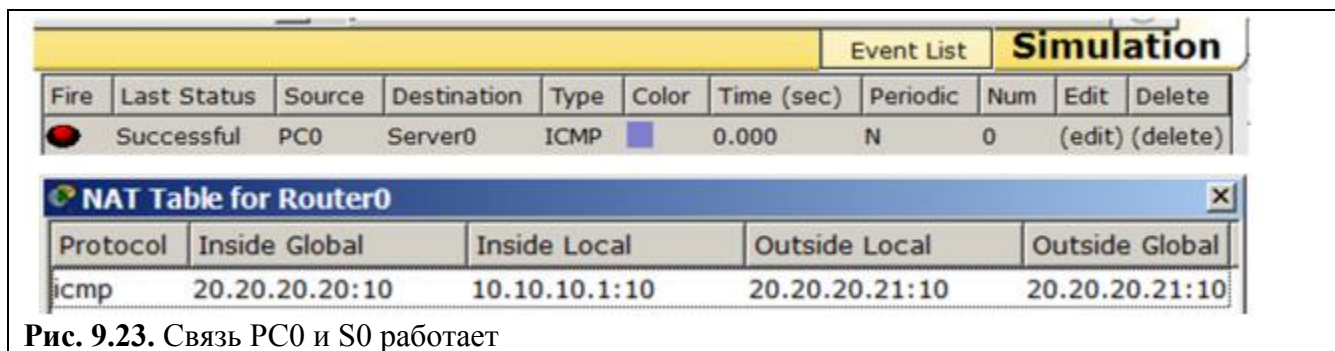


Рис. 9.23. Связь PC0 и S0 работает

### Форма представления и содержание отчета

Отчет в формате DOC или PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.

### *Содержание отчета*

1. Заголовок согласно приложению.
2. Цель работы.
3. Ход работы:
  - г. Данный раздел состоит из последовательного описания значимых выполняемых шагов (с указанием их сути) Пояснения работы команды ping и **содержимго протоколов**.
  - h. копий экранов (должна быть видна набранная команда и реакция системы, если она есть).
4. По каждому пункту практической работы должны быть приведены **выводы**.

### Практическая работа №11. Беспроводные сети

**Целью** данной практической работы является получение навыков по проектированию и настройке беспроводной сети с беспроводным роутером или точкой доступа

#### **Задачи:**

- Ознакомиться с видами беспроводного оборудования и его видами шифрования
- Выполнить задания по настройке беспроводной сети с беспроводным роутером и типами шифрования WEP, WPA2 (задания 1-2)
- Выполнить задания по настройке беспроводной сети с точкой доступа (задания 3-4)
- Составить отчет с последовательным выполнением заданий и скриншотами (задания 1-4)

#### **Теоретическая часть**

##### **Беспроводная сеть WEP**

Создание новой беспроводной сети начинается непосредственно с конфигурации точки доступа - беспроводного маршрутизатора (роутера) подключения к ней компьютеров и другого беспроводного оборудования. Классический способ настройки такой: вначале производится подключение к

точке доступа оборудования, а затем нужно задать вручную имя беспроводной сети и ключ безопасности. Далее мы рассмотрим различные варианты беспроводных сетей и способы их настройки в программе СРТ. Ключ безопасности беспроводной сети - уникальный код (пароль), который закрывает доступ к вашей сети. При этом важен не столько сам ключ, сколько тип шифрования. Дело в том, что вся информация, которая протекает между роутером и ПК шифруется. И если вы ввели неправильный ключ, то ваше устройство просто не сможет декодировать ее. Это сделано для повышения безопасности. Стоит отметить, что на сегодняшний день существует три типа шифрования Wi-Fi подключений: WPA. WPA2. WEP.

**WEP (Wired Equivalent Privacy)** — алгоритм для обеспечения безопасности сетей Wi-Fi. Используется для обеспечения защиты, передаваемых данных

### Задание 1. Беспроводная сеть с беспроводным роутером. WEP шифрование

Схема для работы показана на рис. 10.1.

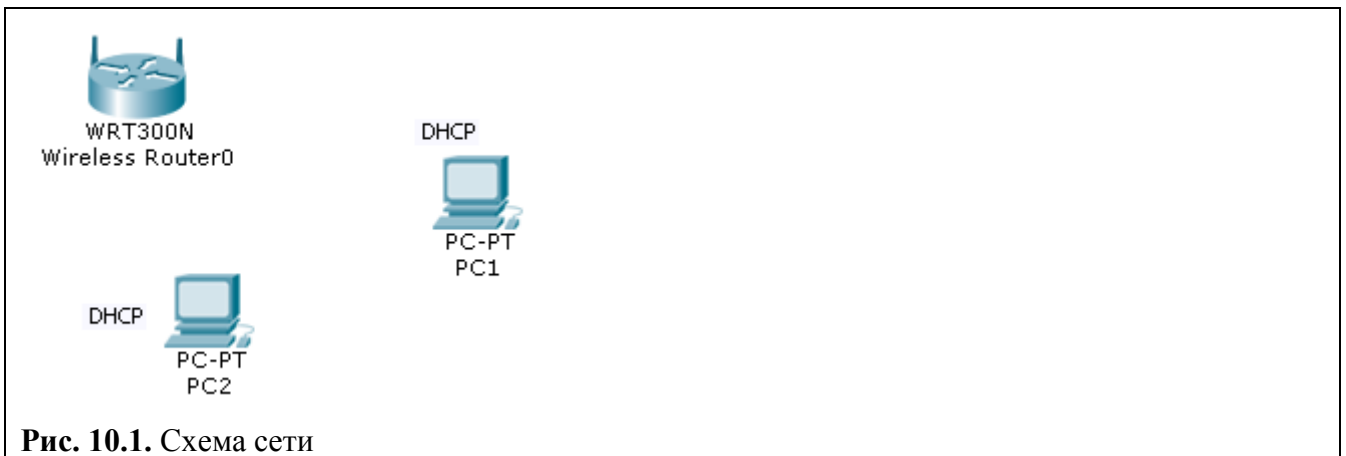
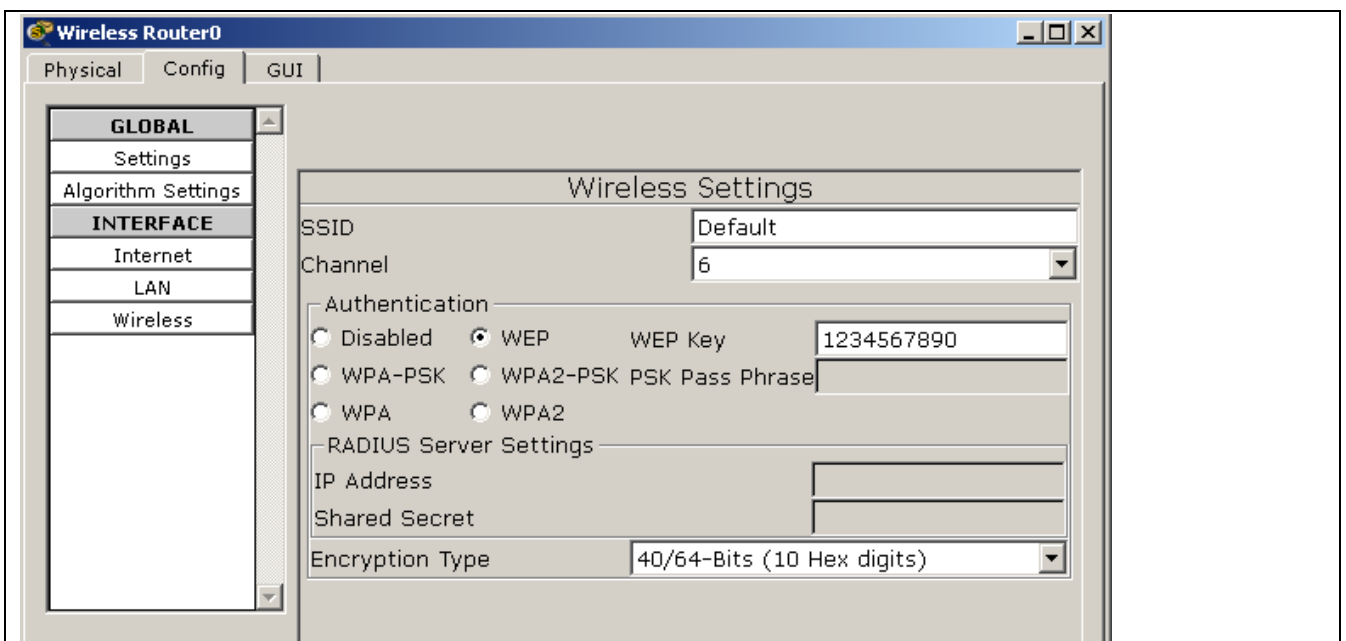


Рис. 10.1. Схема сети

Добавим Wireless Router0 из Wireless Devices на схему сети.

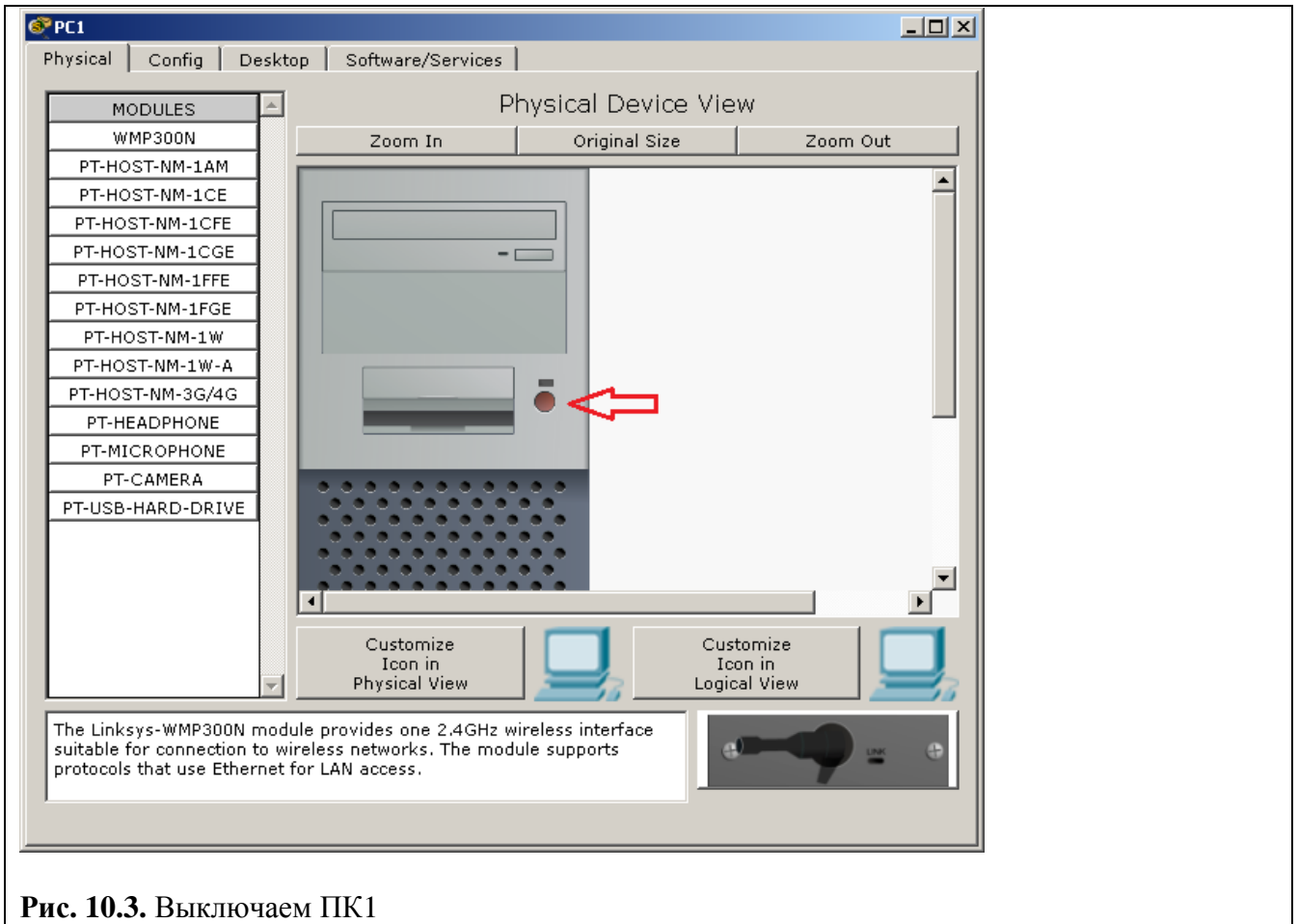
Настроим WEP шифрование в беспроводном роутере Wireless Router0 (рис 10.2)





**Рис. 10.2.** Настраиваем Wireless Router0

Вставим в ПК беспроводный адаптер WMP300N. Для этого сначала выключим ПК и уберем из него модуль PT-HOST-NM-1CFE (рис. 10.3, рис 10.4).



**Рис. 10.3.** Выключаем ПК1

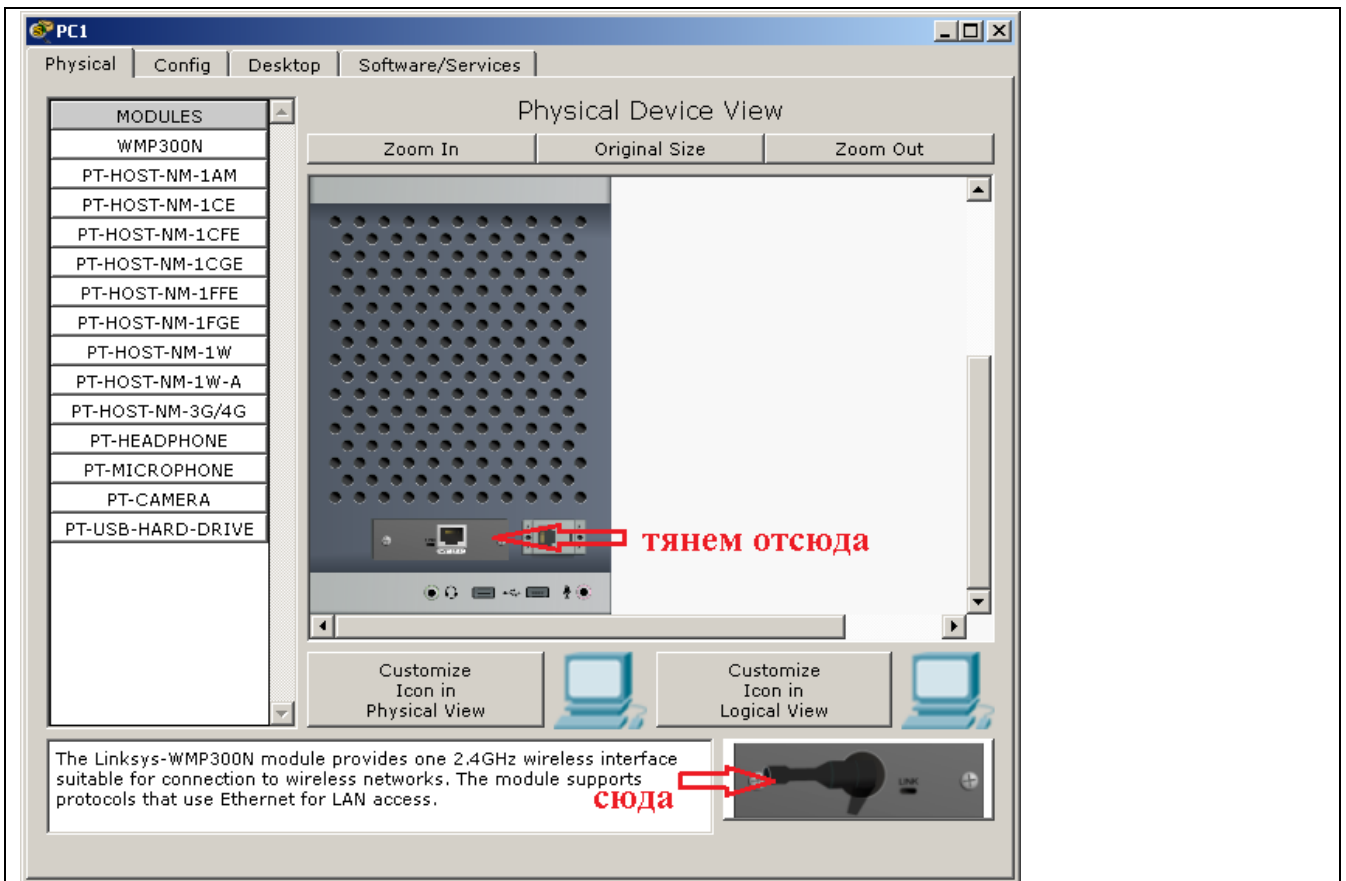


Рис. 10.4. Удаляем модуль PT-HOST-NM-1CFE

Вставляем беспроводной адаптер WMP300N (рис. 10.5).

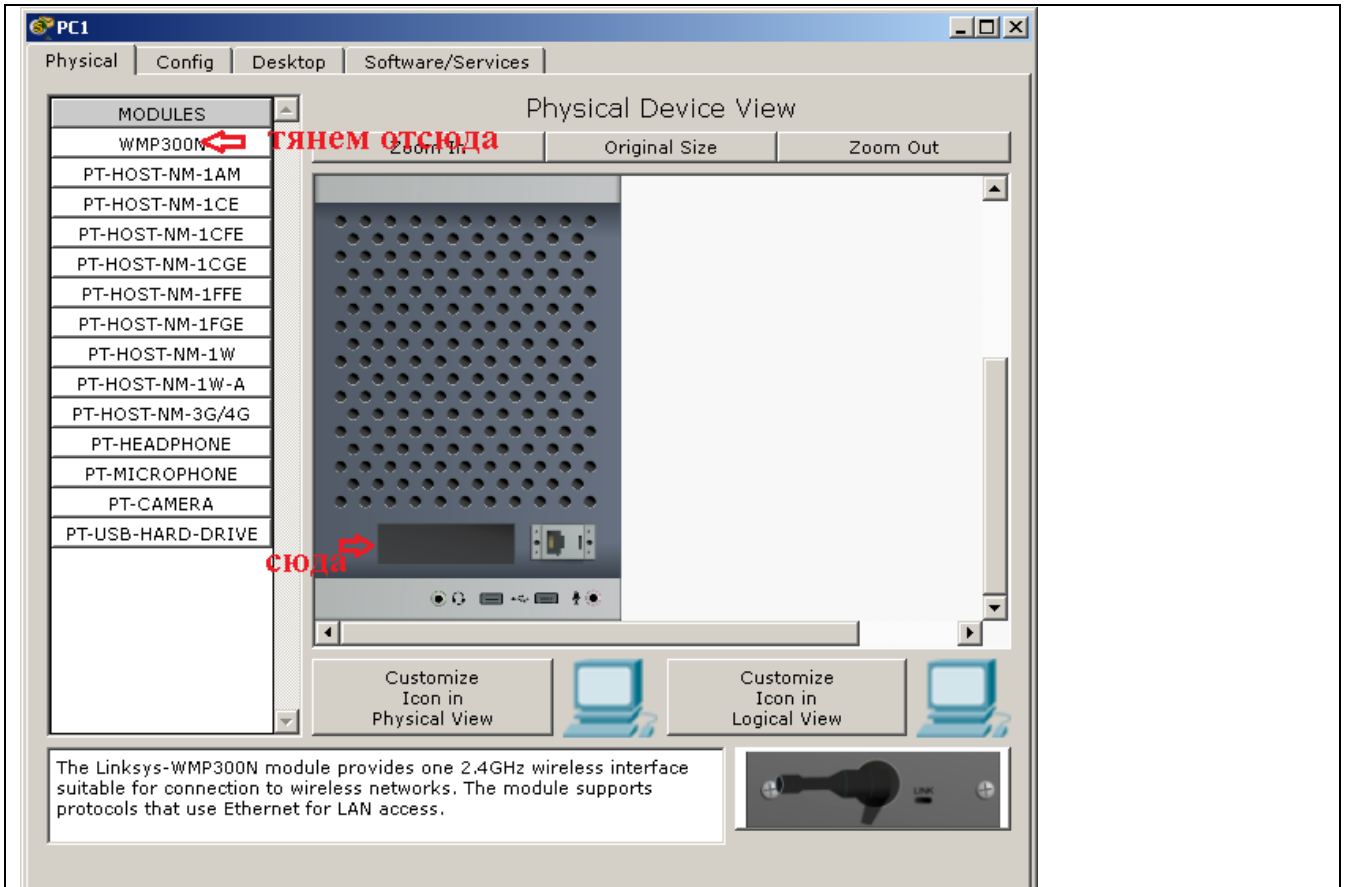


Рис. 10.5. Оснащаем ПК беспроводным адаптером



Реальный вид беспроводного адаптера WMP300N приведен на рис. 10.6.



Рис. 10.6. Беспроводный адаптер WMP300N

Обязательно включим ПК1 и настроим беспроводный адаптер на PC1 (рис. 10.7).

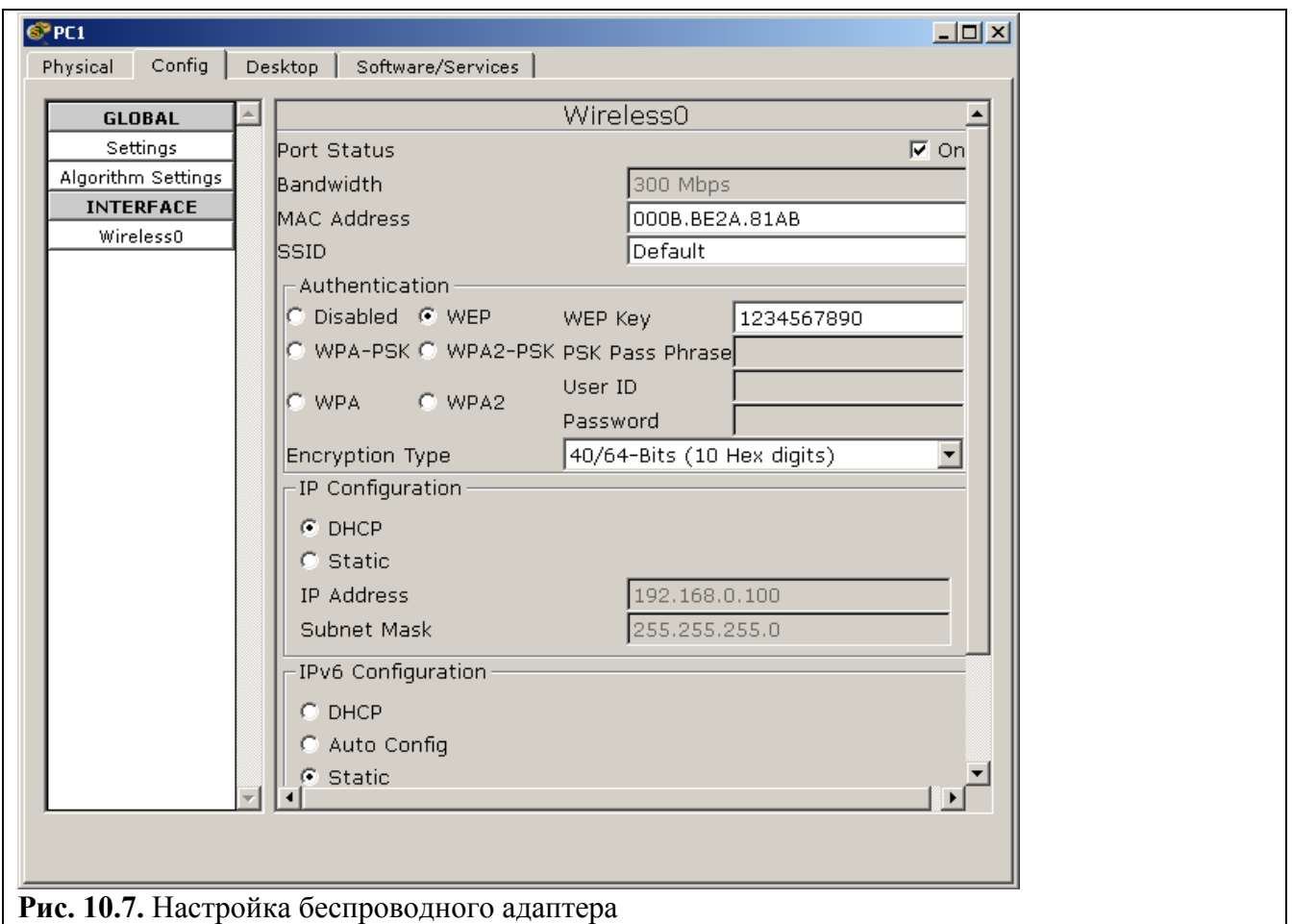
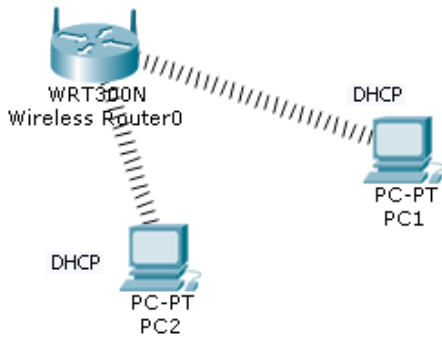


Рис. 10.7. Настройка беспроводного адаптера

Аналогично проделываем для PC2.

В результате должна получиться следующая схема сети (рис 10.8)



**Рис. 10.8.** Установлено беспроводное соединение между устройствами

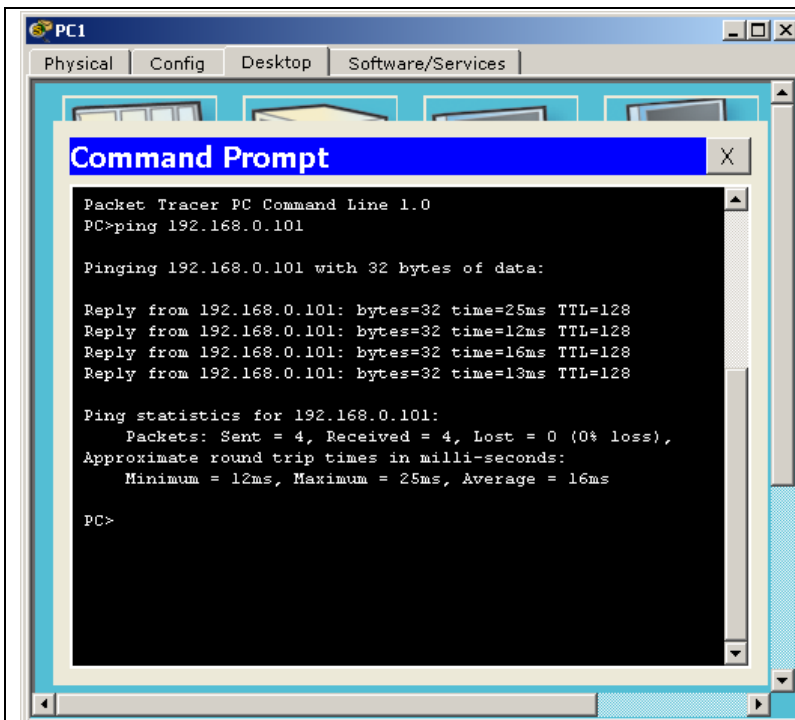
Узнаем ip-адрес PC2 (рис10.9)

PC1

Port	Link	IP Address
Wireless0	Up	192.168.0.101/24

**Рис 10.9** DHCP ip-адрес PC2

Проверим связь между ПК1 и ПК2 (рис. 10.10).



**Рис. 10.10.** Проверка связи между ПК1 и ПК2

Также проверьте связь между ПК1 и ПК2 в режиме симуляции (выбрать только протокол ICMP)

## Беспроводная сеть WPA

Типы шифрования сети WPA и WPA2 требуют от абонентов введение уникального пароля. Без него вы просто не сможете выполнить подключение. После проверки введенного ключа все данные, которые передаются между участниками сети, шифруются. Современные роутеры

поддерживают обе технологии. Но, WPA2 все же предоставляет более высокую защиту. Поэтому по возможности следует выбирать именно его.

**WPA** (Wi-Fi Protected Access) — представляет собой технологию защиты беспроводной Wi-Fi сети. Плюсами WPA являются усиленная безопасность данных и ужесточенный контроль доступа к беспроводным сетям, а также - совместимость между множеством беспроводных устройств как на аппаратном уровне, так и на программном.

Протокол безопасности **WPA2-PSK** - упрощенный вариант WPA2. Технологии защиты беспроводных сетей WPA2 является самой лучшей на сегодняшний день. Но, из соображений совместимости на маршрутизаторах можно встретить ее вариант WPA2-PSK.

## Задание 2. Беспроводная сеть с беспроводным роутером. WPA-2 шифрование

На рис. 10.11 приведена схема сети с беспроводным роутером.



Рис. 10.11. Схема сети с беспроводным роутером

Если мы снабдим оба ПК беспроводным модулем, то в данной сети мы можем наблюдать появление WIFI связи (рис. 10.12).



Рис. 10.12. Мы можем наблюдать появление WIFI связи

Зайдем на роутер и посмотрим на его IP-address. Как видим, включен DHCP service и роутер получает IP адрес автоматически (рис. 10.13).

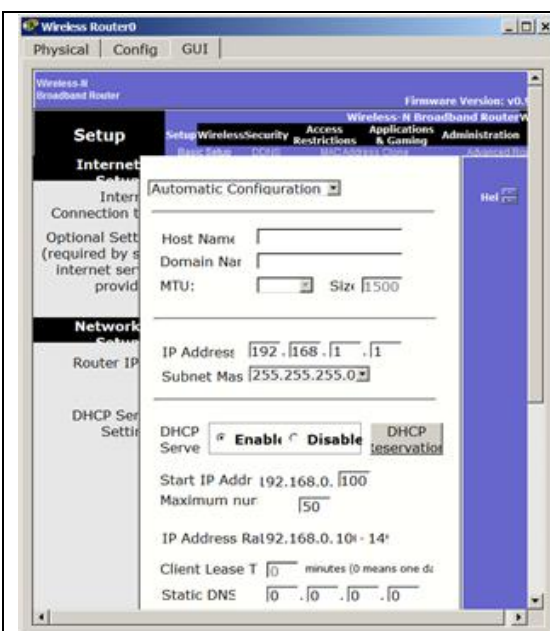


Рис. 10.13. Автоматическое конфигурирование роутера

Теперь на вкладке Config настроим аутентификацию роутера (рис. 10.14).

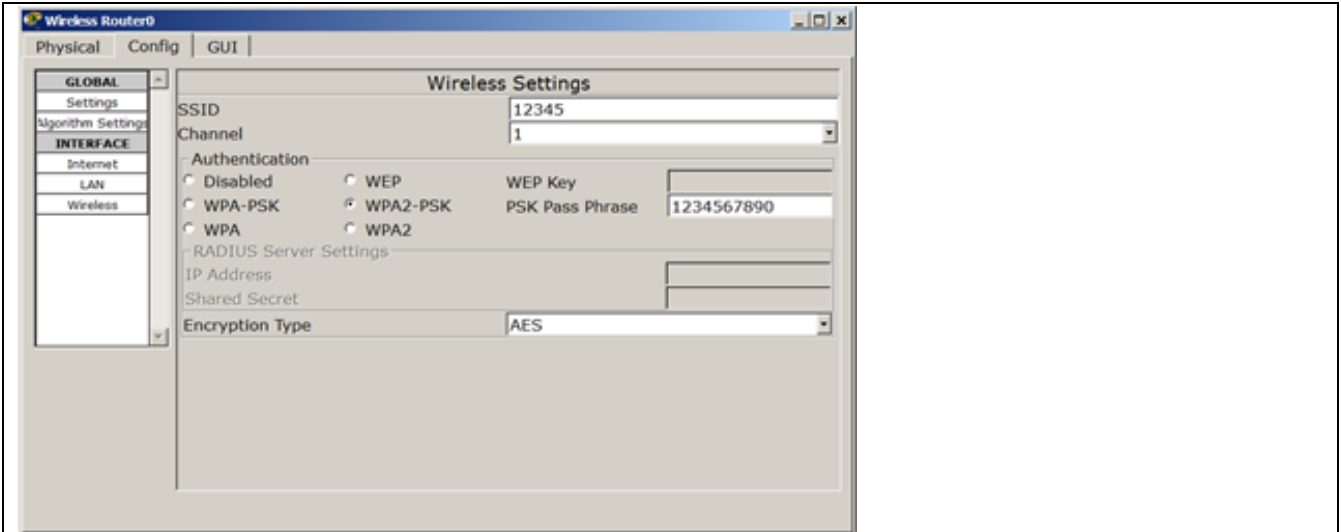


Рис. 10.14. Вводим SSID и WPA2-PSK

Теперь для PC0 заходим в меню PC Wireless (рис. 10.15).



Рис. 10.15. Заходим в меню PC Wireless

Устанавливаем соединение PC0 и роутера (рис. 10.16).



Рис. 10.16. На вкладке Connect нажимаем на кнопку Connect

Для аутентификации необходим WPA2-PSK пароль, т.е. 1234567890 (рис. 10.17).

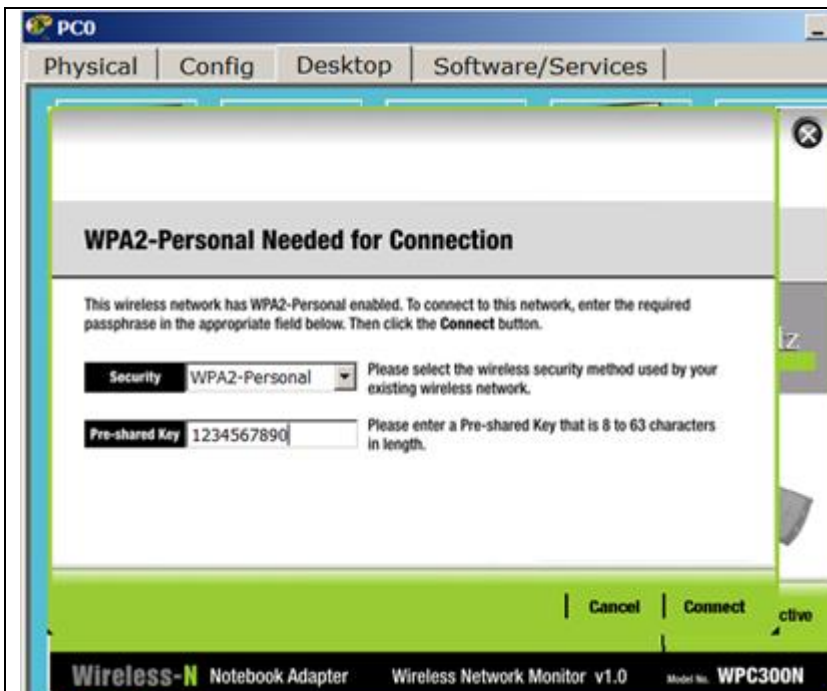


Рис. 10.17. Вводим пароль и нажимаем на кнопку Connect

Итак, мы предъявили наш "пропуск" на вход пользователя в сеть и связь устройств установлена (рис. 10.18).

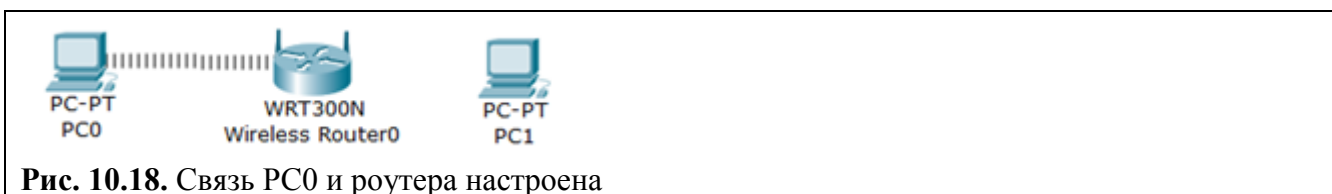


Рис. 10.18. Связь PC0 и роутера настроена

Аналогично подключаем PC1 и появится связь роутера и PC1 (рис. 10.19).

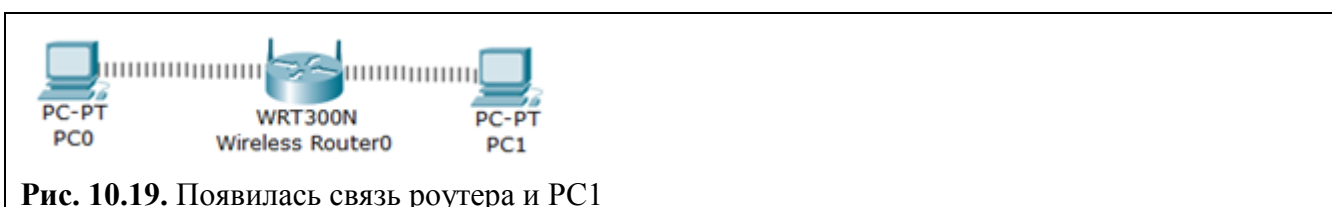


Рис. 10.19. Появилась связь роутера и PC1

Узнаем динамический IP адрес для PC1 и пингуем его с PC0 (рис. 10.20).

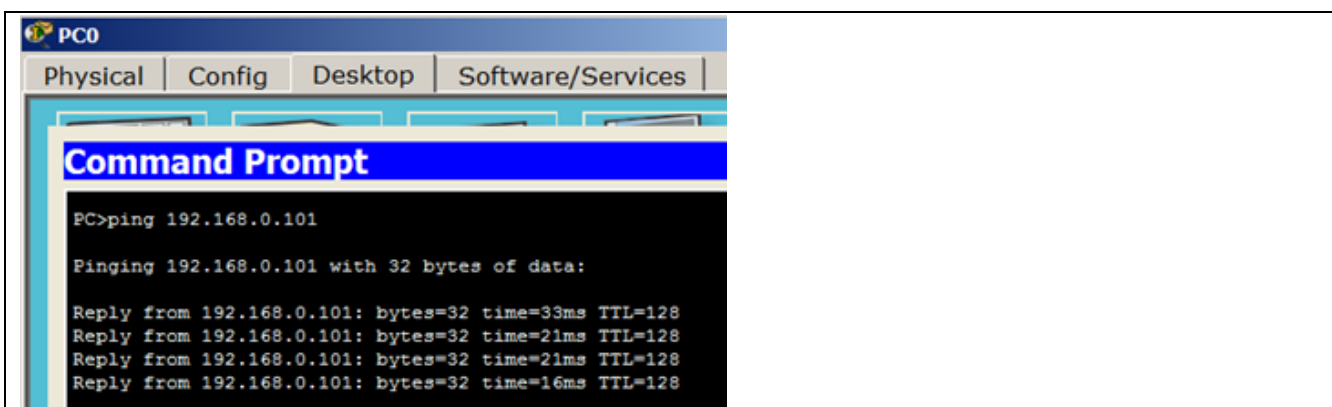


Рис. 10.20. Связь между PC0 и PC1 есть

### Задание 3. Беспроводная сеть с точкой доступа

Соберите схему сети, представленную на рис. 10.21.

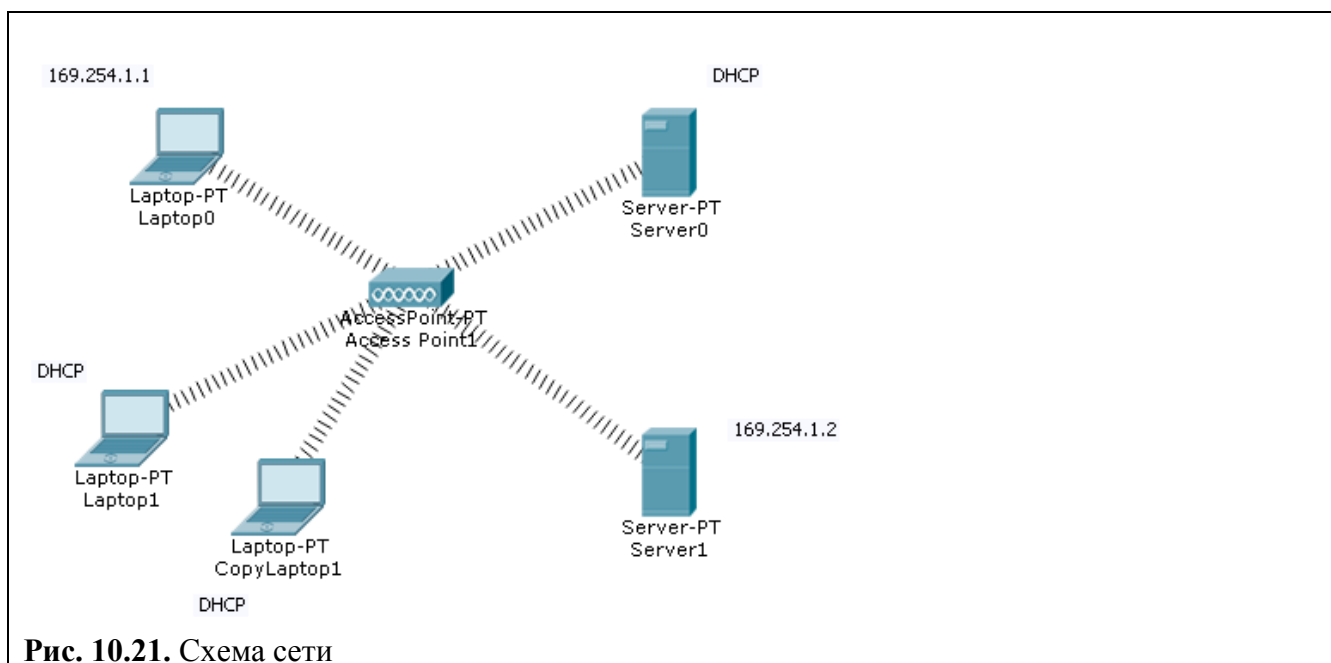


Рис. 10.21. Схема сети

Точка доступа в английской терминологии – Wireless Access Point.

На всех устройствах извлечем сетевой модуль PT-HOST-NM-1CFE и заменим его на беспроводной модуль WMP300N

**Необходимо настроить все устройства по схеме сети (рис 10.21).**

Настройка точки доступа не требуется.

Laptop1 укажем DHCP ip-адрес. (рис 10.22)

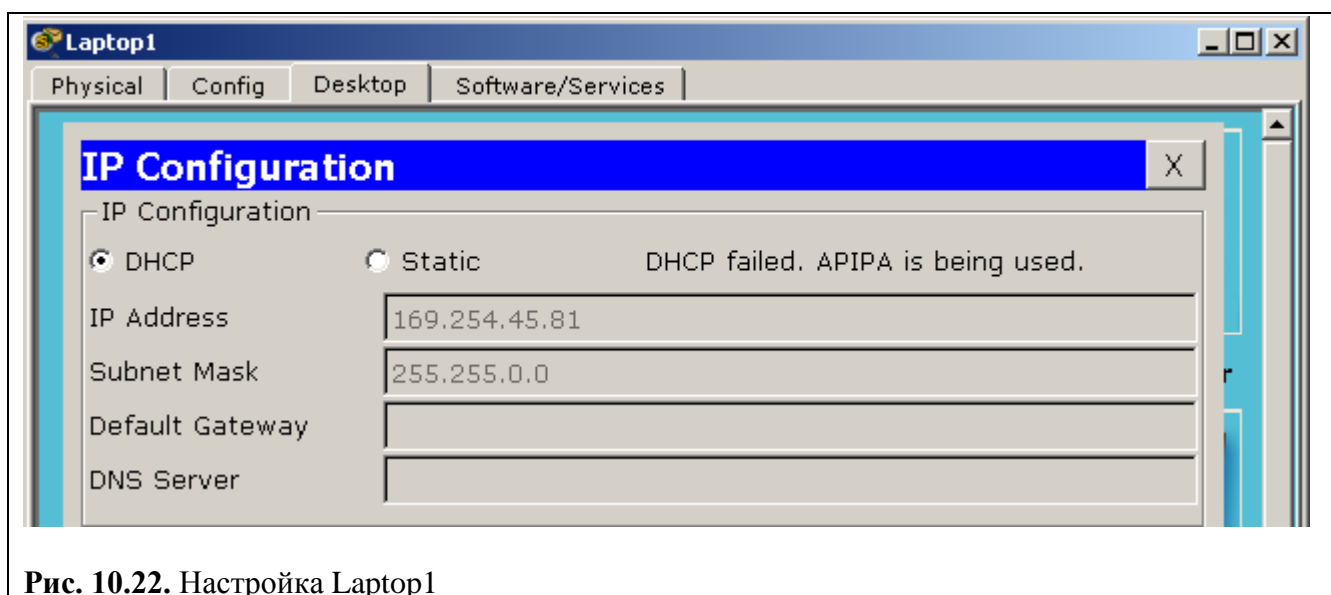


Рис. 10.22. Настройка Laptop1

**CopyLaptop1, Server2 настраиваем аналогично**

Для Laptop 2 укажем статический ip-адрес в той же сети.

Например: DHCP-адрес Laptop1 – 169.254.116.60 и маска -255.255.0.0. Получаем диапазон адресов 169.254.\*.\* (рис 10.23)

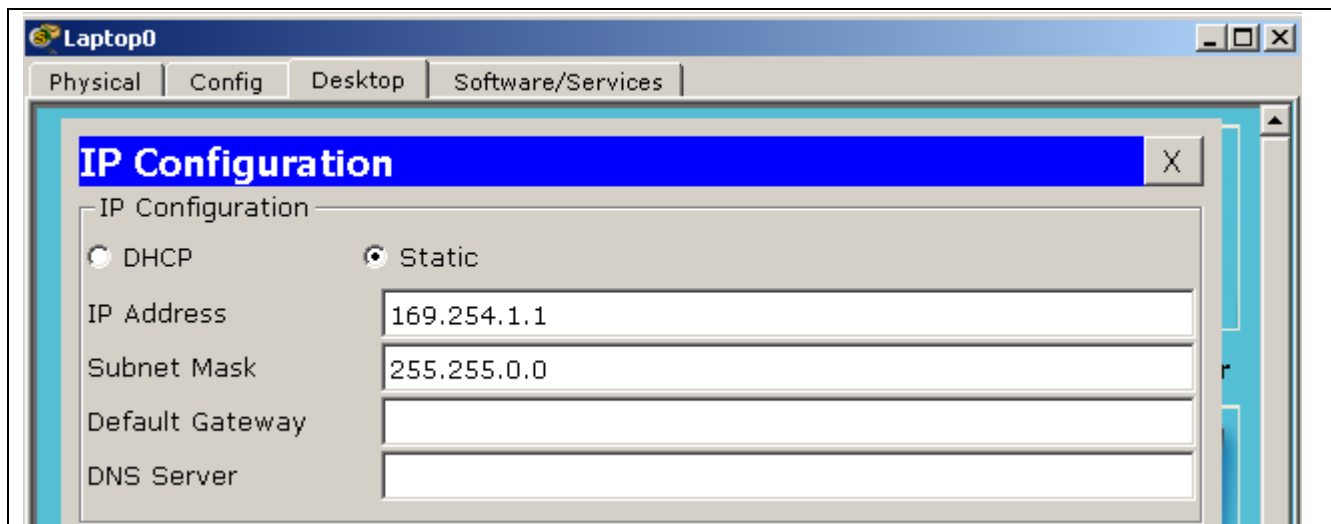


Рис. 10.23. Настройка Laptop0

**Server1 настраиваем аналогично**

Проверяем связь (т.е пингуем) между Laptop0 и Laptop1 (рис 10.24)

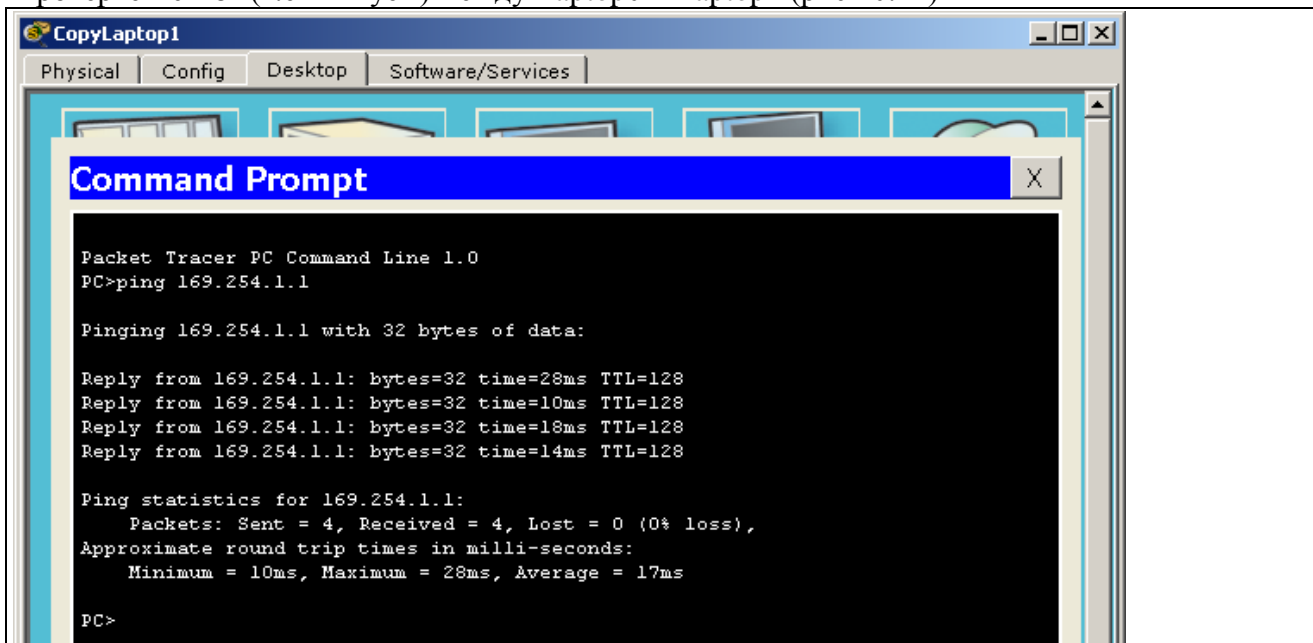


Рис. 10.24. Проверка связи между Laptop0 и Laptop1

**Проверяем связь аналогично между CopyLaptop1 и Laptop1**

Проверяем работает ли http-сервис на серверах Server0 и Server1 (10.25)



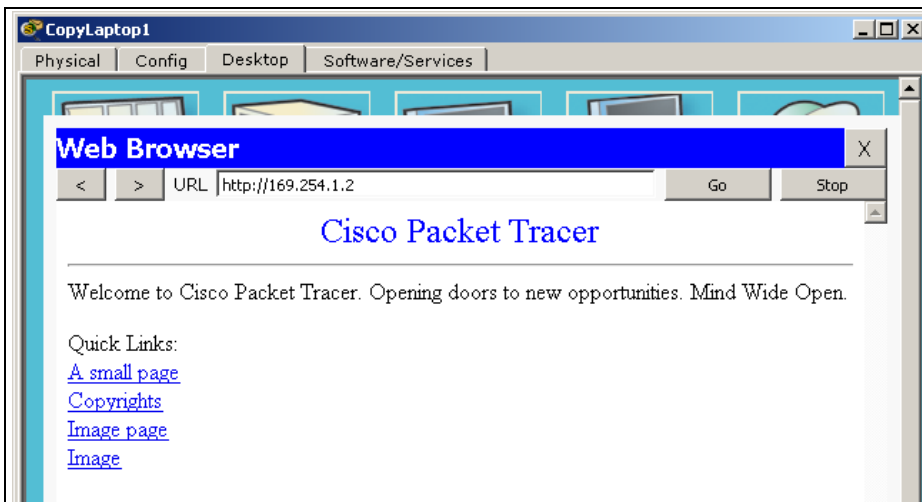


Рис. 10.25. Проверяем http-сервис на Server 1

#### Задание 4. Беспроводная сеть между офисами

Настроим следующую беспроводную сеть (рис. 10.26).

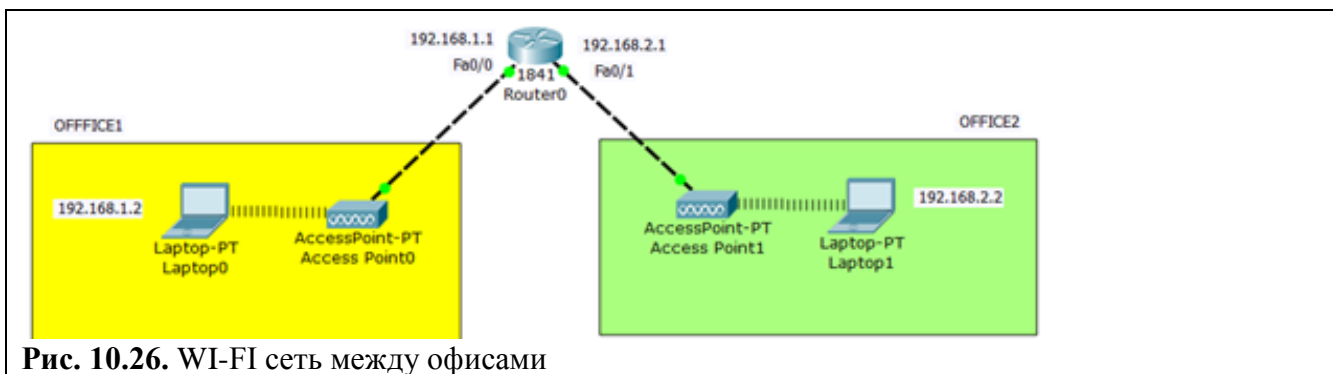


Рис. 10.26. WI-FI сеть между офисами

Снабжаем ноутбука wi-fi адаптерами WPC300N. Настройки обоих ноутбуков аналогичны (рис. 10.27).

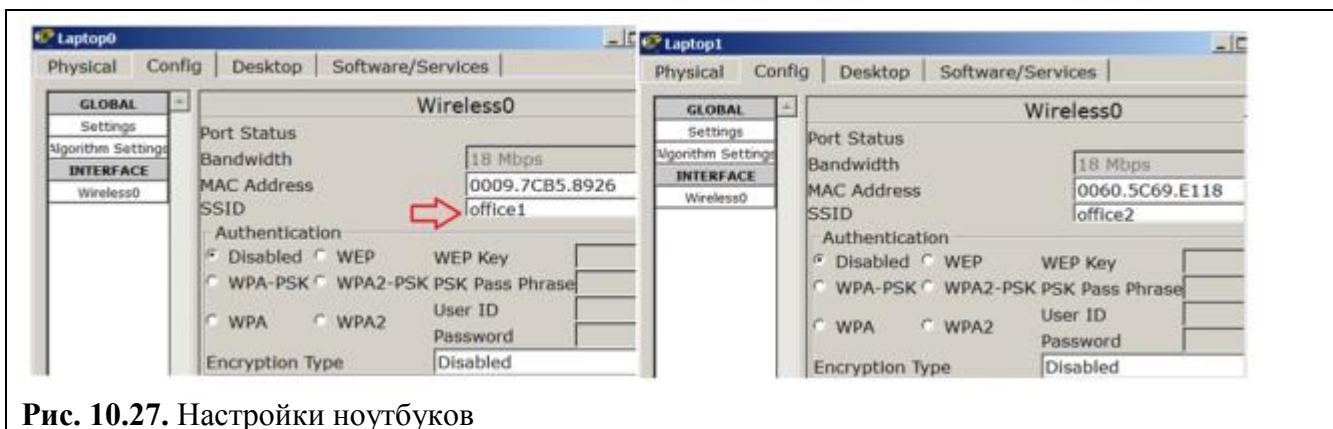


Рис. 10.27. Настройки ноутбуков

#### Примечание

Каждая беспроводная локальная сеть использует уникальное сетевое имя для идентификации сети. Это имя также называется идентификатором обслуживания сети - SSID. Когда вы будете устанавливать адаптер Wi-Fi, нужно будет указать SSID. Если вы хотите подключиться к существующей беспроводной сети, вы должны использовать имя этой сети. Имя может иметь длину до 32 символов и содержать буквы и цифры.

Помимо SSID на ноутбуках настраивается шлюз (рис. 10.28).



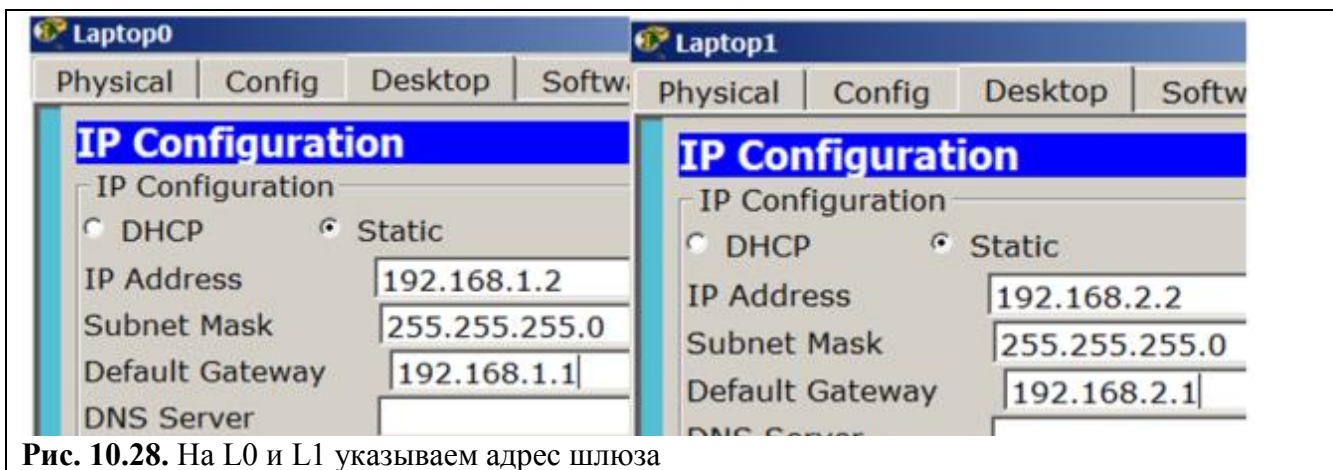


Рис. 10.28. На L0 и L1 указываем адрес шлюза

SSID задаем на обеих точка доступа (рис. 10.29).

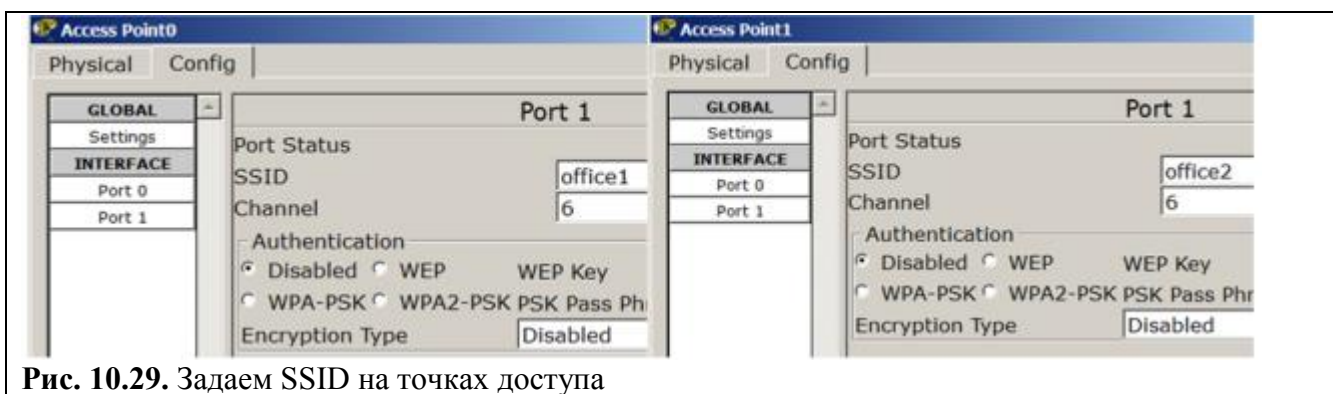


Рис. 10.29. Задаем SSID на точках доступа

Проверяем связь ПК из разных офисов (рис. 10.30).

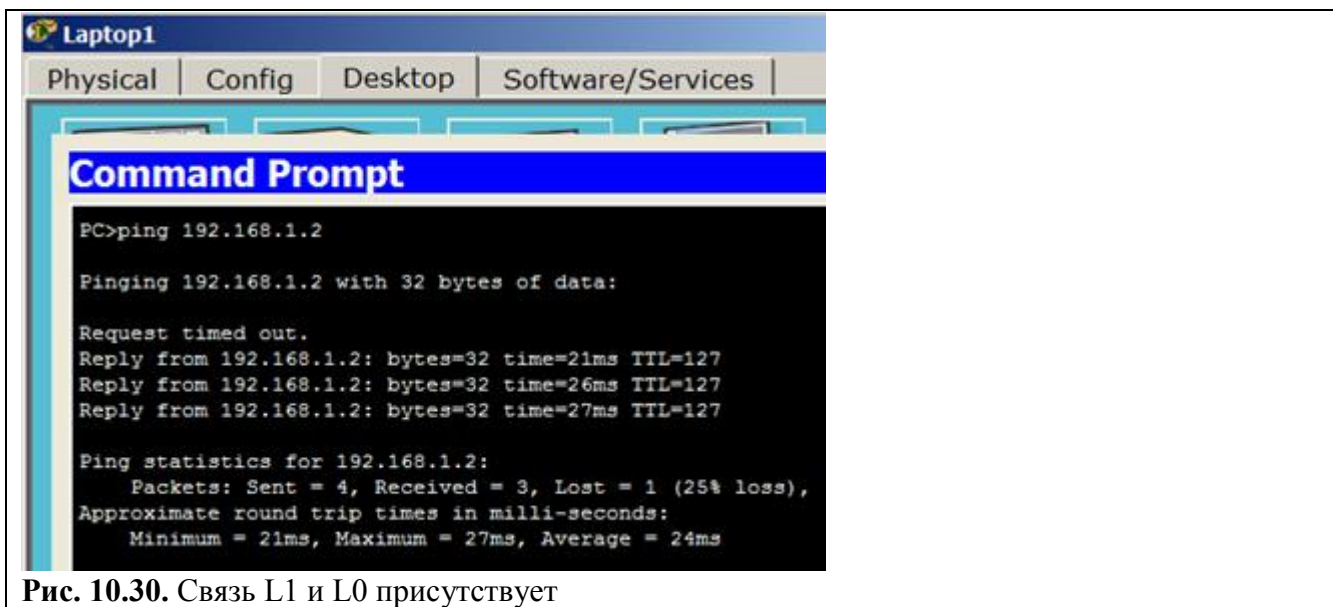


Рис. 10.30. Связь L1 и L0 присутствует

Форма представления и содержание отчета

Отчет в формате DOC или PDF отправляется в электронную среду Moodle вместе с файлом проекта. Отчет выполняется согласно варианту.

## Содержание отчета

1. Заголовок согласно приложению.
2. Цель работы.
3. Ход работы:
  - i. Данный раздел состоит из последовательного описания значимых выполняемых шагов (с указанием их сути) Пояснения работы команды ping и **содержимго протоколов**.
  - j. копий экранов (должна быть видна набранная команда и реакция системы, если она есть).
4. По каждому пункту практической работы должны быть приведены **ВЫВОДЫ**.

### **Практическая работа №12. «Основы IP — адресации. Классы сетей и структура адресов»**

**Цель работы:** Научится решать следующие задачи:

- Идентифицировать 5 различных классов IP-адресов.
- Описывать характеристики и использование классов IP-адресов.
- Определять класс IP-адреса исходя из его значения.
- Определять, какая часть IP-адреса идентифицирует сеть (network ID) и какая – хост (host ID)
- Определять допустимые и недопустимые IP- адреса хостов, исходя из правил адресации
- Определять диапазон адресов и маску подсети по умолчанию для каждого класса адресов

**Ход работы:**

**Задание 1.** Изучить теоретические основы IP-адресации

1. Сколько октетов в IP — адресе?
2. Сколько битов в октете?
3. Сколько бит в маске подсети?
4. В каких диапазонах десятичных и двоичных значений может быть значение первого октета IP-адресов класса «B»?  
Десятичные: от \_\_\_\_ до \_\_\_\_  
Двоичные: от \_\_\_\_\_ до \_\_\_\_\_
5. Какие октеты представляют сетевую часть IP-адреса класса «C»
6. Какие октеты представляют часть адреса хоста в IP-адресе класса «A»?
7. Какой из приведенных ниже адресов является примером широковещательного адреса для сети класса B?

	147.1.1.1
	147.255.255.255
	147.13.0.0
	147.14.255.255

8. Заполните таблицу:

Таблица 1 – характеристики классов IP адресации.

Класс адреса	Старшие биты первого октета	Диапазон дес. значений первого октета	Network / Host ID (N=Network, H=Host)	Маска подсети по умолч.	Количество сетей	Количество хостов (используемых адресов) в сети
A						
B						
C						
D						
E						

**Задание 2.** Определение частей IP- адресов.

1. Заполнить таблицу об идентификации различных классов IP-адресов.

Таблица 2 – идентификация различных классов IP-адресов

IP- адреса хостов	Класс адреса	Адрес сети	Адреса хостов	Широковещательный (broadcast) адрес	Маска подсети по умолчанию
216.14.55.137					
123.1.1.15					
150.127.221.244					
194.125.35.199					
175.12.239.244					

2. Разбить каждую сеть на две подсети, вычислить диапазон каждой, адрес подсети , подкаст.
3. Дан IP- адрес 142.226.0.15
  - Чему равен двоичный эквивалент второго октета?
  - Какому классу принадлежит этот адрес?
  - Чему равен адрес сети, в которой находится хост с этим адресом?
  - Является ли этот адрес хоста допустимым в классической схеме адресации

**Задание 3**

Найти адрес сети, минимальный IP, максимальный IP и число хостов по IP-адресу и маске сети:

IP-адрес: 192.168.215.89

Маска: 255.255.255.0 /24

**Задание 4**

Что произойдет с данными из 1 задачи, если маску сети изменить на 255.255.255.128

**Задание 5**

Найти маску сети, минимальный IP, максимальный IP по IP-адресу и адресу сети:

IP-адрес: 124.165.101.45

Сеть: 124.128.0.0

### **Задание 6**

Найти минимальный IP, максимальный IP по адресу сети и маске:

Маска: 255.255.192.0

Сеть: 92.151.0.0

### **Задание 7**

Найти адрес сети, минимальный IP, максимальный IP и число хостов по IP-адресу и маске сети:

IP-адрес: 85.45.5.33

Маска: 255.252.0.0

#### **Содержание отчета:**

1. Идентификация 5 различных классов IP-адресов – таблица 1;
2. описание характеристики и использование классов IP-адресов — таблица 2;
3. определение класса IP-адреса исходя из его значения;
4. определение допустимых и недопустимых IP- адресов хостов, исходя из правил адресации;
5. определение диапазона адресов и масок подсети по умолчанию для каждого класса адресов;
6. решения задач.

## Практическая работа №13. Механизм адресации в IP-сетях

**Цель работы** – изучить адресацию, общую классификацию адресов в стеке TCP/IP, принцип назначения IP-адресов узлам отдельных подсетей.

### **Задание на лабораторную работу:**

1. Составить конспект по всему материалу.
2. Выполнить все задания приведенные ниже.
3. Ответить на контрольные вопросы.

## **1 Типы адресов стека TCP/IP**

В стеке TCP/IP используются три типа адресов:

1. локальные (называемые также аппаратными)
2. IP-адреса
3. символьные доменные имена

### **1.1.1. Локальные адреса**

**Локальный адрес** в терминологии TCP/IP - это такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, которая сама является элементом составной интерсети.

В разных подсетях допустимы разные сетевые технологии, разные стеки протоколов, поэтому при создании стека TCP/IP уже заранее предполагалось наличие разных типов локальных адресов.

Если подсетью интерсети является локальная сеть, то локальный адрес - это MAC - адрес.

MAC - адрес назначается сетевым адаптерам и сетевым интерфейсам маршрутизаторов.

MAC - адреса назначаются производителями оборудования и являются уникальными, так как управляются централизованно.

Для всех существующих технологий локальных сетей MAC - адрес имеет формат 6 байт, например 11-АО-17-3D-BC-01.

Надо отметить, что поскольку протокол IP может работать и над протоколами более высокого уровня. В этом случае локальными адресами для протокола IP соответственно будут адреса соответствующих протоколов более высокого уровня.

Следует учесть, что компьютер в локальной сети может иметь несколько локальных адресов даже при одном сетевом адаптере. И наоборот, некоторые сетевые устройства вообще не имеют локальных адресов. Например, к таким устройствам относятся глобальные порты маршрутизаторов, предназначенные для соединений типа "точка-точка".

### **1.1.2. IP-адреса - основной тип адресов сетевого уровня.**

На основании IP-адресов сетевой уровень передает пакеты между сетями.

IP-адреса состоят из 4 байт.

IP-адрес назначается администратором во время конфигурирования компьютеров и маршрутизаторов.

IP-адрес состоит из двух частей: номера сети и номера узла.

Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (InternetNetworkInformationCenter, InterNIC), если сеть должна работать как составная часть Internet. Обычно поставщики услуг Internet получают диапазоны адресов у подразделений InterNIC, а затем распределяют их между своими абонентами.

Номер узла в протоколе IP назначается независимо от локального адреса узла!

Маршрутизатор по определению входит сразу в несколько сетей. Поэтому каждый порт маршрутизатора имеет собственный **IP-адрес**.

Конечный узел также может входить в несколько **IP-сетей**. В этом случае компьютер должен иметь несколько **IP-адресов**, по числу сетевых связей.

Таким образом, **IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение**. Напоминаю, что мы поговорим об этом немного позже более подробно.

### 1.1.3. Символьные имена

**Символьные имена** имеют символьный вид и в **IP-сетях** называются **доменными**.

**Доменные имена** строятся по иерархическому признаку. Полное **символьное имя в IP-сетях** состоит из нескольких составляющих, которые разделяются точкой. Они перечисляются в следующем порядке (слева-направо):

- сначала простое **имя конечного узла**
- затем **имя группы узлов** (например, имя организации)
- затем **имя более крупной группы (поддомена)**

И так до имени домена самого высокого уровня (например, домена объединяющего организации по географическому принципу: **UA** - Украина, **RU** - Россия, **UK** - Великобритания, **SU** - США)

Примеров **доменного имени** может служить имя **base2.sales.zil.ru**. Между **доменным именем** и **IP-адресом** узла **нет никакого соответствия**, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел интернета однозначно мог определяться в сети, как по **доменному имени**, так и по **IP-адресу**.

## 2. IP адреса. Классы IP адресов

Самое первое, что надо сразу уяснить - **IP-адреса** назначаются не узлам составной сети. **IP адреса** назначаются сетевым интерфейсам узлов составной сети.

Очень многие (если не большинство) компьютеров в **IP сети** имеют единственный сетевой интерфейс (и как следствие один **IP адрес**). Но компьютеры и другие устройства могут иметь несколько (если не больше) сетевых интерфейсов - и каждый интерфейс будет иметь свой собственный **IP адрес**.

Так устройство с 6 активными интерфейсами (например, маршрутизатор) будет иметь **6 IP адресов** - по одному на каждый интерфейс в каждой сети, к которой он подключен.

Итак, **IP адрес** определяет однозначно **сеть** и **узел**, который подключен к данной сети. **IP адрес** имеет длину **4 байта (8 бит)**, это дает в совокупности **32 бита** доступной информации.

Для улучшения читабельности, **IP адрес** записывается в виде четырех чисел, **разделенных точками**:

например, **128.10.2.30** - десятичная форма представления адреса - **4 (десятичных) числа, разделенных (.) точками**, а **10000000 00001010 00000010 00011110** - двоичная форма представления этого же адреса. **4-ре 8-ми разрядных числа (октета)**

*Так как каждое из четырех чисел - это десятичное представление 8-битного байта, то каждое число может принимать значения от 0 до 255 (что дает 256 уникальных значений - помните, ноль - это тоже величина).*

Здесь надо отметить:

Десятичная форма записи **IP-адреса** используется в основном при в операционных системах, как наиболее удобная при настройке.

Кроме двоичной формы, встречается шестнадцатеричная форма записи **IP-адреса**: **C0.94.1.3**

Для сведения: использование **32-разрядных** двоичных чисел позволяет создавать **4 294 967 296 уникальных IP-адресов** - более чем достаточно для любой частной **интрасети** (хотя сеть **Internet** скоро может начать испытывать нехватку уникальных адресов).

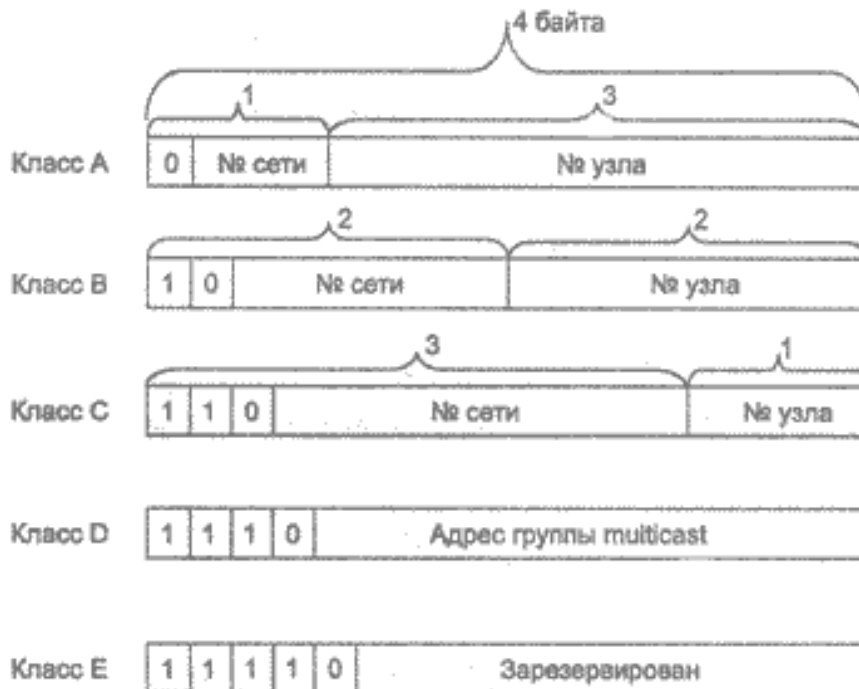
**IP адрес состоит из двух логических частей - номера сети и номера узла в сети.**

Конечно же, сразу возникает вопрос: а как определить в одном адресе, где **номер сети**, а где **номер узла**? Можно условиться использовать, например, первые **8 бит адреса** для **номера сети**, остальные для **номеров узлов** в той сети, или **первые 16 бит**, или **первые 24 бита**. Но в таком случае адресация получается абсолютно не гибкой, мы будем иметь или много маленьких сетей и мало больших, или наоборот.

Для того чтобы более рационально определиться с величиной сети и при том разграничить какая часть **IP-адреса** относится к **номеру сети**, а какая - к **номеру узла** условились использовать **систему классов**. Система классов использует значения **первых бит адреса**.

Но, таким образом, что значения этих **первых бит адреса** являются признаками того, к какому **классу** относится тот или иной **IP-адрес**.

#### Классы IP-адресов:



Итак, давайте в отдельной таблице приведем диапазоны **номеров сетей** и **максимальное число узлов**, соответствующих каждому **классу сетей**:

Класс	Первые биты	Наименьший адрес сети	Наибольший адрес сети	Максимальное количество узлов
<b>A</b>	<b>0</b>	1.0.0.0	126.0.0.0	2 <sup>24</sup> (16 777 216-2)
<b>B</b>	<b>10</b>	128.0.0.0	191.255.0.0	2 <sup>16</sup> (65536-2)
<b>C</b>	<b>110</b>	192.0.1.0	223.255.255.0	2 <sup>8</sup> (256-2)
<b>D</b>	<b>1110</b>	224.0.0.0	239.255.255.255	Multicast
<b>E</b>	<b>11110</b>	240.0.0.0	247.255.255.255	зарезервирован

**Сети класса C являются наиболее распространенными.**

- Если **адрес** начинается с последовательности **1110**, то он является адресом **класса D** и обозначает **особый, групповой адрес - multicast**.

Если в пакете в качестве **адреса назначения** указан **адрес класса D**, то **такой пакет должны получить все узлы**, которым присвоен **данный адрес**. Но об этом мы еще поговорим ниже.

- Если адрес начинается с последовательности **11110**, то это значит, что **данный адрес относится к классу E**. Адреса этого класса зарезервированы для будущих применений.

Таким образом, можно однозначно определить, что:

**Большие сети** получают адреса **класса A**, **средние - класса B**, а **маленькие - класса C**. В зависимости от того к какому классу (**A B C**) принадлежит адрес, **номер сети** может быть представлен **первыми 8, 16 или 24 разрядами**, а **номер хоста - последними 24, 16 или 8 разрядами**.

Такова традиционная **система классов**, но и она не достаточно гибко определяет границы между **номером сети** и **номером узла**. С использованием классов границы проходят по **границам байтов**. Существует другой метод, который может проводить разделение границы между **номером сети** и **номером узла** в одном **IP-адресе по границам битов!** Но всему свое время, и прежде чем, познакомится с этим способом, мы должны рассмотреть следующий, очень немаловажный момент, который касается "правил исключений" в **IP - адресации**.

### Особые IP-адреса

Существуют некоторые значения **IP-адресов**, которые **зарезервированы заранее**, то есть существуют **IP-адреса**, которые предназначены для **особых целей**. Для каких?

1) Если весь **IP-адрес** состоит **только из двоичных нулей**, то он обозначает **адрес того узла, который сгенерировал этот пакет;**

0 0 0 0 ..... 0 0 0 0

этот режим используется только в некоторых сообщениях **протокола межсетевых управляющих сообщений ICMP**.

2) Если в **поле номера сети** стоят только **нули**, то по умолчанию считается, что **узел назначения принадлежит той же самой сети**, что и **узел, который отправил пакет**.

0 0 0 0 .....0 Номер узла



**IP-адрес с нулевым номером хоста** используется для адресации ко всей сети. Например, в сети класса Сс номером **199.60.32** IP-адрес **199.60.32.0** обозначает сеть в целом.

3) Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета.

1 1 1 1 .....1 1
------------------

Такая рассылка называется **ограниченным широковещательным сообщением (limitedbroadcast)**.

4) Если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, пакет с адресом **192.190.21.255** доставляется всем узлам сети **192.190.21.0**.

Номер сети 1111.....11
------------------------

Такая рассылка называется **широковещательным сообщением (broadcast)**.

Предположим, например, что один из хостов в сети класса Сс сетевым адресом **199.60.32.0** собирается направить сообщение всем остальным хостам, находящимся в той же сети. В этом случае сообщение должно быть передано на адрес **199.60.32.255**.

При адресации хостов интерсети администратор должен обязательно учитывать все ограничения, которые вносятся **особым назначением некоторых IP-адресов**.

Таким образом, каждый администратор должен знать, что **ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей**. Отсюда следует, что **максимальное количество узлов**, приведенное в таблице для сетей каждого класса, **на практике должно быть уменьшено на 2**.

**Особый смысл имеет IP-адрес, первый октет которого равен 127**. Этот адрес зарезервирован для тестирования программ и взаимодействия процессов в пределах одной машины.

Когда программа посылает данные по IP-адресу **127.0.0.1**, то образуется как бы "петля".

Данные не передаются по сети, а возвращаются модулям верхнего уровня, как только что принятые.

Поэтому в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со **127!** Этот адрес имеет название **loopback**.

Можно отнести адрес **127.0.0.0** ко внутренней сети модуля маршрутизации узла, а адрес **127.0.0.1** - к адресу этого модуля на внутренней сети.

На самом деле любой адрес сети **127.0.0.0** служит для обозначения своего модуля маршрутизации, а не только **127.0.0.1**, например **127.0.0.3**.

В протоколе IP нет понятия широковещательности в том смысле, в котором оно используется в протоколах **канального уровня** локальных сетей, когда данные должны быть доставлены абсолютно всем узлам.

Как **ограниченный широковещательный IP-адрес**, так и **широковещательный IP-адрес** имеют свои пределы распространения в интерсети.

- они ограничены либо сетью, к которой принадлежит **узел-источник пакета**, либо сетью, **номер которой указан в адресе назначения**. Поэтому деление сети с помощью маршрутизаторов на части локализует широковещательный шторм пределами одной из составляющих общую сеть частей просто потому, что **нет способа адресовать пакет одновременно всем узлам всех сетей составной сети**.

Нами уже упоминалась выше в таблице форма **группового IP-адреса - multicast**. Так вот именно **IP адрес multicast** означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса.

Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Члены какой-либо **группы multicast** не обязательно должны принадлежать одной сети. В общем случае они могут распределяться по совершенно различным сетям, находящимся друг от друга на произвольном количестве хопов.

**Групповой адрес не делится на поля номера сети и узла и обрабатывается маршрутизатором особым образом.**

Основное назначение **multicast-адресов** - распространение информации по схеме "**один-ко-многим**".

Она работает следующим образом: хост, который хочет передавать одну и ту же информацию многим абонентам, с помощью **специального протокола IGMP (Internet Group Management Protocol)** сообщает о создании в сети новой **мультивещательной группы** с определенным адресом.

Маршрутизаторы, поддерживающие **мультивещательность**, распространяют информацию о создании новой группы в сетях, подключенных к портам этого маршрутизатора.

Хосты, которые хотят присоединиться к вновь создаваемой **мультивещательной группе**, сообщают об этом своим локальным маршрутизаторам и те передают эту информацию хосту, инициатору создания новой группы.

Чтобы маршрутизаторы могли автоматически распространять пакеты с адресом **multicast** по составной сети, необходимо использовать в конечных маршрутизаторах специальные **модифицированные протоколы обмена маршрутной информацией**.

В общем, **групповая адресация** была предназначена для экономичного распространения в **Internet** или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей.

Надо сказать, что если такие средства найдут широкое применение (сейчас они представляют в основном небольшие экспериментальные островки в общем **Internet**), то **Internet** сможет создать серьезную конкуренцию радио и телевидению.

Ну что ж, давайте, сделаем итог, который закрепит наше представление о том, что означает **IP-адрес**:

**IP адрес** может означать одно из трех:

1. **Адрес IP сети** (группа **IP** устройств, имеющих доступ к общей среде передаче - например, все устройства в сегменте **Ethernet**). **Сетевой адрес** всегда имеет биты интерфейса (**хоста**) адресного пространства **установленными в 0** (если сеть не разбита на подсети - как мы еще увидим);

2. **Широковещательный адрес IP сети** (адрес для 'разговора' со всеми устройствами в IP сети). Широковещательные адреса для сети всегда имеют **хостовые биты адресного пространства установленными в 1** (если сеть не разбита на подсети - опять же, как мы вскоре увидим).

3. **Адрес интерфейса** (например Ethernet-адаптер или PPP интерфейс хоста, маршрутизатора, сервера печать итд). Эти адреса могут иметь любые значения **хостовых битов, исключая все нули или все единицы** - чтобы не путать с адресами сетей и широковещательными адресами.

Для сети класса А ...

(один байт под адрес сети, три байта под номер хоста)

**10.0.0.0** сеть класса А, потому что **все хостовые биты равны 0**.

**10.0.1.0** адрес хоста в этой сети

**10.255.255.255** широковещательный адрес этой сети, поскольку все сетевые биты установлены в **1**

Для сети класса В...

(два байта под адрес сети, два байта под номер хоста)

**172.17.0.0** сеть класса В

**172.17.0.1** адрес хоста в этой сети

**172.17.255.255** сетевой широковещательный адрес

Для сети класса С...

(три байта под адрес сети, один байт под номер хоста)

**192.168.3.0** адрес сети класса С

**192.168.3.42** хостовый адрес в этой сети

**192.168.3.255** сетевой широковещательный адрес

Едва ли не все доступные сетевые IP адреса принадлежат классу С.

### 3. Маски в IP адресации

Итак, рассмотрена традиционная схема деления IP-адреса на номер сети, и номер узла, которая основана на понятии класса. Класс определяется значениями нескольких первых бит адреса. Теперь, например, можно определить, что поскольку первый байт адреса **185.23.44.206** попадает в диапазон **128-191**, то этот адрес относится к классу В, а значит, номером сети являются первые два байта, дополненные двумя нулевыми байтами - **185.23.0.0**, а номером узла - **0.0.44.206**.

Очевидно, что определение номеров сети по первым байтам адреса также не вполне гибкий механизм для адресации. А что если использовать какой-либо другой признак, с помощью которого можно было бы более гибко устанавливать границу между номером сети и номером узла?

В качестве такого признака сейчас получили широкое распространение маски.

Маска - это тоже 32-разрядное число, она имеет такой же вид, как и IP-адрес. Маска используется в паре с IP-адресом, но не совпадает с ним.

Принцип отделения номера сети и номера узла сети с использованием маски состоит в следующем:

Двоичная запись маски содержит *единицы* в тех разрядах, которые в IP-адресе должны представляться как номер сети и *нули* в тех разрядах, которые представляются как номер хоста.

Каждый класс IP-адресов (А, В и С) имеет свою маску, используемую по умолчанию.

Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность.

Таким образом, для стандартных классов сетей маски имеют следующие значения:

- класс А - 11111111.00000000.00000000.00000000 (255.0.0.0) ;
- класс В - 11111111.11111111.00000000.00000000 (255.255.0.0) ;
- класс С - 11111111.11111111.11111111.00000000 (255.255.255.0) .

Например:

Если адресу 185.23.44.206 назначить маску 255.255.255.0, то смотрим, что единицы в маске заданы в трех байтах, значит номер сети будет 185.23.44.0, а не 185.23.0.0, как это определено правилами системы классов.

Для записи масок используются и другие форматы, например, удобно интерпретировать значение маски, записанной в шестнадцатеричном коде:

**FF.FF.00.00** - маска для адресов класса В.

Часто встречается и такое обозначение: IP-адрес/префикс сети. Например, 185.23.44.206/16 - эта запись говорит о том, что маска для этого адреса содержит 16 единиц (префикс сети), или что в указанном IP-адресе под номер сети отведено 16 двоичных разрядов.

**Нотация с префиксом сети также известна как бесклассовая междоменная маршрутизация (Classless Interdomain Routing -CIDR).**

Таким образом, очень легко, снабжая каждый IP-адрес произвольной маской (не обязательно кратной 8), отказаться от понятий классов адресов и тем самым сделать более гибкой систему IP адресации.

Рассмотрим пример: для IP-адреса 129.64.134.5 назначим маску 255.255.128.0, что в двоичном виде будет выглядеть так:

**IP-адрес 129.64. 134.5 - 10000001.01000000.1 0000110.00000101**

**Маска 255.255.128.0 - 11111111.11111111.1 0000000.00000000**

Здесь 17 последовательных единиц в маске, "накладываются" на IP-адрес, и определяют номер сети : 10000001. 01000000. 10000000. 00000000 или 129.64.128.0,

а номер узла 0000110.00000101 или 0.0.6.5.

Механизм масок очень широко распространен в IP-маршрутизации, причем маски могут использоваться для самых разных целей. С их помощью администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей!

На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых "префиксов" с целью

уменьшения объема **таблиц маршрутизации**, и повышения за счет этого производительности маршрутизаторов. (Создание надсетей).

Маски при записи всегда "неразлучны" с соответствующими адресами, **IP-адрес маска подсети** - именно так теперь и мы будем описывать адрес любого хоста сети.

#### 4. Порядок назначения IP адресов. Автономные IP адреса. Автоматизация назначения IP адресов

**Номера сетей** могут назначаться либо **централизованно**, если сеть является частью **Internet**, либо **произвольно**, если сеть работает **автономно**. **Номера узлов** и в том и в другом случае администратор назначает самостоятельно по своему усмотрению, не выходя, разумеется, из разрешенного для этого класса сети диапазона.

Главную роль в **централизованном распределении IP-адресов** до некоторого времени играла организация **InterNIC** (NetworkInformationCenter), однако с ростом сети задача распределения адресов стала слишком сложной. **InterNIC** делегировала часть своих функций другим организациям и крупным поставщикам услуг **Internet** - **провайдерам**. В частности распределением **IP-адресов** для подключения к сети Internet теперь занимаются **провайдеры**.

С тех пор, как появилась и стала широко распространяться сеть **Internet**, уже прошло не мало времени. И теперь уже становится актуальным вопрос о **дефиците IP-адресов**. Если говорить о реальной обстановке при распределении адресов для пользователей **Internet**, то сейчас очень трудно получить адрес **класса В** и уже практически невозможно стать обладателем адреса **класса А**! При этом всем надо сказать, что **дефицит IP-адресов** вызван не совсем постоянным ростом сетей, а просто нерациональным их использованием. Очень часто владельцы сети **класса С** расходуют лишь небольшую часть из имеющихся у них **254 адресов**.

Рассмотрим пример, когда две сети необходимо соединить глобальной связью.

В таких случаях в качестве канала связи используют два маршрутизатора, соединенных по схеме "точка-точка".



В ситуации, которая приведена в примере, для вырожденной сети, образованной каналом, связывающим порты двух смежных маршрутизаторов, приходится выделять **отдельный номер сети**, хотя в этой сети имеются всего **2 узла**.

Давайте рассмотрим другую ситуацию: какие **IP-адреса** может использовать администратор, если провайдер услуг **Internet** не назначил ему никакого адреса? Если, к примеру, мы точно знаем, что сеть, которую мы администрируем никогда в будущем не будет подключаться к **Internet** (работает в "**автономном режиме**"), тогда мы можем использовать любые **IP-адреса**, соблюдая правила их назначения, о которых шла речь выше. Для простоты

можно использовать адреса **класса С**: в этом случае не придется вычислять значение **маски** подсети и вычислять адрес для каждого хоста.

В этом случае мы должны будем просто назначить каждому сегменту нашей локальной сети его собственный сетевой номер **класса С**.

Если все сегменты нашей локальной сети имеют собственные сетевые номера **класса С**, то в каждом сегменте можно создать по **254** номера хостов.

Однако если у нас есть хотя бы небольшая вероятность того, что когда-либо в будущем наша сеть может быть подключена к **Internet**, не следует использовать такие **IP-адреса**! Они могут привести к конфликту с другими адресами в **Internet**. Чтобы избежать таких конфликтов, нужно использовать **IP-адреса, зарезервированные для частных сетей**.

Для этой цели зарезервированы специально несколько блоков **IP-адресов**, которые называются **автономными**.

### Автономные IP адреса

**Автономные адреса** зарезервированы для использования **частными сетями**. Они обычно используются организациями, которые имеют свою частную большую сеть - **intranet** (локальные сети с архитектурой и логикой **Internet**), но и маленькие сети часто находят их полезными.

**Эти адреса не обрабатываются маршрутизаторами Internet, ни при каких условиях.** Эти адреса выбраны из разных классов.

Класс	От IP-адреса	До IP-адреса	Всего узлов адресов в диапазоне
<b>A</b>	<b>10.0.0.0</b>	<b>10.255.255.255</b>	16 777 216-2
<b>B</b>	<b>172.16.0.0</b>	<b>172.31.255.255</b>	65 536-2
<b>C</b>	<b>192.168.0.0</b>	<b>192.168.255.255</b>	256-2

Эти адреса являются зарезервированными для **частных сетей**. Таким образом, если в будущем мы решим все-таки подключить свою сеть к **Internet**, то даже если трафик с одного из хостов в нашей сети и попадет каким-либо образом в **Internet**, конфликта между адресами произойти не должно. Маршрутизаторы в **Internet** запрограммированы так, чтобы не транслировать сообщения, направляемые с зарезервированных адресов или на них.

Надо сказать, что использование **автономных IP-адресов** имеет и недостатки, которые состоят в том, что если мы будем подключать свою сеть к **Internet**, то нам придется заново настроить конфигурацию хостов, соединяемых с **Internet**.

Можно сказать, что **подсеть** - это метод, состоящий в том, чтобы взять **сетевой IP адрес** и **локально** разбить его так, чтобы этот **один сетевой IP адрес** мог в действительности **использоваться в нескольких взаимосвязанных локальных сетях**.

**Один сетевой IP адрес может использоваться только для одной сети! Самое важное:** разбиение на **подсети** - это **локальная настройка**, она не видна "снаружи". Разбиение одной большой сети на **подсети**, значительно разгружает общий трафик и позволяет повысить безопасность всей сети в целом.

### Алгоритм разбиения сети на подсети

- 1) Устанавливаем физические соединения (сетевые кабели и сетевые соединители - такие как маршрутизаторы);
- 2) Принимаем решение насколько большие/маленькие **подсети** вам нужны, исходя из количества устройств, которое будет подключено к ним, то есть, сколько **IP адресов** требуется использовать в каждом сегменте сети.
- 3) Вычисляем соответствующие **сетевые маски и сетевые адреса**;
- 4) Раздаем каждому интерфейсу в каждой сети свой **IP адрес** и соответствующую **сетевую маску**;
- 5) Настраиваем каждый маршрутизатор и все сетевые устройства;
- 6) Проверяем систему, исправляем ошибки.

Сейчас наша задача разобраться с тем, как выполнить 2-й и 3-й шаги.

### Пример 1

Предположим, что мы хотим разбить нашу сеть на подсети, но имеем только один **IP-адрес сети 210.16.15.0**.

**Решение:**

**IP-адрес 210.16.15.0** - это адрес **класса С**. Сеть **класса С** может иметь до **254** интерфейсов (хостов) плюс адрес **сети (210.16.15.0)** и **широковещательный адрес (210.16.15.255)**.

Первое: определить "**размер**" подсети.

Существует зависимость между количеством создаваемых **подсетей** и "потраченными" **IP адресами**.

Каждая отдельная **IP сеть** имеет два адреса, неиспользуемые для интерфейсов (**хостов**):

- **IP адрес собственно сети и широковещательный адрес.**

При разбивке на подсети **каждая подсеть требует свой собственный уникальный IP адрес сети и широковещательный адрес** - и они должны быть корректно выбраны из диапазона адресов **IP сети**, которую мы делим на подсети.

Итак, если при разбивке **IP сети** на подсети, в каждой из которых есть **два сетевых адреса** и **два широковещательных адреса** - надо помнить, что каждая из них уменьшит количество используемых интерфейсных (хостовых) адресов на два.

Это мы должны всегда учитывать при вычислении **сетевых номеров**. Следующий шаг - **вычисление маски подсети и сетевых номеров**.

**Сетевая маска** - это то, что выполняет все логические манипуляции по разделению **IP сети на подсети**.

Для всех трех классов **IP сетей** существуют стандартные сетевые **маски**:

- **Класс А** (8 сетевых битов) :**255.0.0.0**
- **Класс В** (16 сетевых битов): **255.255.0.0**
- **Класс С** (24 сетевых бита): **255.255.255.0**

Чтобы создать **подсеть**, нужно изменить маску подсети для данного класса адресов.

**Номер подсети** можно задать, позаимствовав нужное для нумерации подсетей количество разрядов в номере хоста. Для этого берутся левые (старшие) разряды из номера хоста, в маске же взятые разряды заполняются единицами, чтобы показать, что эти разряды теперь нумеруют не узел а подсеть. Значения в остающихся разрядах маски подсети оставляются равными **нулю**; это означает, что оставшиеся разряды в номере хоста в **IP-адресе** должны использоваться как **новый (меньший) номер хоста**.

Например, чтобы **разбить сетевой адрес** на **две подсети**, мы должны позаимствовать **один хостовый бит**, установив **соответствующий бит в сетевой маске первого хостового бита в 1**.

Если нам нужно **четыре подсети** - используем **два хостовых бита**, если **восемь подсетей** - **три бита** и т.д. Однозначно, что если нам нужно **пять подсетей**, то мы будем использовать **три хостовых бита**. Соответствующим образом изменяется и **маска подсети**:

Для адресов **класса С**, при разбиении на **2 подсети** это дает **маску** -

**11111111.11111111.11111111.10000000** или **255.255.255.128**

при разбиении на **4 подсети** маска в двоичном виде -

**11111111.11111111.11111111.11000000**, или в десятичном **255.255.255.192**. и т.д.

Для нашего адреса сети класса С **210.16.15.0**, можно определить следующих несколько способов разбивки на **подсети**: -

Число подсетей	Число хостов	Сетевая маска
2	126	255.255.255.128 (11111111.11111111.11111111.10000000)
4	62	255.255.255.192 (11111111.11111111.11111111.11000000)
8	30	255.255.255.224 (11111111.11111111.11111111.11100000)
16	14	255.255.255.240 (11111111.11111111.11111111.11110000)
32	6	255.255.255.248 (11111111.11111111.11111111.11111000)
64	2	255.255.255.252 (11111111.11111111.11111111.11111100)

Теперь нужно решить вопрос об **адресах сетей и широковещательных адресах**, и о диапазоне **IP адресов** для каждой из этих сетей.

Снова, принимая во внимание только сетевые адреса **класса С**. и показав только последнюю (хостовую) часть адресов, мы имеем:

Сетевая маска	Подсети	Сеть	Broadcast	MinIP	MaxIP	Хосты	Всего хостов
128	2	0	127	1	126	126	
		128	255	129	254	126	252



192	4	0	63	1	62	62	248
		64	127	65	126	62	
		128	191	129	190	62	
		192	255	193	254	62	
224	8	0	31	1	30	30	240
		32	63	33	62	30	
		64	95	65	94	30	
		96	127	97	126	30	
		128	159	129	158	30	
		160	191	161	190	30	
		192	223	193	222	30	
		224	255	225	254	30	

Из этой таблицы сразу можем увидеть, что **увеличение количества подсетей сокращает общее количество доступных хостовых адресов**. Теперь, вооруженные этой информацией, мы готовы назначать **хостовые и сетевые IP адреса и сетевые маски**.

## Пример 2

Определим, сколько нужно подсетей для нашей сети **класса С**, чтобы разбить ее на подсети по **10 хостов** в каждой.

### Решение:

Сеть **класса С** может обслуживать всего **254** хоста плюс адрес сети и широковещательный адрес.

Для адресации **10-ти хостов 3-х разрядов** недостаточно, поэтому необходимо **4-е разряда**. Итак, из восьми возможных для класса **С**, нам нужно только **4 разряда** для адресации **10 хостов**, остальные можно использовать как сетевые для адресации **подсетей**. Мы уже знаем, что каждая подсеть уменьшает количество возможных хостовых адресов в два раза.

Для адресации **16 подсетей** необходимо использовать **4 разряда**. Итак, посчитаем теперь количество узлов в каждой из **16 подсетей**:  $2^4 - 2 = 14$  хостов. Это количество с запасом удовлетворяет условие задачи.

Вычислим **маску подсети**, в этом случае она имеет вид:

**11111111.11111111.11111111.11110000** или

**255.255.255.240**

Мы должны будем указать эту маску при настройке конфигурации каждого хоста в нашей сети (независимо от того, в какой подсети находится хост).

Теперь, например, мы можем сказать, адрес **192.168.200.246** с маской **255.255.255.240** - означает номер сети **192.168.200.240** и номер узла **0.0.0.6**.

## Пример 3

Теперь, для всех трех классов определим соответственно маски подсети, и максимальное количество узлов возможное в каждой из этих подсетей, если необходимо разбить соответственно сеть **класса А**, сеть **класса В**, сеть **класса С** на отдельные **4 подсети**.

### Решение:

Для сети **класса А**:

Максимальное количество узлов **16 777 216**. Для адресации 4-х подсетей необходимо **2** разряда, значит остается **22 разряда** для адресации **хостов**. Таким образом, каждая из четырех подсетей способна обслуживать  $2^{22} - 2 = 4\ 194\ 302$  хоста в каждой из подсетей.

Число подсетей	Число хостов	Сетевая маска
4	4 194 302	255.192.0.0 (11111111. 11000000.00000000.00000000)

Для сети **класса В**

Максимальное количество узлов - **65 536**. Для адресации **4-х** подсетей в сетевом адресе **класса В** также нужно использовать **2 разряда**, но теперь свободными остается **14 разрядов**. Таким образом, каждая из подсетей может обслуживать  $2^{14} - 2 = 16\ 382$  хостов.

Число подсетей	Число хостов	Сетевая маска
4	16 382	255.255.192.0 (11111111.11111111. 11000000.00000000)

Пример с сетью **класса С** мы уже рассматривали. Итак, теперь самое главное уметь в **двоичном виде** читать **IP адреса**, а с помощью маски легко можно определить **номер сети** и **номер узла**. Вот теперь, можно сказать, теория заканчивается, для нашей работы очень важно "окрепнуть" в навыках работы с **IP адресами**, уметь разделять сети на **подсети**, вычислять **маски подсети**, и назначать возможные **адреса сетей**, и **адреса хостов** - это прямая обязанность сетевых администраторов.

Надо сказать, что назначение **IP-адресов** узлам сети даже при не очень большом размере сети представляет для администратора очень утомительную процедуру. Поэтому сразу вторым шагом в **IP адресации** разработчики решили автоматизировать этот процесс.

С этой целью был разработан протокол **DynamicHostConfigurationProtocol (DHCP)**, который освобождает администратора от этих проблем, **автоматизируя процесс назначения IP-адресов**.

**DHCP** может поддерживать способ **автоматического динамического распределения адресов**, а также **более простые способы ручного и автоматического статического назначения адресов**. Протокол **DHCP** работает в соответствии с моделью **клиент-сервер**.

Во время старта системы компьютер, являющийся **DHCP-клиентом**, посылает в сеть **широковещательный запрос** на получение **IP-адреса**. **DHCP - сервер** откликается и посылает сообщение-ответ, содержащее **IP-адрес**. Предполагается, что **DHCP-клиент и DHCP-сервер находятся в одной IP-сети**.

При **динамическом** распределении адресов **DHCP-сервер** выдает адрес клиенту на **ограниченное время**, оно называется **временем аренды (leaseduration)** . Это дает

возможность впоследствии повторно использовать этот **IP-адрес** для назначения **другому** компьютеру.

**Основное преимущество DHCP - автоматизация рутинной работы администратора по конфигурированию стека TCP/IP на каждом компьютере.** Иногда **динамическое** разделение адресов позволяет строить **IP-сеть**, количество узлов которой превышает количество имеющихся в распоряжении администратора **IP-адресов**.

В **ручной процедуре** назначения **статических адресов** активное участие принимает администратор, который предоставляет **DHCP - серверу** информацию о **соответствии IP-адресов физическим адресам** или другим идентификаторам клиентов. **DHCP-сервер**, пользуясь этой информацией, всегда выдает определенному клиенту назначенный администратором адрес.

При **автоматическом статическом** способе **DHCP-сервер** присваивает **IP-адрес** из пула **наличных IP-адресов** без вмешательства оператора. А границы пула **назначаемых адресов** задает администратор при конфигурировании **DHCP-сервера**.

Адрес дается клиенту из пула в **постоянное пользование**, то есть с **неограниченным сроком аренды**. Между **идентификатором клиента** и его **IP-адресом** по-прежнему, как и при **ручном** назначении, существует постоянное соответствие. Оно устанавливается в момент **первого** назначения **DHCP-сервером IP-адреса** клиенту. При всех последующих запросах сервер возвращает тот же самый **IP-адрес**.

**DHCP** обеспечивает надежный и простой способ конфигурации сети **TCP/IP**, **гарантируя отсутствие дублирования адресов за счет централизованного управления их распределением**.

Администратору в этом случае остается только управлять **процессом назначения адресов** с помощью параметра "**продолжительность аренды**", которая определяет, как долго компьютер может использовать назначенный **IP-адрес**, перед тем как снова запросить его от **DHCP-сервера** в аренду.

### Задания

- 1) **IP-адрес 190.235.130.N**(где N-номер варианта согласно таблице, данной ниже), сетевая маска **255.255.192.0**. Определите, **адрес сети** и **адрес узла**.
- 2) Определите **маски подсети** для случая разбиения сети с номером **192.0.0.0** на **32 подсети**.
- 3) Существует единая корпоративная сеть, количество узлов сети - **50 450**. Этой сети выделен адрес для выхода в **Internet 192.124.0.0**. Вы решили не требовать от провайдера дополнительных адресов и организовать **8 филиалов** в этой сети. Спрашивается:  
- Какое максимальное количество узлов может быть в каждом из филиалов? Вычислите **сетевые маски** и возможный диапазон **адресов хостов** для каждого из филиалов.
- 4) Вы являетесь администратором корпоративной сети из **6 подсетей**, в каждой подсети по 25 компьютеров. Необходимо используя один номер сети **класса C 192.168.10.0**, определить правильно ли выбран **размер подсети**, и назначить маски и возможные **IP-адреса** хостам сети.
- 5) Разделить IP-сеть на подсети в соответствии с вариантом из таблицы. Для каждой подсети указать широковещательный адрес.


Таблица 5.

Вариант	Сеть	Подсети
1.	192.168.16.0/24	5 подсетей с 100, 20, 10, 6 и 40 узлами
2.	194.45.27.0/24	5 подсетей с 34, 20, 62, 10 и 40 узлами
3.	56.1.1.0/16	4 подсети с 65, 22, 10 и 30 узлами
4.	147.168.0.0/16	5 подсетей с 56, 16, 10 и 70 узлами
5.	193.68.61.0/24	5 подсетей с 100, 20, 10 и 40 узлами
6.	192.100.0.0/24	4 подсети с 80, 20, 12 и 20 узлами
7.	195.18.11.0/24	4 подсети с 110, 11, 10 и 40 узлами
8.	207.15.0.0/24	4 подсети с 28, 80, 10 и 40 узлами
9.	222.11.0.0/24	4 подсети с 110, 20, 10 и 50 узлами
10.	200.2.2.0/24	4 подсети с 100, 20, 10 и 40 узлами
11.	201.111.32.0/16	5 подсетей с 170, 590, 1500, 800 и 254 узлами
12.	128.200.1.0/16	5 подсетей с 115, 300, 200, 128 и 420 узлами
13.	53.11.0.0/16	5 подсетей с 165, 222, 128, 110 и 430 узлами
14.	146.77.0.0/16	5 подсетей с 550, 116, 200, 256 и 170 узлами
15.	194.54.45.0/24	4 подсети с 103, 39, 10 и 16 узлами
16.	142.51.0.0/16	4 подсети с 180, 120, 12 и 30 узлами
17.	43.0.0.0/16	4 подсети с 151, 211, 16 и 70 узлами

### Контрольные вопросы

1. Какие бывают классы IP-адресов.
2. Как по первому байту адреса определить его класс?
3. Что такое маска, на что она указывает?
4. Для чего нужны маски переменной длины?
5. Изложите алгоритм деления сетей на подсети с помощью VLM (variable length mask).

### Содержание отчета:

 В отчете запишите ответы на задания практической работы согласно варианту. Обоснуйте каждый шаг получения результата, аналогично тому, как это сделано в примерах.

**Практическая работа №14. Сетевые утилиты и их использование. Утилиты ipconfig, ping и tracert.**

#### 1. ЦЕЛЬ РАБОТЫ

Определение настроек для подключения к локальной сети и к сети Internet с использованием утилиты ipconfig. Исследование вероятностно-временных характеристик фрагментов сети Internet с использованием утилиты ping. Исследование топологии фрагментов сети Internet с использованием утилиты tracert.

#### 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ

## 2.1. Адресация в IP-сетях

### 2.1.1. Типы адресов: физический (MAC-адрес), сетевой (IP-адрес) и символьный (DNS-имя)

Каждый компьютер в сети TCP/IP имеет адреса трех уровней:

- Локальный адрес узла, определяемый технологией, с помощью которой построена отдельная сеть, в которую входит данный узел. Для узлов, входящих в локальные сети, это MAC-адрес сетевого адаптера или порта маршрутизатора, например, 11-A0-17-3D-BC-01. Эти адреса назначаются производителями оборудования и являются уникальными адресами, так как управляются централизованно. Для всех существующих технологий локальных сетей MAC-адрес имеет формат 6 байтов: старшие 3 байта — идентификатор фирмы производителя, а младшие 3 байта назначаются уникальным образом самим производителем.

- IP-адрес, состоящий из 4 байт, например, 109.26.17.100. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (Network Information Center, NIC), если сеть должна работать как составная часть Internet. Обычно провайдеры услуг Internet получают диапазоны адресов у подразделений NIC, а затем распределяют их между своими абонентами. Номер узла в протоколе IP назначается независимо от локального адреса узла. Деление IP-адреса на поле номера сети и номера узла — гибкое, и граница между этими полями может устанавливаться весьма произвольно. Узел может входить в несколько IP-сетей. В этом случае узел должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

- Символьный идентификатор-имя, например, SERV1.IBM.COM. Этот адрес назначается администратором и состоит из нескольких частей, например, имени машины, имени организации, имени домена. Такой адрес, называемый также DNS-именем, используется на прикладном уровне, например, в протоколах FTP или telnet.

### 2.1.2. Три основных класса IP-адресов

IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме, и разделенных точками, например:

128.10.2.30 — традиционная десятичная форма представления адреса,

10000000 00001010 00000010 00011110 — двоичная форма представления этого же адреса.

Далее показана структура IP-адреса в зависимости от класса сети.

Класс А

0	N сети		N узла		
Класс В					
1	0	N сети		N узла	
Класс С					
1	1	0	N сети		N узла
Класс D					
1	1	1	0	адрес группы multicast	
Класс E					
1	1	1	1	0	зарезервирован

Адрес состоит из двух логических частей — номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая к номеру узла, определяется значениями первых битов адреса:

- Если адрес начинается с 0, то сеть относят к классу А, и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса А имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже.) В сетях класса А количество узлов должно быть больше  $2^{16}$ , но не превышать  $2^{24}$ .

- Если первые два бита адреса равны 10, то сеть относится к классу В и является сетью средних размеров с числом узлов  $2^8$  —  $2^{16}$ . В сетях класса В под адрес сети и под адрес узла отводится по 16 бит, то есть по 2 байта.

- Если адрес начинается с последовательности 110, то это сеть класса С с числом узлов не больше  $2^8$ . Под адрес сети отводится 24 бита, а под адрес узла — 8 бит.

- Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый, групповой адрес — multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.

- Если адрес начинается с последовательности 11110, то это адрес класса E, он зарезервирован для будущих применений.

В таблице приведены диапазоны номеров сетей, соответствующих каждому классу сетей.

Класс	Наименьший адрес	Наибольший адрес
<b>A</b>	0.1.0.0	126.0.0.0
<b>B</b>	128.0.0.0	191.255.0.0
<b>C</b>	192.0.1.0	223.255.255.0

<b>D</b>	224.0.0.0	239.255.255.255
<b>E</b>	240.0.0.0	247.255.255.255

### 2.1.3. Отображение символьных адресов на IP-адреса: служба DNS

*DNS (Domain Name System)* — это распределенная база данных, поддерживающая иерархическую систему имен для идентификации узлов в сети Internet. Служба DNS предназначена для автоматического поиска IP-адреса по известному символьному имени узла. Спецификация DNS определяется стандартами RFC 1034 и 1035. DNS требует статической конфигурации своих таблиц, отображающих имена компьютеров в IP-адрес.

Протокол DNS является служебным протоколом прикладного уровня. Этот протокол несимметричен — в нем определены DNS-серверы и DNS-клиенты. DNS-серверы хранят часть распределенной базы данных о соответствии символьных имен и IP-адресов. Эта база данных распределена по административным доменам сети Internet. Клиенты сервера DNS знают IP-адрес сервера DNS своего административного домена и по протоколу IP передают запрос, в котором сообщают известное символьное имя и просят вернуть соответствующий ему IP-адрес.

Если данные о запрошенном соответствии хранятся в базе данного DNS-сервера, то он сразу посылает ответ клиенту, если же нет — то он посылает запрос DNS-серверу другого домена, который может сам обработать запрос, либо передать его другому DNS-серверу. Все DNS-серверы соединены иерархически, в соответствии с иерархией доменов сети Internet. Клиент опрашивает эти серверы имен, пока не найдет нужные отображения. Этот процесс ускоряется из-за того, что серверы имен постоянно кэшируют информацию, предоставляемую по запросам. Клиентские компьютеры могут использовать в своей работе IP-адреса нескольких DNS-серверов, для повышения надежности своей работы. База данных DNS имеет структуру дерева, называемого доменным пространством имен, в котором каждый домен (узел дерева) имеет имя и может содержать поддомены. Имя домена идентифицирует его положение в этой базе данных по отношению к родительскому домену, причем точки в имени отделяют части, соответствующие узлам домена.

Корень базы данных DNS управляется центром Internet Network Information Center. Домены верхнего уровня назначаются для каждой страны, а также на организационной основе. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, а для различных типов организаций используются следующие аббревиатуры:

- com — коммерческие организации (например, microsoft.com);
- edu — образовательные (например, mit.edu);
- gov — правительственные организации (например, nsf.gov);
- org — некоммерческие организации (например, fidonet.org);

- net — организации, поддерживающие сети (например, nsf.net).

Каждый домен DNS администрируется отдельной организацией, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Каждый домен имеет уникальное имя, а каждый из поддоменов имеет уникальное имя внутри своего домена. Имя домена может содержать до 63 символов. Каждый хост в сети Internet однозначно определяется своим *полным доменным именем (fully qualified domain name, FQDN)*, которое включает имена всех доменов по направлению от хоста к корню. Пример полного DNS-имени:

```
server.aics.acs.cctpu.edu.ru
```

#### **2.1.4. Автоматизация процесса назначения IP-адресов узлам сети — протокол DHCP**

IP-адреса могут назначаться администратором сети вручную. Это представляет для администратора утомительную процедуру. Ситуация усложняется еще тем, что многие пользователи не обладают достаточными знаниями для того, чтобы конфигурировать свои компьютеры для работы в интрасети и должны поэтому полагаться на администраторов.

Протокол *Dynamic Host Configuration Protocol (DHCP)* был разработан для того, чтобы освободить администратора от этих проблем. Основным назначением DHCP является динамическое назначение IP-адресов. Однако, кроме динамического, DHCP может поддерживать и более простые способы ручного и автоматического статического назначения адресов.

В ручной процедуре назначения адресов активное участие принимает администратор, который предоставляет DHCP-серверу информацию о соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. Эти адреса сообщаются клиентам в ответ на их запросы к DHCP-серверу.

При автоматическом статическом способе DHCP-сервер присваивает IP-адрес (и, возможно, другие параметры конфигурации клиента) из пула (набора) наличных IP-адресов без вмешательства оператора. Границы пула назначаемых адресов задает администратор при конфигурировании DHCP-сервера. Между идентификатором клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первичного назначения сервером DHCP IP-адреса клиенту. При всех последующих запросах сервер возвращает тот же самый IP-адрес.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, что дает возможность впоследствии повторно использовать IP-адреса другими компьютерами. Динамическое распределение адресов позволяет строить IP-сеть, количество узлов в которой намного превышает количество имеющихся в распоряжении администратора IP-адресов.



## 2.2. Системные утилиты сетевой диагностики

### 2.2.1. Утилита ipconfig

Утилита ipconfig (IP configuration) предназначена для настройки протокола IP для операционной системы Windows. В данной лабораторной работе эта утилита будет использоваться только для получения информации о соединении по локальной сети. Для получения этой информации выполните «Пуск» → «Выполнить» → cmd и в командной строке введите:

```
ipconfig /all
```

В разделе «Адаптер Ethernet Подключение по локальной сети» для данной лабораторной будут необходимы поля «DHCP», «IP-адрес» и «DNS-серверы».

### 2.2.2. Утилита ping

Утилита ping (Packet Internet Groper) является одним из главных средств, используемых для отладки сетей, и служит для принудительного вызова ответа конкретной машины. Она позволяет проверять работу программ TCP/IP на удаленных машинах, адреса устройств в локальной сети, адрес и маршрут для удаленного сетевого устройства. В выполнении команды ping участвуют система маршрутизации, схемы разрешения адресов и сетевые шлюзы. Это утилита низкого уровня, которая не требует наличия серверных процессов на проверяемой машине, поэтому успешный результат при прохождении запроса вовсе не означает, что выполняются какие-либо сервисные программы высокого уровня, а говорит о том, что сеть находится в рабочем состоянии, питание проверяемой машины включено, и машина не отказала («не висит»).

В Windows утилита ping имеется в комплекте поставки и представляет собой программу, запускаемую из командной строки.

Запросы утилиты ping передаются по протоколу ICMP (Internet Control Message Protocol). Получив такой запрос, программное обеспечение, реализующее протокол IP у адресата, посылает эхо-ответ. Если проверяемая машина в момент получения запроса была загружена более приоритетной работой (например, обработкой и перенаправлением большого объема трафика), то ответ будет отправлен не сразу, а как только закончится выполнение более приоритетной задачи. Поэтому следует учесть, что задержка, рассчитанная утилитой ping, вызвана не только пропускной способностью канала передачи данных до проверяемой машины, но и загруженностью этой машины.

Эхо-запросы посылаются заданное количество раз (ключ -n). По умолчанию передается четыре запроса, после чего выводятся статистические данные.

**Обратите внимание:** поскольку с утилиты ping начинается хакерская атака, некоторые серверы в целях безопасности могут не посылать эхо-ответы

(например, `www.microsoft.com`). Не ждите напрасно, введите команду прерывания (`CTRL + C`).

**Формат команды:** `ping [-t][-a][-n][-l][-f][-i TTL][-v TOS]`  
`[-r][[имя машины]][[-j списокУзлов]][[-k списокУзлов]][-w]`

#### Параметры утилиты `ping`

Ключи	Функции
<b>-t</b>	Отправка пакетов на указанный узел до команды прерывания
<b>-a</b>	Определение имени узла по IP-адресу
<b>-n</b>	Число отправляемых запросов
<b>-l</b>	Размер буфера отправки
<b>-f</b>	Установка флага, запрещающего фрагментацию пакета
<b>-i TTL</b>	Задание времени жизни пакета (поле «Time To Live»)

На практике большинство опций в формате команды можно опустить, тогда в командной строке может быть: `ping имя узла` (для заикливания вывода информации о соединении используется опция `-t`; для вывода информации `n`-раз используется опция `-n` количество раз).

#### Пример :

```
ping -n 20 peak.mountin.net
```

Обмен пакетами с `peak.mountin.net [207.227.119.2]` по 32 байт:

Превышен интервал ожидания для запроса.

Ответ от 207.227.119.2: число байт=32 время=734мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=719мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=704мс TTL=231

Превышен интервал ожидания для запроса.

Ответ от 207.227.119.2: число байт=32 время=719мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=1015мс TTL=231

Превышен интервал ожидания для запроса.

Ответ от 207.227.119.2: число байт=32 время=703мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=782мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231

Превышен интервал ожидания для запроса.

Ответ от 207.227.119.2: число байт=32 время=687мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=735мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=672мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=704мс TTL=231

Статистика Ping для 207.227.119.2:

Пакетов: отправлено = 20, получено = 16, потеряно = 4 (20% потерь),

Приблизительное время передачи и приема:

наименьшее = 672мс, наибольшее = 1015мс, среднее = 580мс

### Пример определения имени узла по IP-адресу

ping -a 194.67.57.26

Обмен пакетами с mail.ru [194.67.57.26] по 32 байт: ...

### 2.2.3. Утилита tracert

Утилита tracert позволяет выявлять последовательность маршрутизаторов, через которые проходит IP-пакет на пути к пункту своего назначения.

**Формат команды:** tracert имя\_машины

имя\_машины может быть именем узла или IP-адресом машины. Выходная информация представляет собой список машин, начиная с первого шлюза и заканчивая пунктом назначения.

**Пример:**

tracert peak.mountin.net

Трассировка маршрута к peak.mountin.net [207.227.119.2]

с максимальным числом прыжков 30:

№	Пакет 1	Пакет 2	Пакет 3	DNS-имя узла и (или) его IP-адрес
1	<10 мс	<10 мс	<10 мс	SLAVE [192.168.0.1]
2	<10 мс	<10 мс	<10 мс	gw.b10.tpu.edu.ru [195.208.164.2]
3	<10 мс	<10 мс	<10 мс	195.208.177.62
4	<10 мс	<10 мс	<10 мс	news.runnet.tomsk.ru [195.208.160.4]
5	<10 мс	<10 мс	16 ms	ra.cctpu.tomsk.su [195.208.161.34]
6	781 ms	563 ms	562 ms	spb-2-gw.runnet.ru [194.85.33.9]
7	547 ms	594 ms	578 ms	spb-gw.runnet.ru [194.85.36.30]
8	937 ms	563 ms	562 ms	20.201.atm0-201.ru-gw.run.net [193.232.80.105]
9	1125 ms	563 ms	547 ms	fi-gw.nordu.net [193.10.252.41]

10	906 ms	1016 ms	578 ms	s-gw.nordu.net [193.10.68.41]
11	844 ms	828 ms	610 ms	dk-gw2.nordu.net [193.10.68.38]
12	578 ms	610 ms	578 ms	sl-gw10-cop-9-0.sprintlink.net [80.77.65.25]
13	610 ms	968 ms	594 ms	sl-bb20-cop-8-0.sprintlink.net [80.77.64.37]
14	641 ms	672 ms	656 ms	sl-bb21-msq-10-0.sprintlink.net [144.232.19.29]
15	671 ms	704 ms	687 ms	sl-bb21-nyc-10-3.sprintlink.net [144.232.9.106]
16	985 ms	703 ms	765 ms	sl-bb22-nyc-14-0.sprintlink.net [144.232.7.102]
17	719 ms	734 ms	688 ms	144.232.18.206
18	891 ms	703 ms	734 ms	p1-0.nycmny1-nbr1.bbnplanet.net [4.24.8.161]
19	719 ms	985 ms	703 ms	so-6-0-0.chcgil2-br2.bbnplanet.net [4.24.4.17]
20	688 ms	687 ms	703 ms	so-7-0-0.chcgil2-br1.bbnplanet.net [4.24.5.217]
21	719 ms	703 ms	672 ms	p1-0.chcgil2-cr9.bbnplanet.net [4.24.8.110]
22	687 ms	719 ms	687 ms	p2-0.nchicago2-cr2.bbnplanet.net [4.0.5.242]
23	781 ms	703 ms	672 ms	p8-0-0.nchicago2-core0.bbnplanet.net [4.0.6.2]
24	672 ms	703 ms	687 ms	fa0.wcnet.bbnplanet.net [207.112.240.102]
25	734 ms	687 ms	688 ms	core0-s1.rac.cyberlynk.net [209.100.155.22]
26	1188 ms	*	890 ms	peak.mountin.net [207.227.119.2]

Трассировка завершена.

Пакеты посылаются по три на каждый узел. Для каждого пакета на экране отображается величина интервала времени между отправкой пакета и получением ответа. Символ \* означает, что ответ на данный пакет не был получен. Если узел не отвечает, то при превышении интервала ожидания ответа выдается сообщение «Превышен интервал ожидания для запроса». Интервал ожидания ответа может быть изменен с помощью опции – w команды tracert.

Команда tracert работает путем установки поля времени жизни (числа переходов) исходящего пакета таким образом, чтобы это время истекло до достижения пакетом пункта назначения. Когда время жизни истечет, текущий шлюз отправит сообщение об ошибке на машину-источник. Каждое приращение поля времени жизни позволяет пакету пройти на один маршрутизатор дальше.

**Примечание:**

*Для вывода информации в файл используйте символ перенаправления потока вывода «>». Данный символ справедлив и для утилит ping и tracert .*

**Пример:**

tracert 195.208.164.1 > tracert.txt

Отчет о трассировке маршрута до указанного узла будет помещен в файл tracert.txt.

### 3. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ

Отчёт по практической работе необходимо оформить в MS Word.

3.1 С помощью утилиты ipconfig определить IP адрес и физический адрес основного сетевого интерфейса компьютера, IP адрес шлюза, IP адреса DNS-серверов и используется ли DHCP.. Результаты представить **в виде таблицы** .

3.2 Проверить состояние связи с любыми двумя узлами (работоспособными) в соответствии с вариантом задания. Число отправляемых запросов должно составлять не менее 20. В качестве результата отразить для каждого из исследуемых узлов **в виде таблицы** :

- a. процент потерянных пакетов;
- b. среднее время приема-передачи;
- c. количество маршрутизаторов (с учетом шлюза) до опрашиваемого узла;
- d. IP адрес узла.
- e. класс сети, к которой принадлежит данный узел;
- f. имя узла, полученное по IP-адресу узла.

В отчёте необходимо пояснить, как были определены значения.

3.3. Произвести трассировку двух работоспособных узлов в соответствии с вариантом задания. Результаты запротоколировать в таблице.

№ узла	Время прохождения пакета №1	Время прохождения пакета №2	Время прохождения пакета №3	Среднее время прохождения пакета	DNS-имя маршрутизатора	IP-адрес маршрутизатора
--------	-----------------------------	-----------------------------	-----------------------------	----------------------------------	------------------------	-------------------------

Если значения времени прохождения трёх пакетов отличаются более, чем на 10 мс, либо если есть потери пакетов, то для соответствующих узлов среднее время прохождения необходимо определять с помощью утилиты ping по 20 пакетам. По результатам таблицы в отчете привести **график** изменения среднего времени прохождения пакета. В отчёте привести одну копию окна с результатами команды tracert. Для каждого опрашиваемого узла определить участок сети между двумя соседними маршрутизаторами, который характеризуется наибольшей задержкой при пересылке пакетов. Для найденных маршрутизаторов с помощью сервиса Whois определить название организации и контактные данные (тел., email). Полученную информацию необходимо указать в отчёте.

\* Сервис Whois позволяет быстро получить всю информацию о регистрации домена, например, дату регистрации и возраст домена, или узнать контакты, по которым можно связаться с организацией или человеком, чей домен вас заинтересовал.

### 4. ВАРИАНТЫ ЗАДАНИЙ

1	<a href="http://www.informika.ru">www.informika.ru</a> <a href="http://www.rfbr.ru">www.rfbr.ru</a> <a href="http://www.ras.ru">www.ras.ru</a>
2	<a href="http://www.gpntb.ru">www.gpntb.ru</a> <a href="http://www.rusmedserv.com">www.rusmedserv.com</a> <a href="http://www.nsc.ru">www.nsc.ru</a>

3	www.chemnet.ru www.rsl.ru www.philosophy.ru
4	www.rbc.ru www.membrana.ru www.mrsu.ru
5	www.viniti.ru www.sostav.ru www.ioffe.ru
6	www.fegi.ru www.elibrary.ru www.extech.ru
7	www.ripn.net www.shpl.ru sai.msu.su
8	www.scsml.rssi.ru www.sccc.ru www.nlr.ru
9	web.ru www.kamaz.ru www.rulex.ru
10	www.jinr.ru uic.nnov.ru www.ruthenia.ru
11	www.tractor.ru www.rsci.ru www.astronet.ru
12	www.keldysh.ru www.fom.ru www.inauka.ru
13	www.gramota.ru www.csa.ru www.bionet.nsc.ru
14	www.inp.nsk.su www.scientific.ru www.med2000.ru
15	www.gpi.ru iki.cosmos.ru www.spsl.nsc.ru
16	www.uiggm.nsc.ru hist.dcn-asu.ru www.cemi.rssi.ru
17	psychology.net.ru www.irex.ru www.medlinks.ru
18	www.viniti.ru www.sostav.ru www.gramota.ru
19	www.sccc.ru www.nlr.ru www.fom.ru
20	uic.nnov.ru www.ruthenia.ru www.rsl.ru

**Практическая работа №15. «IP-адресация в сети»**

**Теоретический материал**

**1. Разделение IP-адреса на номер сети и номер узла на основе классов**

Традиционная схема разделения IP-адреса на номер сети и номер узла основана на понятии класса, определяемого значениями нескольких первых бит адреса.

**Класс А:**

<b>А</b>	<b>Номер сети</b>	<b>Номер узла в сети</b>
0		
	один байт	три байта

**Класс В:**

<b>В</b>	<b>Номер сети</b>	<b>Номер узла в сети</b>
1 0		
	два байта	два байта

**Класс С:**

<b>С</b>	<b>Номер сети</b>	<b>Номер узла в сети</b>
1 1 0		
	три байта	один байт

**Класс А.** Первый байт адреса используется для номера сети, остальные три – для номера узла (количество адресов в сети  $2^{24}$ ).

**Класс В.** Первые два байта используются для номера сети, остальные – для номера узла (количество адресов в сети  $2^{16}$ ).

**Класс С.** Первые три байта используются для номера сети, последний байт – для номера узла (количество адресов в сети  $2^8$ ).

**Класс D.** Первые биты равны 1110 – адреса мультикаст (multicast), предназначены для адресации группы узлов.

Клас с	Первые биты	Наименьший адрес сети	Наибольший адрес сети	Максимальное количество узлов
<b>А</b>	<b>0</b>	1.0.0.0	126.0.0.0	$2^{24}$ (16 777 216-2)
<b>В</b>	<b>10</b>	128.0.0.0	191.255.0.0	$2^{16}$ (65536-2)
<b>С</b>	<b>110</b>	192.0.1.0	223.255.255.0	$2^8$ (256-2)
<b>D</b>	<b>1110</b>	224.0.0.0	239.255.255.255	Multicast
<b>E</b>	<b>11110</b>	240.0.0.0	247.255.255.255	зарезервирован

В некоторых случаях необходимо отдельно записывать номер сети и номер узла, из которых состоит IP-адрес. В записи номера сети соответствующие номеру узла разряды адреса заменяют нулями, в записи номера узла нулями заменяют разряды, соответствующие номеру сети.

**Пример 1**

IP-адрес 192.9.7.5. Поскольку первые биты равны 110, следовательно, это адрес класса С.

11000000	00001001	00000111	00000101
Номер сети			Номер узла

Номер сети (последние три бита) – 192.9.7.0(11000000.00001001.00000111.00000000),

Номер узла (первый бит) – 0.0.0.5(00000000.00000000.00000000.00000101).

**Пример 2**

IP-адрес 62.76.9.17. Поскольку первый бит равен 0, следовательно, это адрес класса А.

00111110	01001100	00001001	00010001
Номер сети	Номер узла		

Номер сети (последний бит) – 62.0.0.0 (00111110.00000000.00000000.00000000)

Номер узла (три первых бита) – 0.76.9.17 (00000000.01001100.00001001.00010001)

## 2. Соответствие блоков адресов номерам сетей на основе классов

Номер сети определяет блок адресов с одинаковым префиксом (одинаковой старшей частью), зависящим от класса адреса.

### Пример 3

Рассмотрим номер сети 192.168.169.0 (11000000.10101000.10101001.00000000)

Первые разряды адреса имеют значение 110, следовательно, это адрес класса С. Этому номеру сети соответствует блок адресов 192.168.169.0 – 192.168.169.255, все адреса этого блока имеют одинаковые первые три октета, равные 192.168.169.

### Пример 4

Рассмотрим номер сети 62.0.0.0 (00111110.00000000.00000000.00000000)

Первый разряд адреса имеет значение 0, следовательно, этот адрес класса А. Этому номеру сети соответствует блок адресов 62.0.0.0 – 62.255.255.255, все адреса этого блока имеют одинаковый первый октет, равный 62.

## 3 Разделение IP-адреса на номер сети и номер узла на основе масок

**Маска** – это используемое совместно с IP-адресом четырехбайтовое число, двоичная запись которого содержит единицы в разрядах, соответствующих в адресе номеру сети, и нули в разрядах, соответствующих номеру узла. Единицы в маске начинаются в первом разряде адреса и не могут чередоваться с нулями.

С помощью маски можно выделять произвольное количество разрядов для номера сети, что позволяет отказаться от понятий классов адресов и сделать более гибкой систему адресации.

### Пример 5. Запись маски и IP-адреса

Десятичная форма	192.168.74.64	255.255.255.192
Двоичная форма	11000011.10101000.01001010.01000000	11111111.11111111.11111111.11000000

Для указания количества разрядов, выделенных для номера сети, также используется указание префикса адреса. Запись адреса с префиксом имеет вид: **IP-адрес/Префикс**, где Префикс – число разрядов, выделенных для номера сети.

Например, запись 192.168.75.64/26 означает, что в адресе 192.168.75.64 под номер сети отведено 26 двоичных разрядов, соответствующая маска 255.255.255.192.

192	168	75	64
1100 0011	1010 1000	0100 1011	0100 0000
255	255	255	192
<b>1111 1111</b>	<b>1111 1111</b>	<b>1111 1111</b>	<b>1100 0011</b>
<b>Номер сети</b>			

Значения масок стандартных классов адресов:

класс А – 11111111.00000000.00000000.00000000 (255.0.0.0);

класс В – 11111111.11111111.00000000.00000000 (255.255.0.0);

класс С – 11111111.11111111.11111111.00000000 (255.255.255.0).



#### 4. Вычисление номера сети и номера узла по заданному IP-адресу и маске

Для вычисления номера сети по заданному IP-адресу и маске необходимо применить побитовую операцию «И» (логическое умножение) к адресу и маске.

Такая операция называется наложением маски на адрес.

X	Y	X И Y
0	0	0
1	0	0
0	1	0
1	1	1

Для вычисления номера узла по заданному IP-адресу и маске необходимо применить побитовую операцию «И» к адресу и результату применения побитовой операции «НЕ» к маске.

#### Пример 8

Вычислим номер сети и номер узла для адреса 215.17.125.177 и маски 255.255.255.240.

IP-адрес:	215.17.125.177	11010111.00010001.01111101.10110001
Маска:	255.255.255.240	11111111.11111111.11111111.11110000

Номер сети (операция И)

```

11010111.00010001.01111101.10110001
× 11111111.11111111.11111111.11110000
-----
11010111.00010001.01111101.10110000
214. 17. 125. 176
    
```

Номер узла:

1. Операция НЕ к маске

```

НЕ 11111111.11111111.11111111.11110000
    00000000.00000000.00000000.00001111
    
```

2. Операция И к адресу и результату п.1.

```

11010111.00010001.01111101.10110001
× 00000000.00000000.00000000.00001111
-----
00000000.00000000.00000000.00000001
0. 0. 0. 1
    
```

Номер сети	215.17.125.176	11010111.00010001.01111101.10110000
Номер узла	0.0.0.1	00000000.00000000.00000000.00000001

#### Пример 9

Вычислим номер сети и номер узла для адреса 67.38.173.245 и маски 255.255.240.0.

IP-адрес (X)	67.38.173.245	01000011.00100110.10101101.11110101	
Маска (Y)	255.255.240.0	11111111.11111111.11110000.00000000	
<b>Номер сети</b>	<b>67.38.160.0</b>	<b>01000011.00100110.10100000.00000000</b>	<b>X И Y</b>
НЕ (Y)		00000000.00000000.00001111.11111111	НЕ (Y)
X		01000011.00100110.10101101.11110101	X
<b>Номер узла</b>	<b>0.0.13.245</b>	<b>00000000.00000000.00001101.11110101</b>	<b>X И (НЕ (Y))</b>

## 5. Соответствие блоков адресов номерам сетей на основе масок

При использовании маски, так же, как и в случае адресации на основе классов, номер сети определяет блок адресов с одинаковым префиксом.

### Пример 10

В маске 255.255.255.192 (11111111.11111111.11111111.11000000) выделено 26 разрядов под номер сети и 6 разрядов под номер узла.

Номеру сети 192.168.74.64 с данной маской соответствует блок адресов:

Маска: 11111111.11111111.11111111.11000000 (255.255.255.192)

Н.с: 11000011.10101000.01001010.01000000 (192.168.74.64)

Адрес 1: 11000011.10101000.01001010.01000000 (192.168.74.64)

Адрес 2: 11000011.10101000.01001010.01000001 (192.168.74.65)

.....

Адрес 63: 11000011.10101000.01001010.01111110 (192.168.74.126)

Адрес 64: 11000011.10101000.01001010.01111111 (192.168.74.127)

Всего в этом блоке  $2^6 = 64$  адресов (192.168.74.64 – 192.168.74.127). Все адреса имеют одинаковый префикс (первые 26 разрядов): 11000011.10101000.01001010.01

### Задание на лабораторную работу

1. Изучить теоретический материал
2. Выполнить практические задания

#### Задание 1. Заполнить таблицу

Класс			
IP-адрес в 10-ой записи	135.155.25.13	201.198.145.2	81.15.14.8
IP-адрес в 2-ой записи			
IP-адрес в 16-ой записи			
Номер сети в 2-ой записи			
Номер сети в 10-ой записи			
Номер узла в 2-ой записи			
Номер узла в 10-ой записи			

**Задание 2. Определить класс адреса и номер сети по IP-адресу в соответствии с таблицей, где N – номер студента по журналу**

Номер варианта (N)	IP-адрес / Префикс					
	1-5	$16*N+2.$	$2*N.$	$10*N+12.$	N.	/
6-10	$17*N+20.$	$10*N+90.$	$5*N+22.$	N.	/	27
11-15	$13*N-25.$	$10*N+32.$	$4*N+5.$	N.	/	25

16-20	10*N+55.	9*N-5.	5*N-13.	N.	/	24
21-25	5*N+78.	7*N.	4*N+4.	N.	/	25
26-30	6*N-20.	7*N.	3*N-15.	N.	/	26

**Задание 3. Вычислить номер сети и номер узла (хоста) для адреса и маски.**

№ п/п	Адрес	Маска
1	192.168.0.15	255.255.255.192
2	192.104.10.25	255.255.255.224
3	16.0.0.15	255.255.255.192

3. Оформить отчет о работе

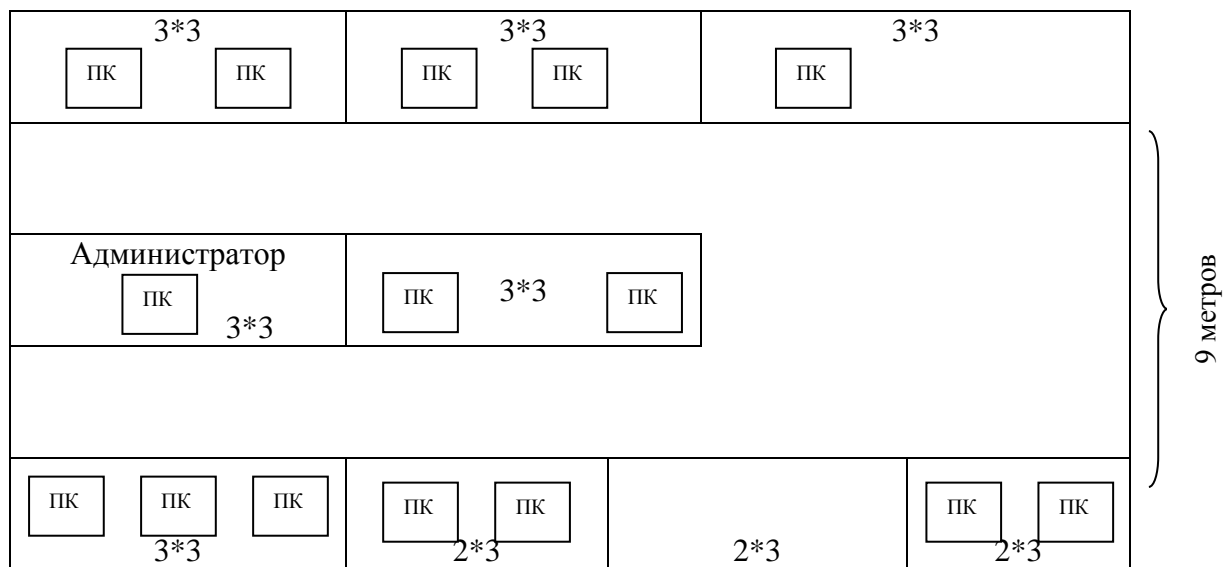
Практическая работа №16. Выполнение самостоятельной работы

**Цели работы:**

- Закрепить полученные знания на практике.

**Задание 1**

Спроектировать компьютерную сеть по топологии «звезда» по заданному изображению. Центром звезды сделать помещение администратора.



Указать тип кабеля, сетевые устройства, дополнительное оборудование. Результаты свести в таблицу.

№ п/п	Вид	Кол-во	Ед. измерения

**Задание 2**

Проанализируйте приведенные ниже IP-адреса и маски подсети. Определите адрес сети (третий столбец) и адрес узла (четвертый столбец) в каждом из указанных случаев

85.13.32.201	255.0.0.0		
131.13.32.201	255.255.0.0		
192.168.1.150	255.255.255.0		
192.168.1.150	255.255.255.128		

### Задание 3

Проанализируйте приведенные ниже IP-адреса и отметьте во втором столбце таблицы те из них, которые можно использовать для *адресации узлов*. В третьем столбце укажите, почему остальные адреса являются некорректными.

10.15.256.30		
169.254.25.38		
244.13.56.91		
0.0.0.0		
92.107.30.0		
127.16.25.30		
255.255.255.255		

### Задание 4

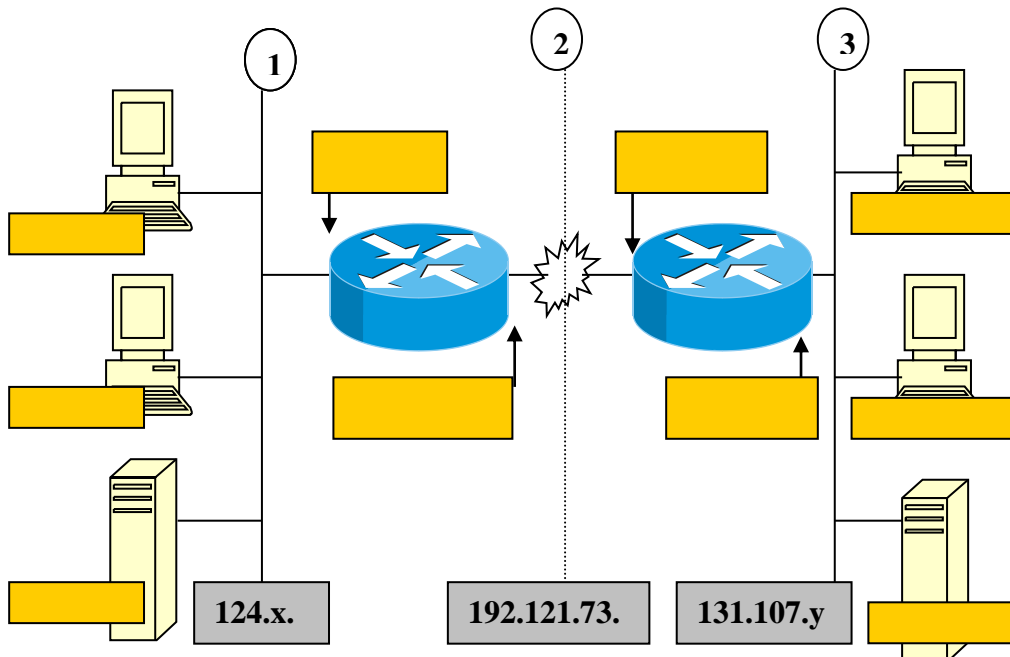
Используйте IP-адрес и соответствующую длину маски сети, чтобы получить IP-адрес сети, маску сети и IP-адрес широковещательной рассылки в данной сети. Дано: **110.157.233.184/12**

### Задание 5

По данным IP-адресам определить, к сети какого класса они принадлежат, получить IP-адрес сети, маску сети и IP-адрес широковещательной рассылки в данной сети: 110.157.233.184.

### Задание 6

Привести пример распределения IP-адресов.

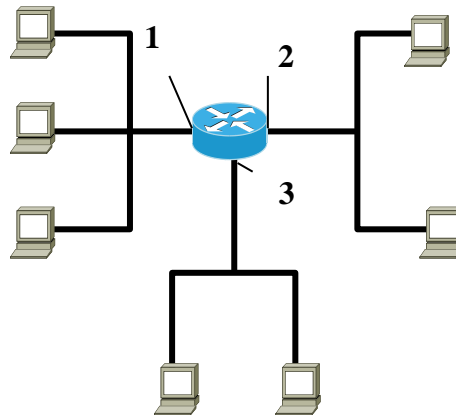


### Задание 7

Пусть IP-адрес некоторого узла подсети равен 198.65.12.67, а значение маски для этой подсети - 255.255.255.240. Определите номер подсети. Какое максимальное число узлов может быть в этой подсети?

### Задание 8

К маршрутизатору подключены три сети класса C. Приведите пример адресов интерфейсов.



### Задание 9

Дано:

Маска сети: 255.255.255.248

Адрес IP: 192.168.1.219

Определите максимальное число сетевых устройств, которые могут быть подключены к данной сети.

### Задание 10

Определите MAC-адрес по известному IP-адресу.

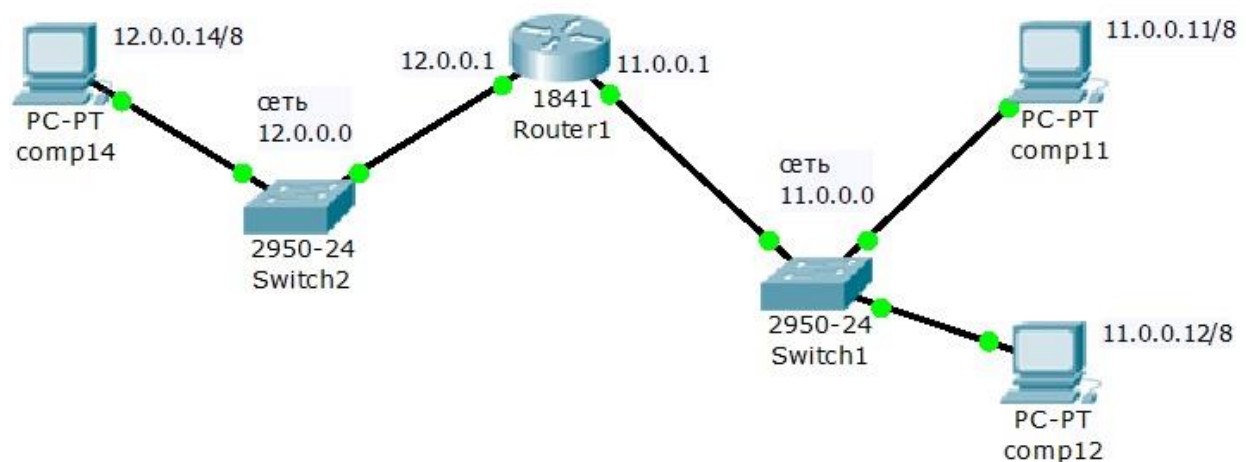
### Задание 11

Принадлежат ли указанные IP-адреса к одной подсети:  
123.65.164.72/27

123.65.168.74 -

### Задание 12

Запустите программу Cisco Packet Tracer и создайте две сети (сеть 11.0.0.0 и 12.0.0.0) по схеме, представленной на рисунке:

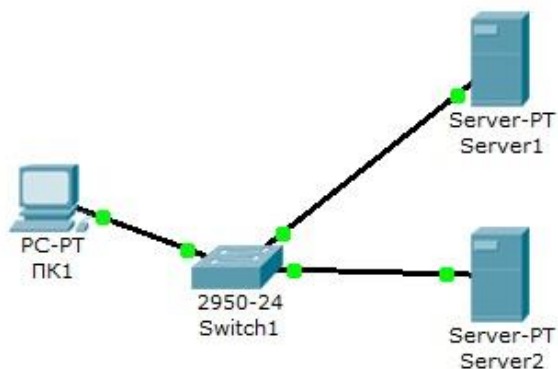


Задайте названия устройств, как показано на схеме.

Задайте параметры протокола TCP/IP и шлюзы для компьютеров comp11, comp12 и comp14, как показано на схеме

### Задание 13

Создайте следующую схему сети, представленную на рисунке:



Задача:

Настроить сеть следующим образом:

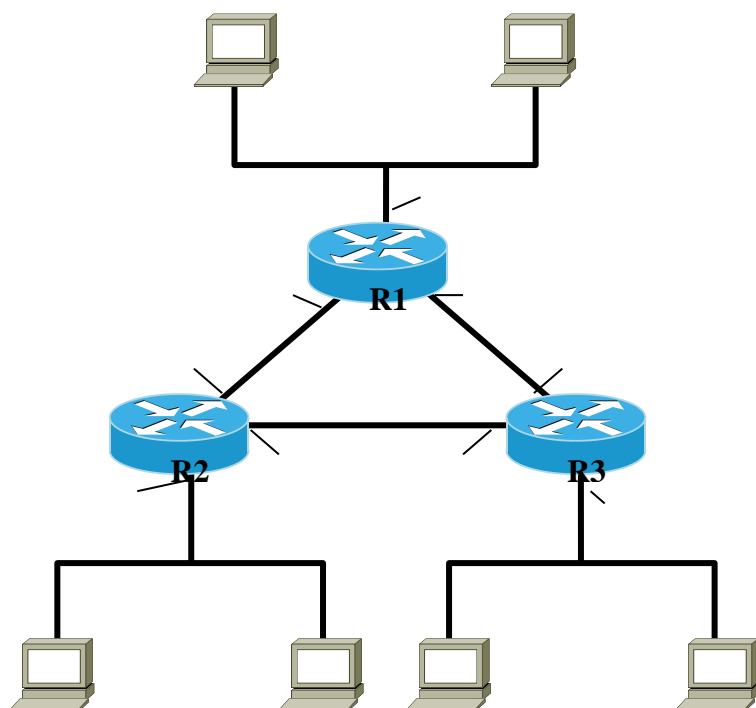
1 - Server1 – DNS и Web сервер;

2 - Server2 – DHCP сервер;

3 - Компьютер ПК1 получает параметры протокола TCP/IP с DHCP сервера и открывает сайт [www.rambler.ru](http://www.rambler.ru) на Server1.

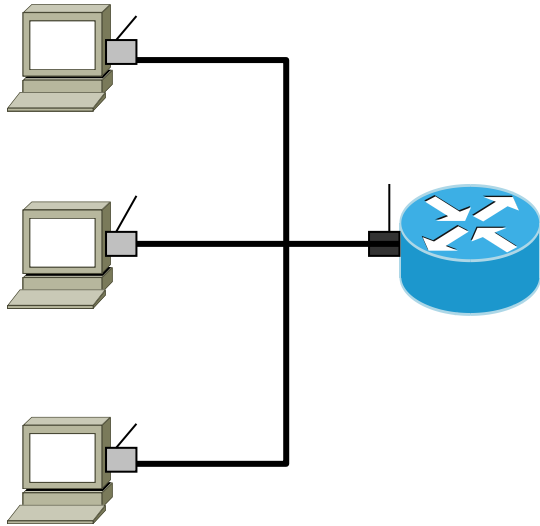
### Задание 14

На рисунке показаны три маршрутизатора, объединяющие три сети класса C. Приведите пример адресов интерфейсов.



### Задание 15

У каждого адаптера, соединенного с локальной сетью, есть уникальный адрес. Приведите пример таких адресов.



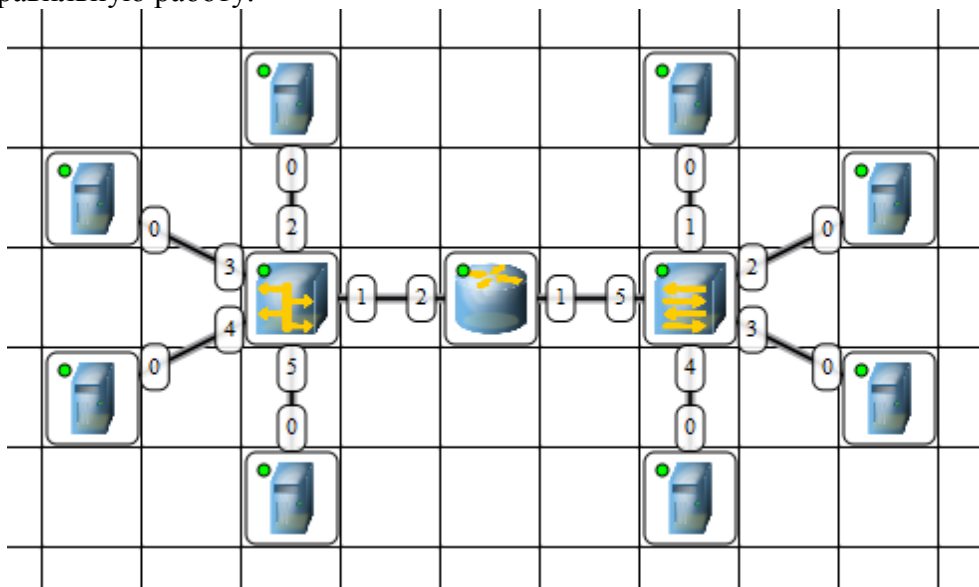
### Задание 16

Какие из ниже приведенных адресов сетей рекомендованы для локального использования и не обрабатываются маршрутизаторами Internet

10.0.0.0	11.0.0.0	172.16.0.0	173.16.0.0
191.16.0.0	172.31.0.0	191.192.0.0	192.168.0.0
192.168.168.0	200.201.202.0	192.168.255.0	223.0.0.0.

### Задание 17

В программе Netemul построить сеть из восьми ПК, хаба, коммутатора и роутера. Настроить ее правильную работу.



### Задание 18

Выполнить трассировку заданного узла. Пояснить полученные результаты. Узел **vvsu.ru**. Поясните полученный результат.

#### Требования к отчету

В отчете запишите ответы на задания самостоятельной работы. Обоснуйте каждый шаг получения результата

### 3.4 ПМ.03 УЧАСТИЕ В ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ

#### УП.03.01. УЧЕБНАЯ ПРАКТИКА К МДК 03.01. ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

##### 7 СЕМЕСТР

##### Практическая работа № 1

**Тема:** Создание простого приложения Qt с применением объекта типа QApplication, который обеспечивает управление всеми ресурсами приложения, а также обработку событий, поступающих от операционной системы.

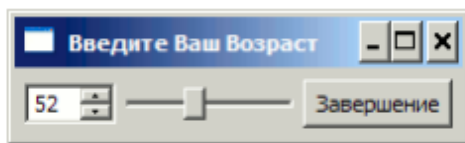
**Цель:** разработать код программного модуля с использованием элемент типа QSpinBox, слайдера – элемент типа QSlider.

##### **Методические указания:**

На рисунке показан внешний вид простого приложения, в котором предлагается ввести возраст с использованием одного из трех вариантов ввода:

- 1) непосредственного ввода числа,
- 2) посредством стрелок (элемент типа QSpinBox), последовательно увеличивающих или уменьшающих значение,
- 3) с помощью специального ползунка (слайдера – элемент типа QSlider).

Кроме того, задано верхнее ограничение вводимого возраста, что должно быть корректно отработано слайдером в крайних положениях, а изменение значения любым способом должно синхронизировать его положение.



##### **Ход работы:**

Создадим директорию ex1, в которую запишем файл ex1.cpp, содержащий следующий код. Для этого можно воспользоваться любым текстовым редактором, например

Блокнот или Notepad.

```
#include <QApplication>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QSlider>
#include <QSpinBox>
#include <QPushButton>
#include <QTextCodec>
```

```
// Преобразуем входную последовательность символов в кодировку UNICODE
```

```
#define RUS( str ) codec->toUnicode(str)
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    QApplication app(argc, argv);
```

```
    // Обеспечиваем перекодировку русских букв из кодировки,
```

```
    // в которой написана программа.
```

```
    // "Windows-1251" - для Windows, "KOI8-R" - Linux и т.д.
```

```
    QTextCodec * codec = QTextCodec::codecForName("Windows-1251");
```

```
// Создаем главное окно
```

```
    QWidget *hbox = new QWidget();
```

```
    hbox->setWindowTitle( RUS("Введите Ваш Возраст") );
```

```
    QSpinBox *spinBox = new QSpinBox( hbox );
```



```

QSlider *slider = new QSlider(Qt::Horizontal, hbox );
spinBox->setRange(0, 130);
slider->setRange(0, 130);
spinBox->setValue(35);
QPushButton * btn = new QPushButton( RUS("Завершение"), hbox );
//*****
QHBoxLayout *layout = new QHBoxLayout; // выравнивание по горизонтали
//QVBoxLayout *layout = new QVBoxLayout; // выравнивание по вертикали
layout->setContentsMargins(5,5,5,5); // устанавливаем внешние границы
layout->setSpacing(5); // устанавливаем интервал элементов внутри
hbox->setLayout(layout); // связываем layout с hbox
// устанавливаем порядок следования элементов
layout->addWidget(spinBox);
layout->addWidget(slider);
layout->addWidget(btn);
//*****
// связываем сигнал изменения spinBox со слотом slider
QObject::connect(spinBox, SIGNAL(valueChanged(int)),
    slider, SLOT(setValue(int)));
// связываем сигнал изменения slider со слотом spinBox
QObject::connect(slider, SIGNAL(valueChanged(int)),
    spinBox, SLOT(setValue(int)));
// связываем сигнал нажатия btn со слотом close главного окна
QObject::connect(btn, SIGNAL(clicked(bool)),
    hbox, SLOT(close()));
hbox->show(); // отображаем окно
return app.exec(); // запускаем цикл обработки сообщений
}

```

В приведенной программе объект типа QApplication обеспечивает управление всеми ресурсами приложения, а также обработку событий, поступающих от операционной системы.

По терминологии Qt, визуальные объекты – элементы графического интерфейса пользователя называются виджетами (widget – window gadget) и являются потомками класса QWidget. Любой из них может стать главным окном или быть использован другим виджетом, выполняющим роль контейнера. В представленном выше примере виджет hbox назначается главным окном, а его отображение обеспечивается вызовом функции hbox->show(). В главном окне (виджете) размещено три подчиненных виджета – объекты spinBox и slider типов QSpinBox и QSlider, а также btn типа QPushButton (кнопка). Для каждого виджета может быть выбрана одна схема размещения подчиненных виджетов (Layout). В частности для hbox используем схему горизонтального выравнивания QHBoxLayout \* layout. Связывание главного виджета со схемой размещения выполняется вызовом

hbox->setLayout(layout). Подчиненные виджеты будут отображены в порядке их добавления (вызовы layout->addWidget).

Следующие вызовы обеспечат связывание сигналов valueChanged() объектов spinBox, slider таким образом, чтобы изменение любого из них приводило к изменению другого.

```
QObject::connect(spinBox, SIGNAL(valueChanged(int)), slider, SLOT(setValue(int)));
```

```
QObject::connect(slider, SIGNAL(valueChanged(int)), spinBox, SLOT(setValue(int)));
```

Связывание сигнала clicked() объекта btn со слотом close() главного окна приведет к тому, что по нажатию на эту кнопку, окно hbox будет закрыто.

```
QObject::connect(btn, SIGNAL(clicked(bool)), hbox, SLOT(close()));
```

### **Компиляция и компоновка проекта.**

После того, как сpp-файл с текстом программы создан, необходимо создать qmake проект для её сборки. Для использования средств разработки, необходимо, чтобы переменные

среды окружения, указывающие местоположение и тип компилятора, который будет использован при сборке Qt-проектов, были правильно определены.

Для создания файла-проекта, включающего файлы текущей директории, воспользуемся специальной командой `qmake -project`.

В текущей директории должен появиться файл `ex1.pro`. Содержимое этого файла определяет характеристики процесса сборки исполняемого файла из файлов проекта.

Файл отредактируем следующим образом:

```
TEMPLATE = app      # тип исполняемого файла - .exe
TARGET = ex1       # имя исполняемого файла – ex1
QT += gui
CONFIG += release   # имя поддиректории для исполняемого файла
# Input
SOURCES = ex1.cpp   # имя исходного файла программы
```

Запускаем `qmake ex1.pro`, который на базе файла `ex1.pro` сформирует файл `Makefile`, определяющий фактический порядок сборки программы, положение компилятора и необходимых библиотек.

После того, как в текущей директории появился файл `Makefile`, вводим команду `nmake` (`mingw32-make` или `make` в зависимости от компилятора и при их наличии в текущей версии Qt).

Результат сборки программы – файл `ex1.exe`.

В процессе работы `ex1.exe` могут потребоваться динамические библиотеки `QtCore4.dll` и `QtGui4.dll`, которые должны находиться в путях автовызова, устанавливаемых системной переменной `Path`, или быть скопированы в директорию запуска приложения `ex1.exe`.

Задание: Замените в программе схему выравнивания `QHBoxLayout` на `QVBoxLayout` и зафиксируйте результат.

## Практическая работа № 2

**Тема:** Создание простого приложения в QtDesigner

**Цель:** разработка кода программного модуля, созданного в QtDesigner, позволяющего использовать менеджеры компоновки.

### **Методические указания:**

Рассмотрим разработку прототипов диалогов в QtDesigner.

QtDesigner, помимо разнообразных виджетов, позволяет использовать менеджеры компоновки.

В MS Windows исполняемый файл QtDesigner называется `designer.exe` и может быть запущен из консоли «Qt Command Prompt» по своему имени `designer` или через меню Пуск/Qt SDK by Nokia v.../Tools/Qt Designer. Qt Designer входит в любой комплект

Qt, однако, в комплекте Qt SDK может не иметь ярлыка, доступного из меню «Пуск», так как его функции в Qt SDK дублирует QtCreator. В этом случае можно непосредственно запустить программу `c:\Qt\xx\qt\bin\designer.exe`.

Визуальное проектирование формы приложения

После запуска QtDesigner появится окно, показанное на рисунке 6. В качестве примера приложения создадим диалог без кнопок по шаблону «Dialog without Buttons».

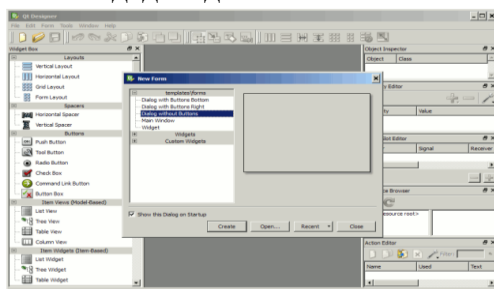


Рисунок 6 – Выбор шаблона формы в QtDesigner

После выбора шаблона появится форма, показанная на рисунке 7. Изменим название диалога посредством редактирования свойства `objectName=DialogEx2` в правой колонке.

### Практическая работа № 3

**Тема:** визуальный компонент `QLabel`

**Цель:** Создать код программы с визуальным компонентом `QLabel`, который отображает надпись "Hello, Qt!".

Ниже приводится текст простейшей Qt программы:

```
1 #include <qapplication.h>
2 #include <qlabel.h>
3 int main(int argc, char *argv[])
4 {
5     QApplication app(argc, argv);
6     QLabel *label = new QLabel("Hello, Qt!", 0);
7     app.setMainWidget(label);
8     label->show();
9     return app.exec();
10 }
```

Здесь, в строках 1 и 2, подключаются определения классов `QApplication` и `QLabel`.

В строке 5 создается экземпляр класса `QApplication`, который управляет ресурсами приложения. Конструктору `QApplication` передаются аргументы `argc` и `argv`, поскольку Qt имеет возможность обрабатывать аргументы командной строки.

В строке 6 создается визуальный компонент `QLabel`, который отображает надпись "Hello, Qt!". В терминологии Qt, все визуальные компоненты, из которых строится графический интерфейс, называются *виджетами* (widgets). Кнопки, меню, полосы прокрутки и разнообразные рамки -- все это виджеты. Одни виджеты могут содержать в себе другие виджеты, например, главное окно приложения -- это самый обычный виджет, который может содержать `QMenuBar`, `QToolBar`, `QStatusBar` и др. Аргумент 0, передаваемый конструктору `QLabel` (в строке 6) -- это "пустой" (null) указатель, который сообщает о том, что этот виджет не имеет "хозяина", т.е. не включается в другой виджет.

В строке 7 назначается "главный" виджет приложения. Когда пользователь закрывает "главный" виджет приложения (например, нажатием на кнопку "X" в заголовке окна), то программа завершает свою работу. Если в программе не назначить главный виджет, то она продолжит исполнение в фоновом режиме даже после того, как пользователь закроет окно.

В строке 8, метка делается видимой. Виджеты всегда создаются невидимыми, чтобы у программиста оставалась возможность настроить параметры отображения до того, как они станут видимы.

В строке 9 выполняется передача управления библиотеке Qt. С этого момента программа переходит в режим ожидания, когда она ничего не делает, а просто ждет действий пользователя, например, нажатие на клавишу или кнопку мыши.

Любое действие пользователя порождает *событие*, в ответ на которое программа может вызвать одну или более функций. В этом смысле, приложения с графическим интерфейсом кардинально отличаются от обычных программ, с пакетной обработкой данных, которые приняв ввод от пользователя, они самостоятельно обрабатывают его, выдают результаты и завершают свою работу без дальнейшего участия человека.



Теперь самое время проверить работу приложения. Скопируйте текст программы в файл, с именем `hello.cpp`, в каталог `hello`.

Перейдите в этот каталог и дайте команду:

```
qmake -project
```

она создаст платформу-независимый файл проекта (hello.pro), а затем дайте следующую команду:

```
qmake hello.pro
```

Эта команда создаст Makefile, на основе файла проекта. Дайте команду make, чтобы скомпилировать программу и затем запустите ее, набрав в командной строке **hello** (в Windows) или **./hello** (в Unix) или **open hello.app** (в Mac OS X). Если вы работаете в Windows и используете Microsoft Visual C++, то вместо команды make вы должны дать команду nmake. Как альтернативный вариант -- вы можете создать проект Visual Studio из файла hello.pro, запустив команду:

```
qmake -tp vc hello.pro
```

и затем скомпилировать программу в Visual Studio.

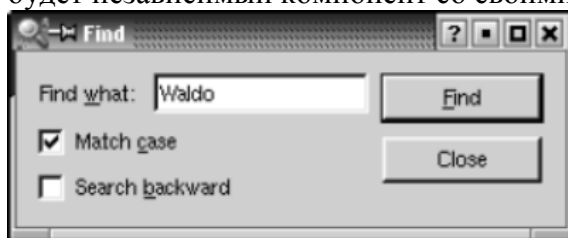


#### Практическая работа № 4

**Тема:** Создание дочернего класса от QDialog

**Цель:** разработка кода программного модуля, использующего создание дочернего класса от QDialog.

Диалог будет реализован в виде класса, со своей собственной функциональностью. Это будет независимый компонент со своими сигналами и слотами.



Исходные тексты приложения будут размещаться в двух файлах: finddialog.h и finddialog.cpp.

Начнем с файла finddialog.h.

```
1 #ifndef FINDDIALOG_H
2 #define FINDDIALOG_H
3 #include <qdialog.h>
4 class QCheckBox;
5 class QLabel;
6 class QLineEdit;
7 class QPushButton;
```

Строки 1 и 2 (и 27) предотвращают множественное подключение файла-заголовка.

В строке 3 подключается определение класса QDialog -- базового для всех диалогов в Qt. QDialog порожден от класса QWidget.

Строки с 4 по 7 -- это опережающие описания классов Qt, которые используются в нашем диалоге. **Опережающее описание** сообщает компилятору C++ о том, что этот класс существует, но подробности описания класса (обычно расположенного в отдельном заголовочном файле) здесь использоваться не будут. Ниже мы еще вернемся к этой теме.

Теперь определим класс FindDialog, указав в качестве родительского класса QDialog:

```
8 class FindDialog : public QDialog
9 {
```

## 10 Q\_OBJECT

11 public:

```
12 FindDialog(QWidget *parent = 0, const char *name = 0);
```

Определение класса начинается с вызова макроса **Q\_OBJECT**. Это обязательное требование для всех классов, которые определяют свои собственные сигналы и слоты.

Далее следует типичный, для всех виджетов в Qt, конструктор -- FindDialog. Параметр **parent** задает "владельца", т.е. виджет, на котором будет размещен данный компонент, а параметр **name** -- имя (название) виджета. Параметр **name** не является обязательным, в основном он используется для нужд отладки и тестирования.

13 signals:

```
14 void findNext(const QString &str, bool caseSensitive);
```

```
15 void findPrev(const QString &str, bool caseSensitive);
```

В секции **signals** описаны два сигнала, которые может посылать наше диалоговое окно, при нажатии на кнопку Find. Если включен флажок "Search backward" (поиск в обратном направлении), то посылается сигнал findPrev(), иначе -- findNext().

Ключевое слово **signals**, фактически является макроопределением. Препроцессор C++ преобразует его в стандартное представление C++ и только потом оно будет передано компилятору.

16 private slots:

```
17 void findClicked();
```

```
18 void enableFindButton(const QString &text);
```

19 private:

```
20 QLabel *label;
```

```
21 QLineEdit *lineEdit;
```

```
22 QCheckBox *caseCheckBox;
```

```
23 QCheckBox *backwardCheckBox;
```

```
24 QPushButton *findButton;
```

```
25 QPushButton *closeButton;
```

```
26 };
```

```
27 #endif
```

В приватной секции класса мы объявили два слота. Они необходимы для обеспечения взаимодействия с подчиненными виджетами, указатели на которые описаны чуть ниже. Ключевое слово **slots**, так же как и **signals**, является макроопределением.

Поскольку все поля-переменные, - это указатели, нам нет нужды подключать заголовочные файлы, содержащие полные определения этих классов. Благодаря наличию опережающего описания, компилятор удовлетворится тем, что есть. Вместо опережающего описания классов мы могли бы подключить соответствующие заголовочные файлы (<qcheckbox.h>, <qlabel.h>), но это отрицательно скажется на скорости компиляции. Для маленьких приложений это не так заметно, но для больших проектов опережающее описание может дать существенный выигрыш.

Перейдем к файлу finddialog.cpp, который содержит реализацию класса FindDialog:

```
1 #include <qcheckbox.h>
```

```
2 #include <qlabel.h>
```

```
3 #include <qlayout.h>
```

```
4 #include <qlineedit.h>
```

```
5 #include <qpushbutton.h>
```

```
6 #include "finddialog.h"
```

Здесь подключаются заголовочные файлы с описаниями используемых классов Qt. В строке 6 подключается определение нашего класса. Для большинства классов Qt, их определения

находятся в заголовочных файлах, имена которых повторяют имя класса (все символы в именах файлов переводятся в нижний регистр) и дополняются символами .h.

```
7 FindDialog::FindDialog(QWidget *parent, const char *name)
8   : QDialog(parent, name)
9 {
10  setCaption(tr("Find"));
11  label = new QLabel(tr("Find &what:"), this);
12  lineEdit = new QLineEdit(this);
13  label->setBuddy(lineEdit);
14  caseCheckBox = new QCheckBox(tr("Match &case"), this);
15  backwardCheckBox = new QCheckBox(tr("Search &backward"), this);
16  findButton = new QPushButton(tr("&Find"), this);
17  findButton->setDefault(true);
18  findButton->setEnabled(false);
19  closeButton = new QPushButton(tr("Close"), this);
```

В строке 8, конструктору базового класса передаются параметры **parent** и **name**.

В строке 10 задается надпись, которая будет выводиться в заголовке окна -- "Find". Функция tr() определена в классе QObject и любом другом подклассе, описание которого содержит вызов макроса **Q\_ОБЪЕКТ**. Она выполняет трансляцию текста, передаваемого ей, на другие языки человеческого общения. Считается хорошим тоном, все строки, которые будут выводиться на экран, передавать через эту функцию, даже в том случае, если вы не планируете интернационализацию своего приложения. Затем, начиная со строки 11, создаются подчиненные виджеты. Здесь символ амперсанда ('&') используется для обозначения клавиши быстрого доступа (акселератор). Например, в строке 16 создается кнопка Find, активировать которую можно нажатием на комбинацию клавиш Alt+F. Амперсанды так же могут использоваться для передачи фокуса ввода: в строке 11 создается метка, с акселератором (Alt+W), а в строке 13 метке назначается "дружественный" (buddy) компонент -- однострочное поле ввода. "**Дружественный**" виджет -- это такой компонент, который будет получать фокус ввода при нажатии ускоряющей комбинации клавиш метки. Таким образом, когда пользователь нажмет комбинацию клавиш Alt+W (акселератор метки), то фокус ввода будет передан "дружественному" виджету, т.е. -- полю ввода.

В строке 17, вызовом метода setDefault(true) [5], назначается кнопка по-умолчанию. Кнопка по-умолчанию -- это такая кнопка, которая будет активироваться при нажатии на клавишу Enter. В строке 18 накладывается запрет на кнопку Find. Когда виджет запрещен, он обычно отображается в серых тонах и не реагирует на действия пользователя.

```
20 connect(lineEdit, SIGNAL(textChanged(const QString &)),
21   this, SLOT(enableFindButton(const QString &)));
22 connect(findButton, SIGNAL(clicked()),
23   this, SLOT(findClicked()));
24 connect(closeButton, SIGNAL(clicked()),
25   this, SLOT(close()));
```

Приватный слот enableFindButton(const QString &) вызывается при изменении содержимого поля ввода. Приватный слот findClicked() вызывается, когда пользователь щелкает по кнопке Find. Работа приложения завершается после щелчка по кнопке Close. Слот close() наследуется от класса QWidget и его поведение по-умолчанию -- сокрытие виджета. Реализация слотов findClicked() и enableFindButton(const QString &) будет приведена ниже.

Поскольку QObject является одним из предков класса FindDialog, то мы можем опустить префикс QObject:: перед именем метода connect().

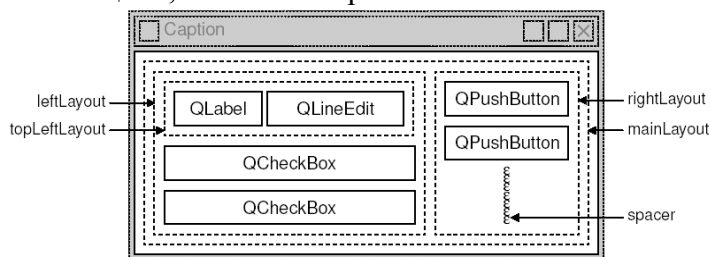
```
26 QHBoxLayout *topLeftLayout = new QHBoxLayout;
27 topLeftLayout->addWidget(label);
28 topLeftLayout->addWidget(lineEdit);
29 QVBoxLayout *leftLayout = new QVBoxLayout;
```

```

30 leftLayout->addLayout(topLeftLayout);
31 leftLayout->addWidget(caseCheckBox);
32 leftLayout->addWidget(backwardCheckBox);
33 QVBoxLayout *rightLayout = new QVBoxLayout;
34 rightLayout->addWidget(findButton);
35 rightLayout->addWidget(closeButton);
36 rightLayout->addStretch(1);
37 QHBoxLayout *mainLayout = new QHBoxLayout(this);
38 mainLayout->setMargin(11);
39 mainLayout->setSpacing(6);
40 mainLayout->addLayout(leftLayout);
41 mainLayout->addLayout(rightLayout);
42 }

```

На заключительном этапе выполняется выравнивание виджетов с помощью **менеджеров размещения**. Менеджер размещения (layout manager) -- это объект, который управляет размерами и положением виджетов. Qt предоставляет в наше распоряжение три менеджера размещения: `QHBoxLayout` выравнивает виджеты по горизонтали, `QVBoxLayout` -- по вертикали и `QGridLayout` -- по сетке. Менеджеры размещения (или, если хотите, менеджеры компоновки) могут содержать как отдельные виджеты, так и другие менеджеры размещения. Вкладывая друг в друга `QHBoxLayout`, `QVBoxLayout` и `QGridLayouts`, в различных комбинациях, можно выстроить весьма замысловатый интерфейс диалога.



В приложении мы использовали два `QHBoxLayout` и два `QVBoxLayout`. Внешний менеджер компоновки (`mainLayout`) является главным, поскольку при создании ему был назначен, в качестве владельца, экземпляр класса `FindDialog` -- (**this**). Он отвечает за размещение всех визуальных компонентов в области окна приложения. Оставшиеся 3 менеджера размещения являются подчиненными. Маленькая "пружинка", которая видна в правом нижнем углу рисунка -- это **распорка** (`spacer`). Она заполняет пустое пространство под кнопками Find и Close, заставляя их держаться в верхней части области выравнивания.

Здесь есть один важный момент. Менеджеры компоновки не являются виджетами. Они порождены от класса `QLayout`, который в свою очередь порожден от класса `QObject`. На рисунке, контуры виджетов отрисованы сплошными линиями, а областей компоновки -- пунктирными, чтобы подчеркнуть различия, имеющиеся между ними. Во время работы приложения, менеджеры размещения (области выравнивания) не видны.

Хотя менеджеры компоновки и не являются виджетами (визуальными компонентами), тем не менее они могут иметь как владельца, так и подчиненные компоненты. Понятия термина "владелец", у менеджера размещения и виджета, различаются. Если менеджер размещения встраивается в виджет (который передается менеджеру в качестве владельца), как это происходит в случае с **mainLayout**, то он автоматически встраивается в этот виджет. Если менеджер создается без владельца (в данном случае это: **topLeftLayout**, **leftLayout** и **rightLayout**), то он должен быть включен в состав другого менеджера, вызовом `addLayout()`.

Механизм "владелец-подчиненный" реализован в классе `QObject`, который является предком как для `QWidget`, так и для `QLayout`. Когда создается некий объект (виджет, менеджер компоновки или что-то еще), для которого явно указывается владелец, то он добавляется владельцем в свой список подчиненных компонентов. Когда владелец уничтожается, он проходит по списку подчиненных компонентов и уничтожает их одного за другим.

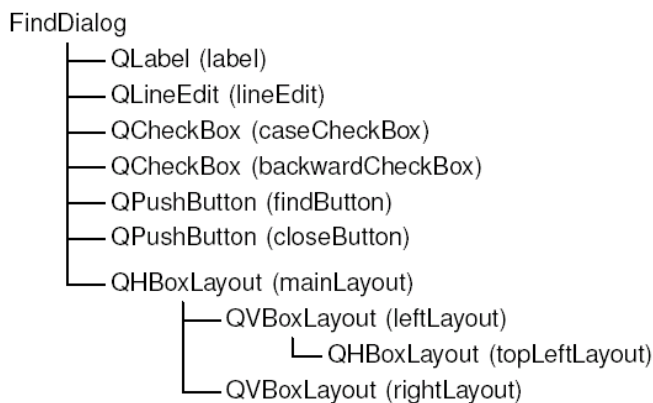
Подчиненные компоненты, в свою очередь просматривают свои списки и уничтожают компоненты, подчиненные им и так до тех пор, пока дело не дойдет до компонентов, которые не имеют подчиненных объектов.

Этот механизм упрощает управление памятью в приложении, снижая риск "утечки". Единственные объекты, которые необходимо уничтожать явно -- это те, которые были созданы оператором `new`, и не имеют владельца. Если первым удаляется подчиненный компонент, то Qt автоматически исключит его из списка владельца.

Владелец имеет особое значение для виджетов. Подчиненные виджеты отображаются на экране внутри области, принадлежащей его владельцу. Когда удаляется какой-либо владелец, он не только удалит подчиненные объекты из памяти, но и сотрет их на экране.

Когда один менеджер размещения вставляется в другой, с помощью функции `addLayout()`, то вложенный менеджер автоматически делается подчиненным объемлющему. В противоположность этому, когда в менеджер размещения вставляется виджет, вызовом `addWidget()`, то последний не меняет своего владельца.

На рисунке показано дерево "владелец-подчиненный" для приложения Find. Порядок взаимоотношений в этом дереве легко выводится из текста конструктора `FindDialog`, достаточно выделить строки, содержащие `new` и `addLayout()`. Важное замечание: Запомните, менеджеры размещения НЕ ЯВЛЯЮТСЯ владельцами виджетов, размещением которых они управляют.



В дополнение к менеджерам размещения, Qt предоставляет несколько **виджетов размещения**: `QHBoxLayout`, `QVBoxLayout` и `QGridLayout`. Для своих подчиненных компонентов, эти классы выступают как в качестве владельцев так и в качестве менеджеров размещения. Для небольших приложений виджеты размещения более удобны, но они менее гибкие и требуют больший объем ресурсов, чем менеджеры.

На этом мы заканчиваем рассмотрение реализации конструктора `FindDialog`. Поскольку подчиненные виджеты и менеджеры размещения создавались нами с помощью оператора `new`, то, казалось бы, возникает необходимость в написании деструктора, который удалил бы из памяти все, созданные нами компоненты, во время завершения приложения. Однако в этом нет необходимости, поскольку Qt автоматически сделает это во время уничтожения владельца, т.е. во время уничтожения класса `FindDialog`.

Перейдем к рассмотрению слотов:

```
43 void FindDialog::findClicked()
44 {
45     QString text = lineEdit->text();
46     bool caseSensitive = caseCheckBox->isOn();

47     if (backwardCheckBox->isOn())
48         emit findPrev(text, caseSensitive);
49     else
50         emit findNext(text, caseSensitive);
51 }

52 void FindDialog::enableFindButton(const QString &text)
```



```

53 {
54  findButton->setEnabled(!text.isEmpty());
55 }

```

Слот `findClicked()` вызывается всякий раз, когда пользователь щелкает мышкой по кнопке Find. Кнопка выдает сигнал `findPrev()` или `findNext()`, в зависимости от состояния флажка Search backward. Ключевое слово **emit** является макросом, определенным в библиотеке Qt.

Слот `enableFindButton()` вызывается, когда пользователь изменяет содержимое поля ввода. Если оно содержит какие-либо символы, то разрешается кнопка Find, в противном случае она запрещается.

Реализацией этих двух слотов завершается разработка нашего диалога. Теперь создадим файл `main.cpp`, в котором разместим текст основной программы для тестирования нашего виджета:

```

1 #include <qapplication.h>

2 #include "finddialog.h"

3 int main(int argc, char *argv[])
4 {
5     QApplication app(argc, argv);
6     FindDialog *dialog = new FindDialog;
7     app.setMainWidget(dialog);
8     dialog->show();
9     return app.exec();
10 }

```

дадим команду `qmake`, как обычно. На этот раз, поскольку наш класс `FindDialog` содержит вызов макроопределения **Q\_OBJECT**, утилита `qmake` включит в `Makefile` правила, вызывающие утилиту `moc -- компилятор метаобъектов`.

Для корректной работы утилиты `moc` необходимо, чтобы описание класса размещалось в заголовочном файле, отдельно от файла с реализацией. Код, сгенерированный утилитой `moc` подключает этот заголовочный файл.

Классы, в определении которых встречается макрос **Q\_OBJECT**, должны обрабатываться компилятором метаобъектов в обязательном порядке. На самом деле, это не такая большая проблема, поскольку `qmake` автоматически добавит все необходимые правила в `Makefile`. Но если вы забудете регенерировать `Makefile`, то линковщик будет "жаловаться" на отсутствие некоторых функций. Эти сообщения об ошибках могут порой вводить в заблуждение. Например, GCC выдает примерно такое предупреждение:

```

finddialog.o(.text+0x28): undefined reference to
FindDialog::QPaintDevice virtual table

```

Visual C++ такое:

```

finddialog.obj : error LNK2001: unresolved external symbol
"public:~virtual bool __thiscall FindDialog::qt_property(int,
int,class QVariant *)"

```

Если это произошло, то перезапустите `qmake` еще раз, чтобы обновить `Makefile`, а затем пересоберите приложение.

Теперь запустите программу. Проверьте работу акселераторов `Alt+W`, `Alt+C`, `Alt+B` и `Alt+F`. Попробуйте "пройтись" по виджетам с помощью клавиши `Tab`. По-умолчанию, порядок навигации с помощью клавиши `Tab`, соответствует порядку, в котором создавались компоненты. Но он может быть изменен вызовом метода `QWidget::setTabOrder()`. Установку акселераторов и настройку порядка навигации по компонентам, с помощью клавиши `Tab`, можно считать дружественным жестом в сторону пользователей, которые не могут или не

хотят пользоваться мышью. Удобное управление с клавиатуры высоко оценят опытные пользователи.

Мы будем использовать наш диалог в реально работающем приложении. Там сигналы `findPrev()` и `findNext()` будут подключаться к соответствующим слотам.

## Практическая работа № 5

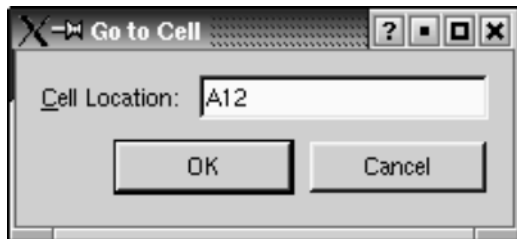
**Тема: Быстрая разработка диалогов.**

**Цель:** разработка кода программного модуля с помощью **Qt Designer**.

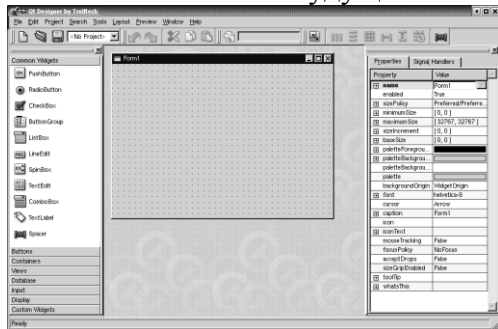
Qt разрабатывалась так, чтобы можно было писать код программы вручную без особого напряжения. Тем не менее, **Qt Designer** еще больше расширяет возможности программиста, предоставляя ему возможность визуального дизайна.

С помощью **Qt Designer**, напишем диалог Go-to-Cell ("Перейти к ячейке"), показанный на рисунке ниже. Совершенно неважно, как разрабатывается диалог - вручную ли, или с помощью **Qt Designer**, всегда выполняется одна и та же последовательность действий:

- Создаются и инициализируются подчиненные виджеты.
- Подчиненные виджеты вставляются в менеджеры размещения.
- Настраивается порядок навигации по виджетам клавишей Tab.
- Устанавливаются соединения сигнал-слот
- Реализуются дополнительные слоты диалога, если это необходимо.

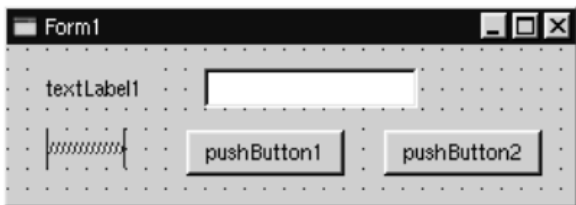


Чтобы запустить **Qt Designer**, выберите пункт Qt 3.2.x|Qt Designer в меню "Пуск" - в ОС Windows. После того, как программа запустится, она предложит на выбор список шаблонов. Щелкните по шаблону "**Dialog**" и нажмите кнопку ОК. После этого перед вами должна появиться заготовка будущего окна диалога с именем "Form1".



Для начала разместим виджеты на форме. На инструментальной панели, слева, щелкните по компоненту `TextLabel`, затем щелкните по форме - в результате на форме появится компонент "метка". Аналогичным образом разместите на форме одно поле ввода (`LineEdit`), одну горизонтальную распорку (`Spacer`) и две кнопки (`PushButton`). Разместите их так, чтобы у вас получилось нечто похожее на рисунке.. Не тратьте слишком много времени на позиционирование виджетов. Мы все равно будем использовать менеджеры размещения, которые выполняют эту работу за нас.

Распорка (`spacer`) отображается на заготовке в виде синей пружинки. Во время работы уже готовой программы распорки не будут отображаться.



Установите свойства для каждого из виджетов, используя **Редактор свойств**, расположенный в правой части главного окна **Qt Designer**.

1. Щелкните по компоненту TextLabel и запишите в его свойство **name** строку "label", а в свойство **text** -- "&Cell Location:".
2. Щелкните по компоненту LineEdit и запишите в свойство **name** строку "lineEdit".
3. Для распорки запишите в свойство **orientation** "Horizontal".
4. Для первой кнопки запишите в свойство **name** строку "okButton", в свойство **enabled** -- "False", в свойство **default** -- "True" и в свойство **text** -- "OK".
5. Для второй кнопки. Запишите в свойство **name** строку "cancelButton", а в свойство **text** -- в "OK".
6. Щелкните в любом свободном месте формы и запишите в свойство **name** строку "GoToCellDialog", а в свойство **caption** -- "Go to Cell".

Но это еще не все, нам нужно назначить дружественный компонент для метки, который будет реагировать на акселератор Alt+C. На данный момент метка отображается как "&Cell Location:". Выберите пункт меню Tools|Set Buddy (курсор мыши приобретет вид крестика). Затем поместите указатель мыши на метку, нажмите левую кнопку и, удерживая ее в нажатом положении, переместите указатель мыши на компонент LineEdit. Затем отпустите кнопку мыши. Изображение метки изменится -- символ амперсанда исчезнет, а первый символ метки приобретет знак подчеркивания. В принципе, то же самое можно сделать внутри редактора свойств, установкой свойства **buddy** метки.



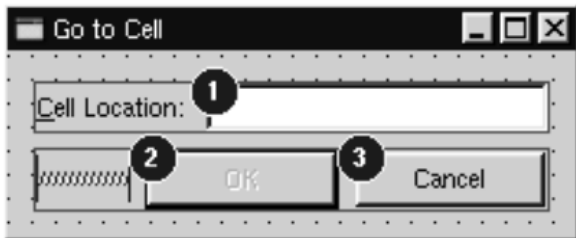
Следующий шаг -- размещение виджетов на форме с помощью менеджеров компоновки:

1. Щелкните мышью по метке. Нажмите клавишу Ctrl и удерживая ее -- щелкните по полю ввода. Оба компонента окажутся выделенными. Теперь выберите пункт меню Layout|Lay Out Horizontally.
2. Щелкните мышью по распорке. Нажмите клавишу Shift и удерживая ее -- щелкните сначала по кнопке "OK", а затем по кнопке "Cancel". Теперь выберите пункт меню Layout|Lay Out Horizontally.
3. Щелкните по свободному пространству на форме и выберите пункт меню Layout|Lay Out Vertically.
4. Выберите пункт меню Layout|Adjust Size.

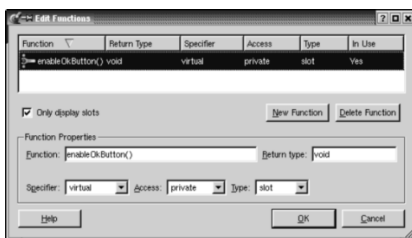
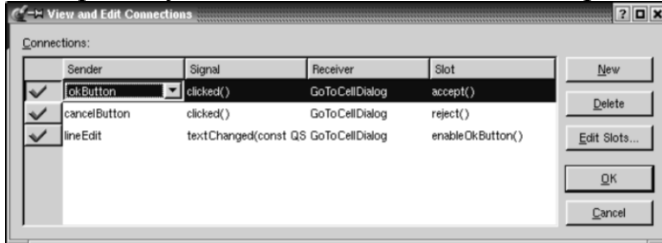
Красные контуры областей выравнивания, которые сейчас видны на заготовке, во время работы программы отображаться не будут.



Теперь выберите пункт меню Tools|Tab Order. На каждом из виджетов, которые могут принимать фокус, появятся цифры в синих кружочках. Щелчками мыши по компонентам установите желаемый порядок навигации клавишей Tab и нажмите Esc.



На этом дизайн внешнего вида формы можно считать завершенным. Теперь перейдем к настройке функциональной части -- свяжем сигналы и слоты и создадим свой слот. Выберите пункт меню **Edit|Connections**, перед вами откроется окно редактора связей:



Для начала создадим новый слот: щелкните по кнопке **Edit Slots...**. Перед вами откроется окно редактора слотов. Создайте приватный слот с именем `enableOkButton()`.

Затем необходимо настроить три соединения. Чтобы создать соединение -- щелкните по кнопке "New" и установите поля **Sender**, **Signal**, **Receiver** и **Slot**, выбирая требуемые значения из выпадающих списков в каждом из них. У вас должно получиться следующее:

```
okButton    clicked()          GoToCellDialog  accept()
cancelButton clicked()          GoToCellDialog  reject()
lineEdit    textChanged(const QString &) GoToCellDialog  enableOkButton()
```

Чтобы посмотреть, как будет выглядеть окно диалога во время работы программы -- выберите пункт меню **Preview|Preview Form**. Проверьте порядок навигации клавишей **Tab**. Проверьте работу акселератора **Alt+C** (поле ввода должно получить фокус ввода). Нажмите кнопку **Cancel**, чтобы закрыть окно.

Сохраните результаты работы в файле `gotocelldialog.ui` в каталоге `gotocell` и создайте файл `main.cpp`, в том же каталоге, с помощью обычного текстового редактора:

```
#include <qapplication.h>
#include "gotocelldialog.h"
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    GoToCellDialog *dialog = new GoToCellDialog;
    app.setMainWidget(dialog);
    dialog->show();
    return app.exec();
}
```

Создайте файл проекта и **Makefile** утилитой `qmake` (`qmake -project; qmake gotocell.pro`). Утилита `qmake` сама обнаружит файл `gotocelldialog.ui` и добавит в **Makefile** все необходимые правила по созданию `gotocelldialog.h` и `gotocelldialog.cpp`. Все `.ui` файлы преобразуются в код **C++** с помощью утилиты `uic` (**User Interface Compiler** -- **Компилятор Пользовательских Интерфейсов**).

Вся прелесть **Qt Designer**-а состоит в том, что вы можете свободно изменять дизайн формы без необходимости вторгаться в исходный код на C++. Когда разработка дизайна ведется в тексте программы (вручную) то это может отнять довольно значительное время. **Qt Designer** сохранит ваши силы и время.

Если теперь запустить программу, то вы заметите:

- Кнопка "ОК" всегда остается запрещенной.
- Поле ввода принимает не только те символы, из которых может состоять номер искомой ячейки, но и любые другие.

Мы должны решить эти проблемы.

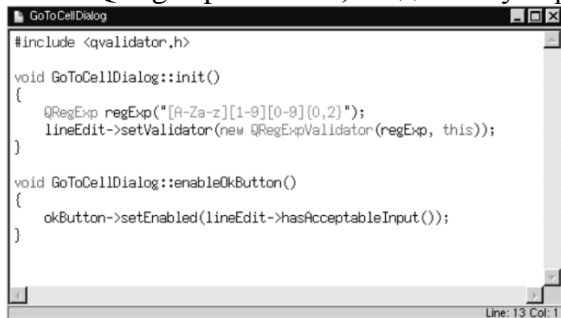
Щелкните дважды по свободному пространству на форме, чтобы вызвать редактор исходного кода. В окне редактора добавьте следующие строки:

```
#include <qvalidator.h>
void GoToCellDialog::init()
{
    QRegExp regExp("[A-Za-z][1-9][0-9]{0,2}");
    lineEdit->setValidator(new QRegExpValidator(regExp, this));
}
void GoToCellDialog::enableOkButton()
{
    okButton->setEnabled(lineEdit->hasAcceptableInput());
}
```

Функция `init()` автоматически вызывается конструктором формы (конструктор генерируется утилитой `uic`). Она настраивает проверку корректности ввода для `LineEdit`. Qt предоставляет три класса, выполняющих проверку на корректность: `QIntValidator`, `QDoubleValidator` и `QRegExpValidator`. Для своих нужд мы будем использовать последний, который будет выполнять проверку на основе регулярного выражения: `"[A-Za-z][1-9][0-9]{0,2}"`. Это выражение означает: "Позволить ввод одного алфавитного символа в верхнем или нижнем регистре, за которым должна следовать одна цифра, в диапазоне от 1 до 9, за которой может следовать до двух цифр, в диапазоне от 0 до 9".

Передавая аргумент **this** (в вызов конструктора `QRegExpValidator()`), мы делаем объект класса `QRegExpValidator` подчиненным, по отношению к `GoToCellDialog`. Таким образом мы снимаем с себя ответственность за удаление этого объекта из памяти по завершении работы приложения.

Слот `enableOkButton()` разрешает или запрещает кнопку "ОК", в зависимости от того, насколько правильный номер ячейки содержится в поле ввода. Для проверки правильности используется функция `QLineEdit::hasAcceptableInput()`, которая обращается к объекту класса `QRegExpValidator`, созданному в функции `init()`.



После этого опять сохраните диалог. **Qt Designer** сохранит оба файла -- и `gotocelldialog.ui`, и `gotocelldialog.ui.h`. Пересоберите приложение и запустите его. Введите в поле ввода строку "A12" -- кнопка "ОК" перейдет в разрешенное состояние. Попробуйте набрать произвольный текст и понаблюдайте за тем, как работает проверка корректности ввода. Нажмите кнопку "Cancel", чтобы завершить работу программы.

В этом примере мы создали диалог с помощью **Qt Designer** и добавили некоторый код, с помощью редактора исходного кода **Qt Designer**-а. Интерфейсная часть диалога была сохранена в файле `gotocelldialog.ui` (по сути файл формата XML), а исходный текст -- в

файл `gotocelldialog.ui.h`. Это очень удобно, поскольку `gotocelldialog.ui.h` можно править в любом текстовом редакторе.

Альтернативный подход заключается в разработке формы с помощью **Qt Designer** (как обычно), а затем создается дополнительный класс, порожденный от класса формы, в котором реализуется вся необходимая функциональность. Например, для нашего диалога `Go-to-Cell` можно было бы создать класс `GoToCellDialogImpl`, как наследник класса `GoToCellDialog` и реализовать в нем все необходимые функции. В результате такого подхода, новый заголовочный файл должен получиться таким:

```
#ifndef GOTOCELLDIALOGIMPL_H
#define GOTOCELLDIALOGIMPL_H
#include "gotocelldialog.h"
class GoToCellDialogImpl : public GoToCellDialog
{
    Q_OBJECT
public:
    GoToCellDialogImpl(QWidget *parent = 0, const char *name = 0);
private slots:
    void enableOkButton();
};
#endif
```

А файл с реализацией:

```
#include <qlineedit.h>
#include <qpushbutton.h>
#include <qvalidator.h>
#include "gotocelldialogimpl.h"
GoToCellDialogImpl::GoToCellDialogImpl(QWidget *parent,
                                         const char *name)
    : GoToCellDialog(parent, name)
{
    QRegExp regExp("[A-Za-z][1-9][0-9]{0,2}");
    lineEdit->setValidator(new QRegExpValidator(regExp, this));
}
void GoToCellDialogImpl::enableOkButton()
{
    okButton->setEnabled(lineEdit->hasAcceptableInput());
}
```

Создание классов-наследников может быть упрощено с помощью утилиты `uic` и набора дополнительных аргументов командной строки. Так, например, утилита `uic`, с ключом **-subdecl**, создаст скелетон заголовочного файла, а с ключом **-subimpl** -- соответствующий файл реализации.

В данной книге мы будем работать только с файлами `.ui.h`, поскольку это наиболее общеупотребимая практика, а создание дочерних классов, с помощью `uic`, довольно простая задача. Чтобы поглубже разобраться в различиях этих двух подходов, рекомендуем прочитать главу "Designer Approach" в справочном руководстве, поставляемом вместе с **Qt Designer**. Кроме того, прочитайте главу "Creating Dialogs", где показывается, как можно использовать вкладку "Members" для создания полей (переменных-членов) в классе формы.

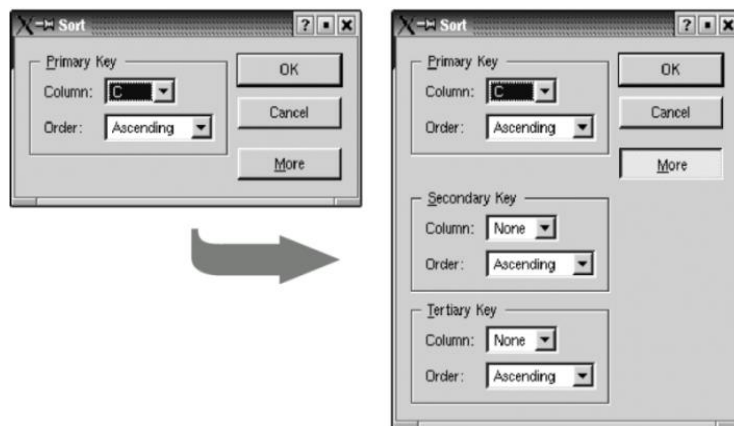
## **Практическая работа № 6**

**Тема:** Диалоги с изменяющимся внешним видом.

**Цель:** Разработка кода программного модуля с использованием *Qt Designer* для создания диалога.

В некоторых случаях желательно иметь диалоги, которые могут динамически изменять свое представление. Наиболее часто на практике встречаются расширяемые диалоги и многостраничные диалоги. Оба вида диалогов могут быть созданы как в **Qt Designer**, так и в результате ручного кодирования.

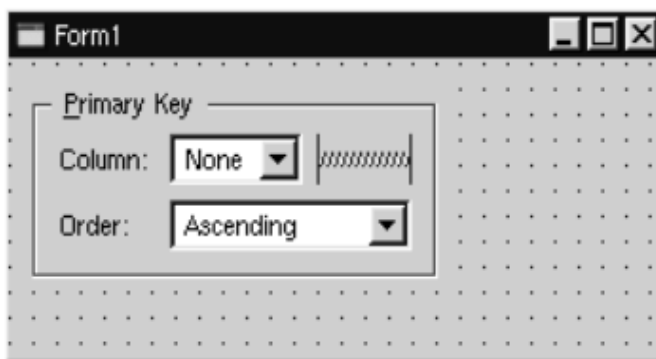
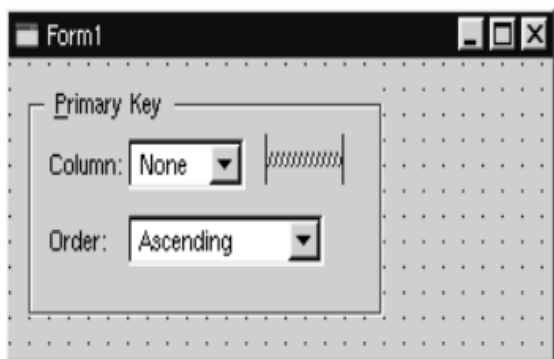
Расширяемые диалоги, обычно выводятся на экран в сокращенном варианте, но дают пользователю возможность выбирать между сокращенным и расширенным режимом представления информации. Расширяемые диалоги как правило используются в тех случаях, когда необходимо скрыть дополнительные сведения, которые не являются обязательными и пользователь явно не выразил свое желание видеть их. В этом разделе мы разберем процесс создания расширяемого диалога, показанного на рисунке ниже с помощью **Qt Designer**.



Это диалог сортировки, используемый в электронных таблицах. Он появляется, когда пользователь пытается отсортировать данные по одному или нескольким столбцам. В сокращенном представлении диалог позволяет выбрать столбец и порядок сортировки, в расширенном варианте добавляется возможность указать еще два столбца и порядок сортировки по каждому из них. Кнопка "More" позволяет переходить из сокращенного режима -- в расширенный и обратно.

Мы создадим диалог в его расширенном виде, а потом, во время исполнения, будем скрывать дополнительные виджеты, чтобы обеспечить краткую форму диалога.

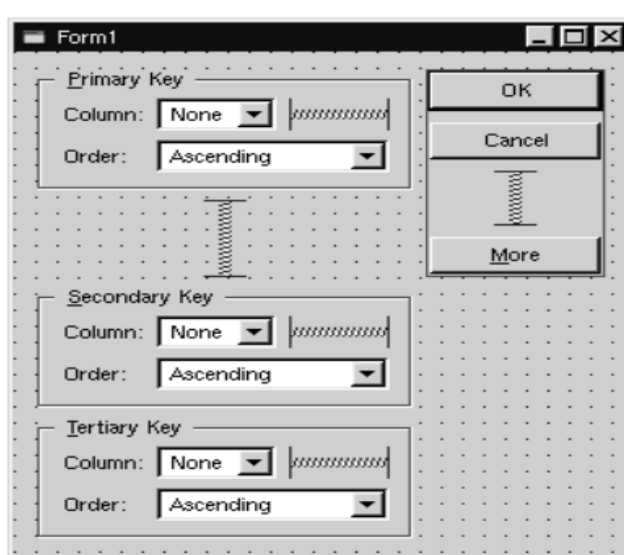
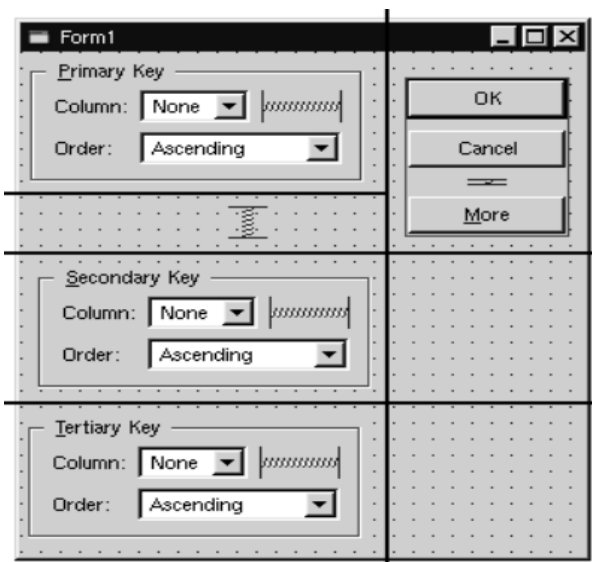
1. Положите на заготовку формы `GroupBox`, два `TextLabel`, два `ComboBox` и одну горизонтальную распорку.
2. "Растяните" `GroupBox` побольше, ухватив мышкой за правый нижний его угол.
3. Разместите виджеты внутри `GroupBox`-а примерно так, как показано на рисунке ниже.
4. Ухватив мышкой за правый край второго `ComboBox`-а, сделайте его примерно в два раза больше первого.
5. Запишите в свойство **title**, `GroupBox`-а, строку "&Primary Key". В свойство **text** первой метки -- "Column:", второй метки -- "Order:".
6. Щелкните мышкой дважды по первому `ComboBox`. Перед вами появится окно редактора, в котором добавьте один элемент с текстом "None".
7. Щелкните мышкой дважды по второму `ComboBox` и добавьте элементы "Ascending" и "Descending".
8. Теперь скомпилируем виджеты внутри `GroupBox`, для этого, щелкните по `GroupBox` и выберите пункт меню `Layout|Lay Out in a Grid`. В результате вы должны получить нечто похожее на рисунок.



Если компоновка выполнена не так как надо или вы допустили какую-нибудь ошибку, вы всегда можете выбрать пункт меню Edit|Undo и отменить произведенное действие. После чего можете повторить попытку.

Теперь добавим группы виджетов для расширенного представления:

1. Растяните форму диалога, чтобы хватило места для дополнительных виджетов. Выберите GroupBox и скопируйте его в буфер обмена, выбрав пункт меню Edit|Copy. Вставьте новые группы виджетов, дважды выбрав пункт меню Edit|Paste. Переместите новые GroupBox-ы на свои места. Измените у них свойство **title**.
2. Создайте кнопки "OK", "Cancel" и "More".
3. Для кнопки "OK" установите свойство **default** в TRUE.
4. Добавьте две вертикальные распорки.
5. Расположите кнопки "OK", "Cancel" и "More" по вертикали. Переместите одну из распорок так, чтобы она встала между кнопками "Cancel" и "More". Выделите все четыре элемента и выберите пункт меню Layout|Lay Out Vertically.
6. Вторую распорку переместите так, чтобы она встала между первым и вторым GroupBox.
7. Установите свойство **sizeHint** у вертикальных распорок в значение (20, 10).
8. Разместите виджеты так, как это показано на рисунке.
9. Выберите пункт меню Layout|Lay Out in a Grid. У вас должна получиться заготовка, показанная на рисунке.

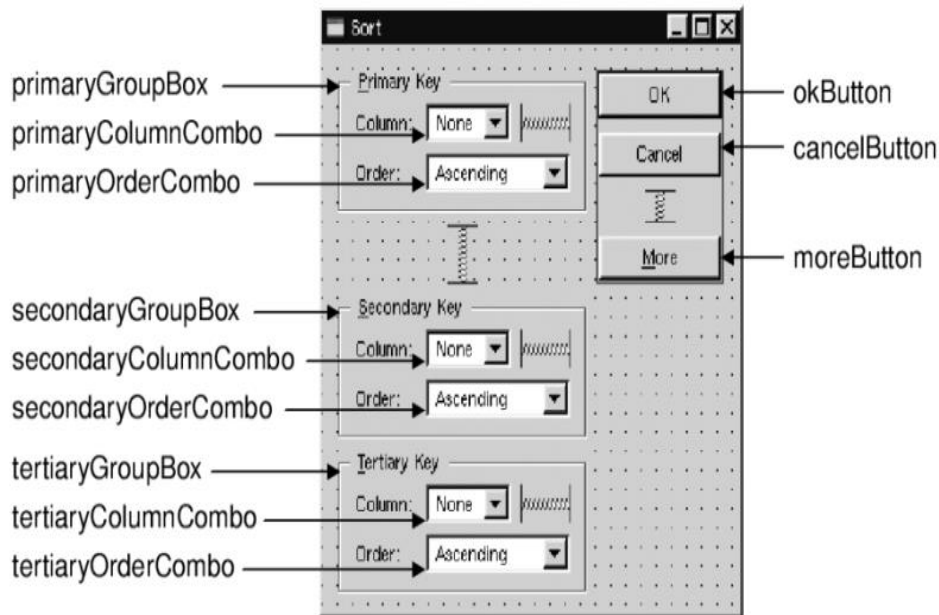


В результате такого размещения мы получили "сетку" из двух колонок и четырех строк -- всего восемь ячеек. Первый GroupBox, левая вертикальная распорка, второй и третий GroupBox занимают по одной ячейке. Кнопки "OK", "Cancel", "More" и правая вертикальная распорка занимают две ячейки. И в правом нижнем углу диалога у нас остаются две пустых ячейки. Если у вас не получилась такая компоновка виджетов -- отмените ее и повторите попытку.



Проверьте свойство формы **resizeMode**. Оно должно быть установлено как "Fixed", благодаря чему пользователь не сможет растягивать окно диалога. Тогда, всю ответственность за размер окна диалога возьмут на себя менеджеры размещения, изменяя его в случае, когда подчиненные виджеты показываются или скрываются. Это гарантирует показ окна диалога с оптимальными размерами.

Дайте форме имя "SortDialog" и установите свойство **caption** в "Sort". Дайте имена виджетам, в соответствии с рисунком.



В заключение соединим сигналы и слоты:

1. Соедините **okButton clicked()** с **SortDialog accept()**
2. Соедините **cancelButton clicked()** с **SortDialog reject()**
3. Соедините **moreButton toggled(bool)** с **secondaryGroupBox setShown(bool)**
4. Соедините **moreButton toggled(bool)** с **tertiaryGroupBox setShown(bool)**

Щелкните мышкой по форме дважды, чтобы запустить редактор и добавьте следующий текст:

```

1 void SortDialog::init()
2 {
3     secondaryGroupBox->hide();
4     tertiaryGroupBox->hide();
5     setColumnRange('A', 'Z');
6 }

7 void SortDialog::setColumnRange(QChar first, QChar last)
8 {
9     primaryColumnCombo->clear();
10    secondaryColumnCombo->clear();
11    tertiaryColumnCombo->clear();

12    secondaryColumnCombo->insertItem(tr("None"));
13    tertiaryColumnCombo->insertItem(tr("None"));

14    primaryColumnCombo->setMinimumSize(
15        secondaryColumnCombo->sizeHint());
16    QChar ch = first;
17    while (ch <= last) {
18        primaryColumnCombo->insertItem(ch);
19        secondaryColumnCombo->insertItem(ch);

```

```

20     tertiaryColumnCombo->insertItem(ch);
21     ch = ch.unicode() + 1;
22 }
23 }

```

Функция `init` делает второй и третий `GroupBox` невидимыми.

Слот `setColumnRange()` инициализирует содержимое выпадающих списков, в соответствии с именами выделенных колонок в электронной таблице. Мы вставили элемент "None", во второй и третий выпадающий списки, на тот случай, если пользователь пожелает выполнить сортировку только по одному столбцу. Не смотря на то, что мы не создавали это слот в **Qt Designer**, тем не менее он его обнаружит самостоятельно, а `uis` создаст соответствующее объявление в определении класса `SortDialog`.

В строках 14 и 15 можно наблюдать один хитрый трюк, связанный с размещением компонента. Функция `QWidget::sizeHint()` возвращает "идеальный" размер виджета, который пробует соблюсти система размещения. Дело в том, что виджетам с различным содержимым могут быть заданы различные размеры. Для выпадающих списков это означает, что второй и третий списки, содержащие слово "None", могут иметь больший размер, чем первый, в котором указано односимвольное имя столбца. Чтобы избежать такой несогласованности, мы задаем минимальный размер, для первого выпадающего списка, равный "идеальному" размеру второго.

Ниже приводится текст функции `main()`, которая устанавливает диапазон выделенных столбцов от "C" до "F" и затем вызывает диалог:

```

#include <qapplication.h>
#include "sortdialog.h"
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    SortDialog *dialog = new SortDialog;
    app.setMainWidget(dialog);
    dialog->setColumnRange('C', 'F');
    dialog->show();
    return app.exec();
}

```

На этом мы завершаем рассмотрение расширяемого диалога. Из примера видно, что разработка расширяемых диалогов ненамного сложнее, чем обычных. Все что нам потребовалось добавить -- это кнопка перехода из режима в режим, несколько дополнительных сигналов и слотов, и фиксированный размер формы.

Другой тип диалогов, изменяющих свое представление -- это многостраничные диалоги. Создание многостраничных диалогов проходит еще проще. Такого рода диалоги можно строить несколькими способами:

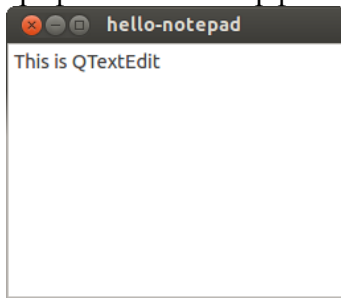
- В качестве основы можно использовать `QTabWidget`. Сверху он имеет набор вкладок, которые управляются встроенным `QWidgetStack`.
- Можно использовать связку `QListBox` и `QWidgetStack`, в которой текущий элемент `QListBox`-а определяет страницу в `QWidgetStack`.
- Или связку из классов `QListView` или `QIconView` и `QWidgetStack`, объединяемые так же как и в случае с `QListBox`.

## Практическая работа № 7

**Тема:** графический интерфейс Qt.

**Цель:** разработка кода программного модуля, создающее простое окно для редактирования текста.

Создадим простое окно для редактирования текста. Это самая элементарная программа с графическим интерфейсом Qt.



Исходный код примера:

```
#include <QApplication>
#include <QTextEdit>
int main(int argv, char **args)
{
    QApplication app(argv, args);
    QTextEdit textEdit;
    textEdit.show();
    return app.exec();
}
```

Рассмотрим подробно каждую строчку кода. В первых двух, мы подключаем заголовочные файлы классов [QApplication](#) и [QTextEdit](#), которые необходимы для работы этого примера. Каждому классу в Qt соответствует заголовочный файл с таким же названием.

В строке 6 создается объект класса [QApplication](#), который управляет основными ресурсами, необходимыми для запуска любой программы с графическим интерфейсом Qt. Ему необходимо передать аргументы argv и args функции main(), т.к. Qt использует некоторые параметры командной строки, передаваемые при запуске программы.

В восьмой строке кода создается объект класса [QTextEdit](#). [QTextEdit](#) — это визуальный элемент графического интерфейса. В Qt используются определенные виджеты — например, полосы прокрутки, метки и кнопки [radio](#). Виджет может быть контейнером для хранения других виджетов. Наглядным примером является главное диалоговое окно программы.

В строке 9, окно редактирования текста выводится на экран в главном фрейме программы. Поскольку виджеты могут работать как контейнеры (В экземпляре класса [QMainWindow](#) находятся полосы панели инструментов, меню, строка состояния и несколько других виджетов), мы можем отобразить только один виджет в окне нашей программы. По умолчанию, виджеты не видны. Функция [show\(\)](#) используется для их отображения.

В строке 11, создается объект [QApplication](#), который генерирует цикл событий в процессе работы приложения. События генерируются и передаются на обработку виджетам. Примерами событий могут являться клики мыши, нажатия клавиш на клавиатуре и т.п. Когда вы вводите текст в окне редактирования виджета [QTextEdit](#), нажатия клавиш обрабатываются средствами виджета, и вводимый текст отображается в процессе набора.

Для запуска программы, откройте командную строку и зайдите в директорию с .cpp файлом программы. Выполните следующие команды shell-интерпретатора, чтобы скомпилировать пример.

```
Qmake -project
Qmake
Make
```

После успешного выполнения предыдущих команд, скомпилированная программа будет размещена в директории текущего проекта (в Windows вы можете использовать `nmake` вместо `make`. Исполняемые файлы будут размещены в директориях `debug` или `release`, которые

создадутся командой `make`. `qmake`— это утилита, которая создает файлы конфигурации Qt-проекта, если ей передан аргумент-`project`. После создания файла конфигурации (`.pro`), `qmake` генерирует `Makefile`, который используется утилитой `make` для сборки приложения. Позже, мы рассмотрим процесс написания собственных `.pro` файлов.

## **Практическая работа № 8**

**Тема:** Добавление кнопки выхода

**Цель:** разработка кода программного модуля, использующего виджет `QTextEdit` и кнопку выхода `QPushButton`.

В реальных проектах, обычно используется более одного виджета. Добавим кнопку `QPushButton` под полем редактирования текста. При клике на нее, программа будет завершаться.



Рассмотрим исходный код программы.

```
#include <QtGui>
//#include <QTextEdit>
int main(int argc, char **argv)
{
    QApplication app(argc, argv);
    QTextEdit textEdit;
    QPushButton quitButton("Quit");
    QObject::connect(&quitButton, SIGNAL(clicked()), qApp, SLOT(quit()));
    QVBoxLayout layout;
    layout.addWidget(&textEdit);
    layout.addWidget(&quitButton);
    QWidget window;
    window.setLayout(&layout);
    window.show();

    return app.exec();
}
```

Сначала подключается заголовочный файл `QtGui`. В нем содержатся все классы графического интерфейса Qt.

В строке 10 применяется механизм сигналов и слотов, для того, чтобы закрыть программу после нажатия на кнопку выхода. Слот — это функция, которая может быть вызвана в процессе выполнения программы. Сигнал — функция, вызывающая функции-слоты, которые с ним посредством `QObject::connect`.

`quit()` — слот класса `QApplication`, завершающий работу приложения. `clicked()` — сигнал, который передает нажатая кнопка `QPushButton`. Статическая функция `QObject::connect` связывает определенный слот и сигнал. `SIGNAL()` и `SLOT()` — макросы, которые принимают сигнатуры сигналов и слотов, для их связи между собой. Также,

функции connect() необходимо передать указатели на объекты, которые будут принимать и рассылать сигналы.

В строке 12 создается объект класса QVBoxLayout. Как уже было сказано, виджеты могут содержать в себе другие виджеты. Можно задавать координаты и размер вложенных виджетов непосредственно, но обычно для этого используют слои (layouts). Слой управляет границами вложенных виджетов. Например, объект QVBoxLayout размещает виджеты в одну вертикальную колонку.

В строках 13 и 14, мы добавляем поле редактирования текста и кнопку выхода к слою layout.

В строке 17 задается главный слой для всего приложения.

## **Практическая работа № 9**

**Тема:** Наследование QWidget

**Цель:** разработка кода программного модуля с наследником QWidget и добавление в него слота, который будет привязан к кнопке выхода.

Когда пользователь хочет закрыть приложение, ему можно показать всплывающее окно, в котором программа спросит подтверждение данного действия. Создадим наследник QWidget и добавим в него слот, который будет привязан к кнопке выхода.



Рассмотрим следующий код.

```
Class Notepad: public QWidget
```

```
{ Q_OBJECT
```

```
Public:
```

```
Notepad();
```

```
Private slots:
```

```
    void quit();
```

```
Private:
```

```
    QTextEdit *textEdit;
```

```
    QPushButton *quitButton;
```

```
};
```

Макрос Q\_OBJECT объявляет класс как QObject. Он должен находиться в самом начале определения класса. QObject добавляет несколько расширенных возможностей к обычному классу C++. Например, имена классов и слотов можно запросить в процессе выполнения. В строке 13 объявляется слот quit(). В последствии, мы присоединим этот слот к сигналу.

В предыдущих примерах, создание графики и присоединение слотов осуществлялось внутри функции main(). Сейчас мы будем использовать конструктор Notepad.

```
Notepad::Notepad()
```

```
{
```

```
    textEdit = new QTextEdit;
```

```
    quitButton = new QPushButton(tr("Quit"));
```

```
    connect(quitButton,SIGNAL(clicked()),this,SLOT(quit()));
```

```
    QVBoxLayout * layout = new QVBoxLayout;
```

```
    Layout->addWidget(textEdit);
```

```
    Layout->addWidget(quitButton);
```

```
    setLayout(layout);
```

```
    setWindowTitle(tr("Notepad"));
```

```
}
```

Как видно из определения класса, мы используем указатели на объекты `textEdit` и `quitButton`. Для объектов `QObject`, почти всегда рациональнее выделять память в куче, вместо их копирования.

Мы используем функцию `tr()` для обработки всех строк, которые видны пользователю. Эта функция применяется, когда приложение нужно перевести на несколько языков. Здесь мы не будем углубляться в детали, но вы можете найти нужную информацию в описании библиотеки `Qt Linguist`.

Создание файлов `.pro`

Ранее, мы использовали команду `qmake -project` для создания файлов `.pro`. В следующем примере, мы создадим этот файл вручную.

```
HEADERS=notepad.h
```

```
SOURCES=notepad.cpp\  
main.cpp
```

Теперь для сборки программ на Qt, будут использоваться следующие команды.

```
qmake;
```

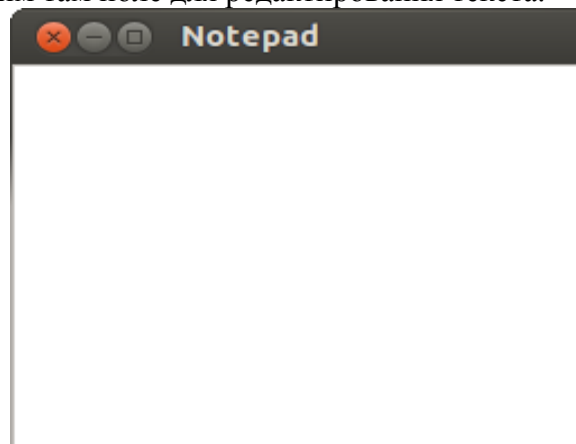
```
make;
```

### Практическая работа № 10

**Тема:** Использование `QMainWindow`.

**Цель:** разработка кода программного модуля, с использованием виджета `QMainWindow`.

Во многих приложениях удобно использовать `QMainWindow` — слой, в который можно добавлять контекстные меню, встроенные виджеты, панели инструментов и строку состояния. В `QMainWindow` есть центральная область, куда можно добавлять любые дочерние виджеты. В этом примере мы разместим там поле для редактирования текста.



Рассмотрим определение класса `Notepad`.

```
#include <QtGui>
```

```
class Notepad : public QMainWindow
```

```
{
```

```
    Q_OBJECT;
```

```
public :
```

```
    Notepad();
```

```
private slots:
```

```
    void open();
```

```

void save();
void quit();

private :
    QTextEdit *textEdit;
    QAction *saveAction;
    QAction *exitAction;

    QMenu *fileMenu;

};

```

Мы создали два дополнительных слота — open() и save(). Они будут открывать и сохранять документ. Пример их реализации будет представлен позже.

Очень часто, один и тот же слот используется одновременно несколькими виджетами. Например, в пунктах меню и соответственных кнопках на панели инструментов. Чтобы упростить этот процесс, в Qt существует класс QAction, который можно передавать сразу нескольким виджетам и присоединять к слоту. Например, QMenu и QToolBar могут создавать пункты меню, используя одно действие QAction. Скоро вы убедитесь, как это облегчает процессе разработки.

Сначала, мы используем конструктор класса Notepad для создания графического интерфейса программы.

```

Notepad::Notepad()
{
    openAction = new QAction(tr( "&Open" ), this );
    saveAction = new QAction(tr( "&Save" ), this );
    exitAction = new QAction(tr( "E&xit" ), this );

    connect(openAction, SIGNAL(triggered()), this, SLOT(open()));
    connect(saveAction, SIGNAL(triggered()), this, SLOT(save()));
    connect(exitAction, SIGNAL(triggered()), qApp, SLOT(quit()));

    fileMenu = menuBar()->addMenu(tr( "&File" ));
    fileMenu->addAction(openAction);
}

```

```
fileMenu->addAction(saveAction);
```

```
fileMenu->addSeparator();
```

```
fileMenu->addAction(exitAction);
```

```
textEdit = new QTextEdit;  
setCentralWidget(textEdit);
```

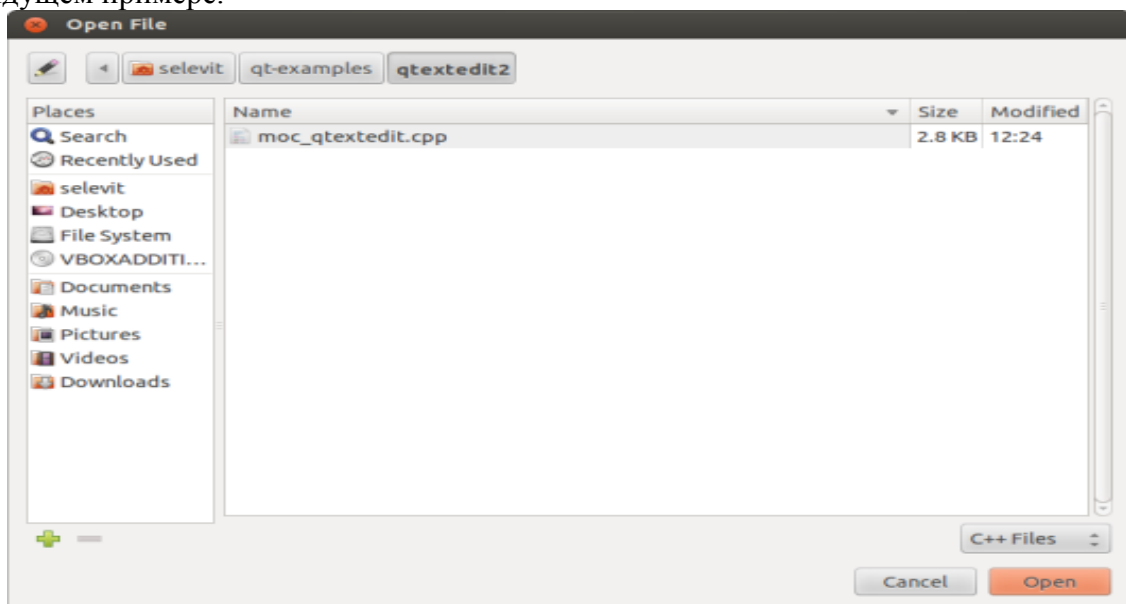
```
setWindowTitle(tr( "Notepad" ));
```

```
}
```

При создании действий, задается текст, отображаемый внутри виджетов, в которые будут добавлены эти действия (в нашем случае, пункты меню). Если нам нужно добавить кнопки для этих действий в панель инструментов, мы можем передать иконку для каждого действия. Когда пользователь кликает на пункте меню, рассылается сигнал действия и вызывается соответствующий слот.

Сохранение и открытие

В этом примере, мы реализуем функционал слотов `open()` и `save()`, которые были добавлены в предыдущем примере.



```
void Notepad::open()
```

```
{
```

```
QString fileName = QFileDialog::getOpenFileName( this, tr( "Open File" ), "",
```

```
tr( "Text Files (*.txt);;C++ Files (*.cpp *.h)" ));
```



```

    if (fileName != "" ) {
        QFile file(fileName);
        if (!file.open(QIODevice::ReadOnly)) {
            QMessageBox::critical( this , tr( "Error" ), tr( "Could not open file" ));
        }
        return ;
    }
    QTextStream in(&file);
    textEdit->setText(in.readAll());
    file.close();
}
}

```

Первый шаг — это запрос у пользователя имени файла для открытия. В состав Qt входит класс `QFileDialog` — диалоговое окно, в котором пользователь может выбрать файл. Статическая функция `getOpenFileName()` открывает модальное окно для выбора файла и возвращает путь к выбранному файлу. Если пользователь отменил действие, функция возвращает пустую строку.

После того, как было получено имя файла, мы пробуем открыть его с помощью функции `open()`. Она возвращает логическое `true` в случае успешного завершения. Если файл открыть не удалось, мы используем класс `QMessageBox` для того, чтобы показать пользователю всплывающее окно с сообщением об ошибке (смотрите описание класса `QMessageBox`, для получения более подробной информации).

Чтение данных становится тривиальной задачей, благодаря классу `QTextStream` — оберткой над объектом `QFile`. Функция `readAll()` возвращает содержимое файла, как объект `QString`. Это содержимое мы поместим в поле для редактирования текста. Затем, мы закрываем файл с помощью функции `close()`, чтобы вернуть файловый дескриптор операционной системе. Перейдем к реализации слота `save()`.

```

void Notepad::save()
{
    QString fileName = QFileDialog::getSaveFileName( this , tr( "Save File" ), "",
        tr( "Text Files (*.txt);;C++ Files (*.cpp *.h)" ));
    if (fileName != "" ) {
        QFile file(fileName);
        if (!file.open(QIODevice::WriteOnly)) {
            // error message
        } else {
            QTextStream stream(&file);

```

```
stream << textEdit->toPlainText();
```

```
stream.flush();
```

```
file.close();
```

```
}
```

```
}
```

```
}
```

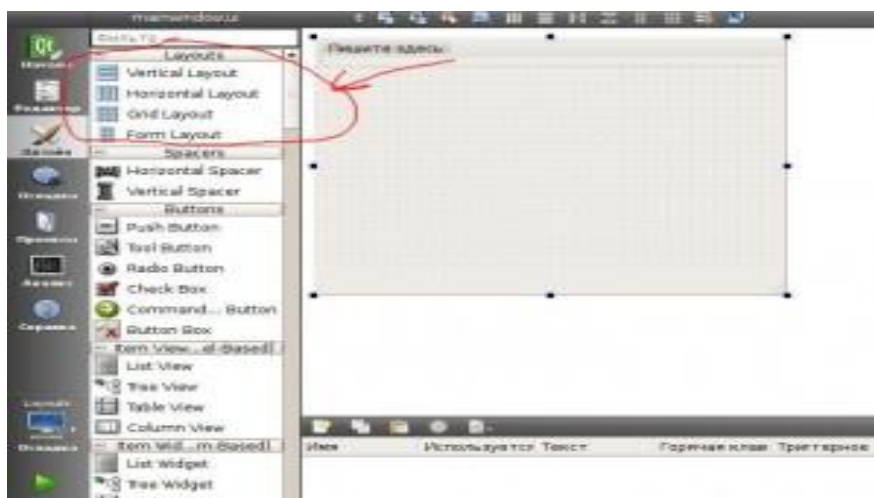
Для сохранения данных мы снова используем класс QTextStream. При помощи него, можно записывать строки QString в файл, используя оператор <<.

### Практическая работа № 11

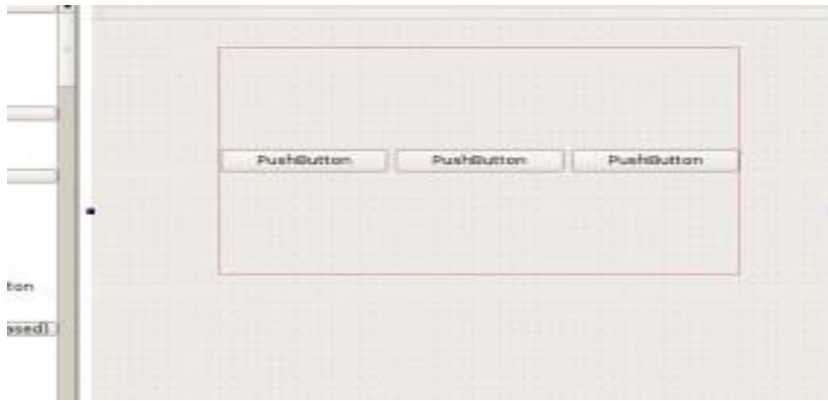
**Тема:** использование Qt layout для оптимизации интерфейса программы.

**Цель:** разместить компонент на **Qt layout**

Средства **Qt** позволяют автоматически **оптимизировать интерфейс программы** для любого разрешения монитора или изменения размера окна. Сделать это можно с помощью **Qt layout**. При добавлении компонентов обычным методом (перетягиванием компонента в **дизайнере форм**) он имеет фиксированный размер. Это может вызвать нежелательные артефакты при изменении размеров окна. Если же разместить компонент на **Qt layout**, размер компонента будет автоматически подстраиваться под окно пользователя. **Qt layout** бывает нескольких видов, в зависимости от нужд пользователя и желания феншуйности интерфейса.

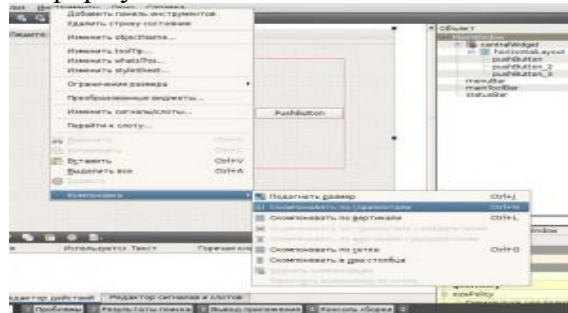


Чтобы начать использовать **Qt layout**, нужно в **Form Designer** добавить необходимый layout (см. рис. выше), потом разместить туда необходимые компоненты.



Можно разместить несколько **Qt layout** на форме в зависимости от нужд.

Дальше, для автоматического изменения размера **Qt layout** 'ов при изменении размера окна необходимо скомпонировать форму:



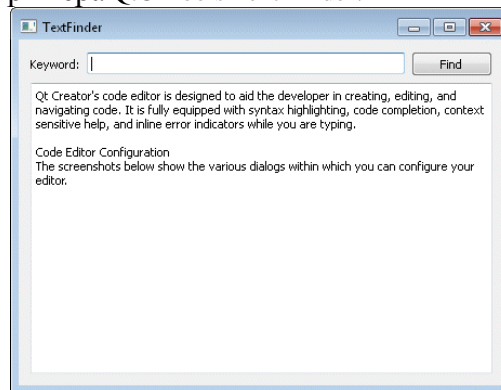
Можно неограниченно вкладывать **Qt layout**'ы друг в друга. Если необходимо сохранить размер отдельно компонента, можно разместить его внутри отдельного **Qt Widget**, вложенного в **Qt layout**.

## Практическая работа № 12

**Тема:** Создание приложения Qt на C++

**Цель:** написание программы Text Finder с использованием Qt Creator.

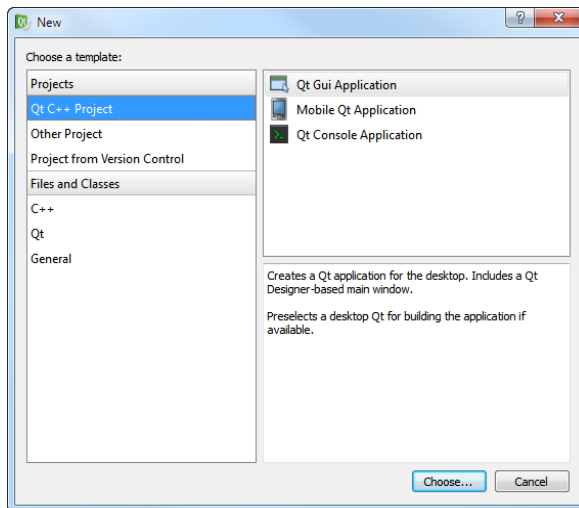
Это упрощённая версия примера QtUiTools Text Finder.



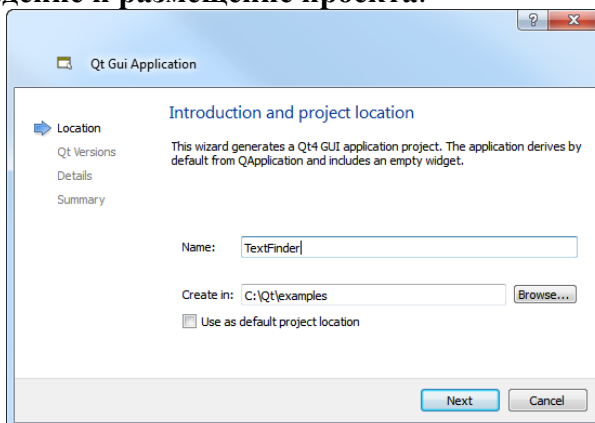
### **Создание проекта Text Finder**

**Замечание:** Создайте проект с активным режимом **Справка** чтобы вы могли следовать этим инструкциям во время работы.

1. Выберите **Файл > Новый файл или проект... > Проект Qt C++ > GUI приложение Qt > Выбрать....**



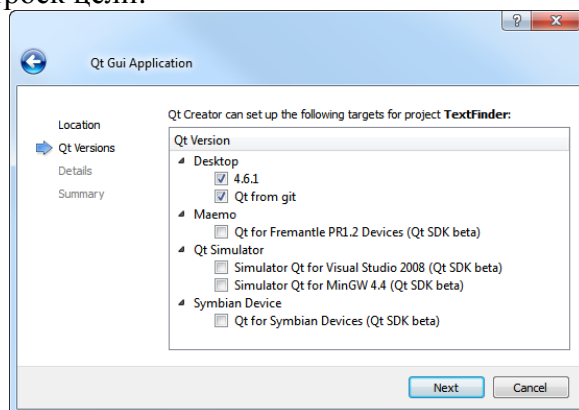
Откроется диалог **Введение и размещение проекта**.



2. В поле **Имя** введите **TextFinder**.

3. В поле **Создать в** введите путь к файлам проекта. Например, C:\Qt\examples, и нажмите **Вперёд**.

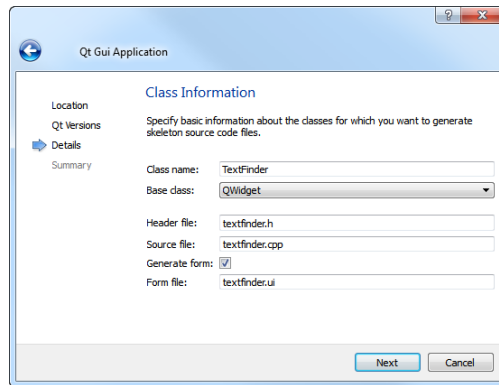
Откроется диалог настроек цели.



4. Выберите нужный профиль Qt для сборки вашего проекта и нажмите **Далее**.

**Замечание:** Если у вас установлен только один профиль Qt, этот диалог будет пропущен.

Откроется диалог **Информация о классе**.



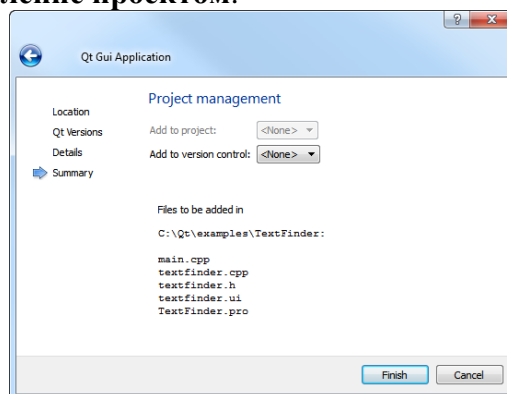
5. В поле **Имя класса** введите **TextFinder** в качестве имени класса.

6. В списке **Базовый класс** выберите **QWidget** в качестве типа базового класса.

**Замечание:** Поля **Заголовочный файл**, **Файл исходников** и **Файл формы** будут обновлены автоматически в соответствии с выбранным вами именем класса.

7. Нажмите **Вперёд**.

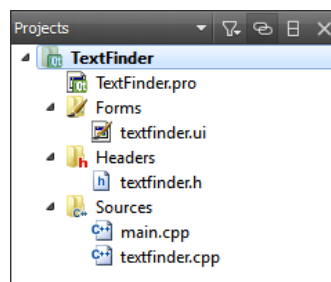
Откроется диалог **Управление проектом**.



8. Проверьте настройки проекта и нажмите **Завершить** для создания проекта.

Проект TextFinder будет содержать следующие файлы:

- textfinder.h
- textfinder.cpp
- main.cpp
- textfinder.ui
- textfinder.pro

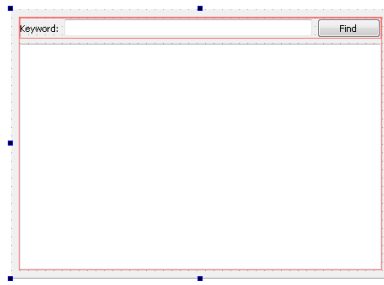


Файлы .h и .cpp содержат необходимые строки кода. Файл .pro полностью завершён.

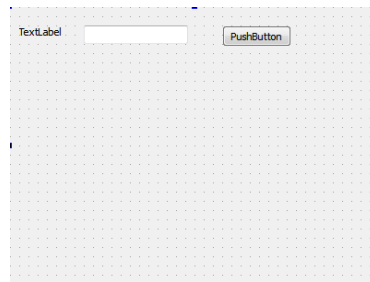
### ***Заполнение недостающих кусков***

Начнём с проектирования интерфейса пользователя и затем перейдём к заполнению недостающего кода. В заключение добавим поиск.

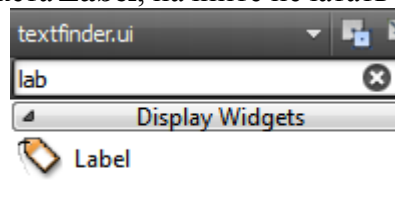
## **Проектирование пользовательского интерфейса**



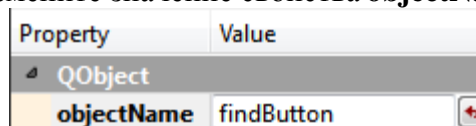
1. В режиме **Редактор** дважды нажмите на файле `textfinder.ui` в виде **Проекты** для запуска интегрированного *Qt Designer*.
2. Перетащите следующие виджеты на форму:
  1. **Label** (QLabel)
  2. **Line Edit** (QLineEdit)
  3. **Push Button** (QPushButton)



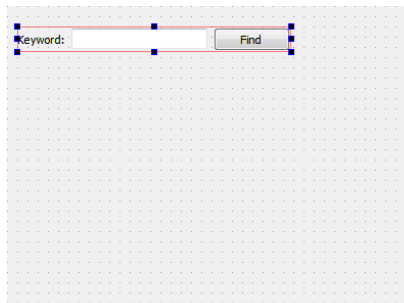
**Замечание:** Для быстрого поиска виджетов, используйте строку поиска в **боковой панели**. Например, для поиска виджета **Label**, начните печатать слово **label**.



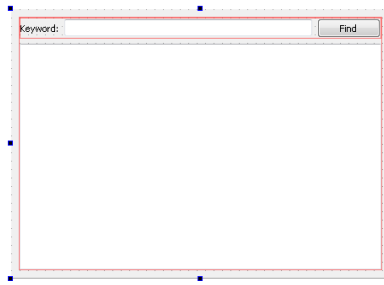
- Дважды нажмите на виджет **Label** и введите текст **Keyword**.  
 Дважды нажмите на виджет **Push Button** и введите текст **Find**.  
 В панели **Свойства** измените значение свойства **objectName** на **findButton**.



Нажмите **Ctrl+A** для выбора виджетов и нажмите **Скомпоновать по горизонтали** (или нажмите **Ctrl+H**) для применения горизонтальной компоновки (QHBoxLayout).



- Перетащите виджет **Text Edit** (QTextEdit) на форму.  
 Выберите область экрана и нажмите **Скомпоновать по вертикали** (или нажмите **Ctrl+L**) для применения вертикальной компоновки (QVBoxLayout).



Применение вертикальной и горизонтальной компонок обеспечивает масштабирование интерфейса приложения при различных разрешениях экрана.

Для вызова функции поиска при нажатии пользователем кнопки **Find** вы будете использовать механизм сигналов и слотов Qt. Сигнал вырабатывается когда происходит определённое событие, а слот - это функция которая вызывается в ответ на определённый сигнал. У виджетов Qt есть predefined сигналы и слоты которые вы можете использовать прямо из *Qt Designer*. Чтобы добавить слот для функции поиска:

0. Нажмите правой кнопкой мыши на кнопке **Find** для открытия контекстного меню.
1. Выберите **Перейти к слоту... > clicked()**, и нажмите **ОК**.

Закрывающийся слот `on_findButton_clicked()` будет добавлен в заголовочный файл `textfinder.h` и закрытая функция `TextFinder::on_findButton_clicked()` будет добавлена в файл исходных кодов `textfinder.cpp`.

Нажмите **Ctrl+S** для сохранения изменений.

Для получения дополнительной информации о проектировании форм с *Qt Designer* смотрите Руководство по Qt Designer.

#### Завершение заголовочного файла

Файл `textfinder.h` уже содержит необходимые директивы `#include`, конструктор, деструктор и объект `Ui`. Вам потребуется добавить закрытую функцию `loadTextFile()` для чтения и отображения содержимого входного файла в `QTextEdit`.

В боковой панели **Проекты** в режиме **Редактор** нажмите два раза на файле `textfinder.h` чтобы открыть его для редактирования.

Добавьте закрытую функцию в секцию `private` за `Ui::TextFinder`, как продемонстрировано в следующем фрагменте кода:

```
private slots:
    void on_findButton_clicked();
private:
    Ui::TextFinder *ui;
    void loadTextFile();
```

#### Завершение файла исходных кодов

Теперь заголовочный файл завершён, перейдём к файлу исходных кодов `textfinder.cpp`.

1. В боковой панели **Проекты** в режиме **Редактор** нажмите два раза на файле `textfinder.cpp` чтобы открыть его для редактирования.

2. Добавьте код для загрузки текстового файла с помощью `QFile`, чтения его с помощью `QTextStream`, а затем отображения его в `textEdit` с помощью `setPlainText()`. Это продемонстрировано в следующем фрагменте кода:

```
3. void TextFinder::loadTextFile()
4. {
5.     QFile inputFile(":/input.txt");
6.     inputFile.open(QIODevice::ReadOnly);
7.
8.     QTextStream in(&inputFile);
9.     QString line = in.readAll();
10.    inputFile.close();
```

```

11.
12.   ui->textEdit->setPlainText(line);
13.   QTextCursor cursor = ui->textEdit->textCursor();
14.   cursor.movePosition(QTextCursor::Start, QTextCursor::MoveAnchor, 1);
15.   }
15. Для использования QFile и QTextStream добавьте следующие директивы #includes в
textfinder.cpp:
16. #include <QtCore/QFile>
17. #include <QtCore/QTextStream>
17. В слоте on_findButton_clicked() добавьте код извлечения строки поиска и
использования функции find() для поиска строки в текстовом файле. Это продемонстрировано в
следующем фрагменте кода:
18. void TextFinder::on_findButton_clicked()
19. {
20.     QString searchString = ui->lineEdit->text();
21.     ui->textEdit->find(searchString, QTextDocument::FindWholeWords);
22. }
22. После того как эти функции завершены, добавьте строку для вызова loadTextFile() в
конструкторе, как продемонстрировано в следующем фрагменте кода:
23. TextFinder::TextFinder(QWidget *parent)
24.     : QWidget(parent), ui(new Ui::TextFinder)
25. {
26.     ui->setupUi(this);
27.     loadTextFile();
28. }

```

Слот on\_findButton\_clicked() будет вызван автоматически в сгенерированном uiс ui\_textfinder.h этой строчкой кода:

```
QMetaObject::connectSlotsByName(TextFinder);
```

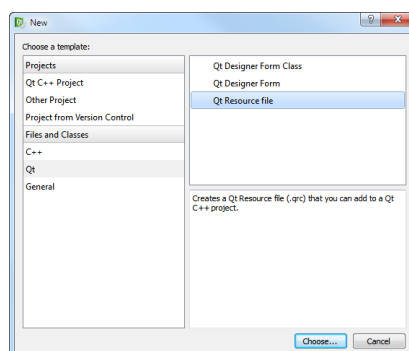
### Создание файла ресурсов

Вам потребуется файл ресурсов (.qrc) в котором вы сохраните входной текстовый файл. Входным файлом может быть любой файл .txt с абзацем текста. Создайте текстовый файл с именем input.txt и сохраните его в каталоге textfinder.

Чтобы добавить файл ресурсов:

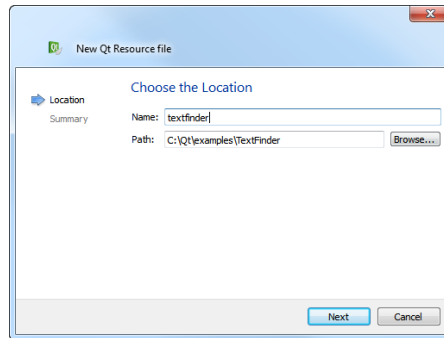
1. Выберите **Файл > Новый файл или проект > Qt > Файл ресурсов Qt >**

**Выбрать....**

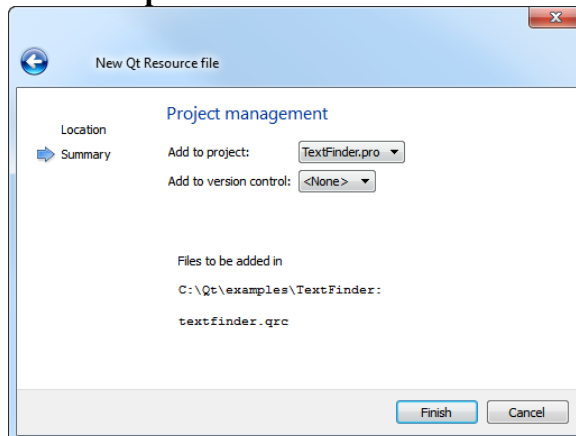


Откроется диалог **Выбор размещения**.

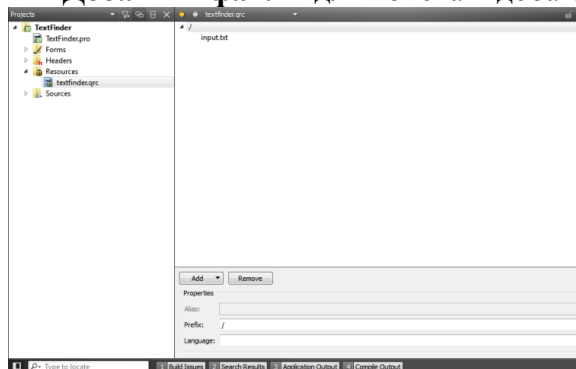





2. В поле **Имя** введите **textfinder**.
  3. В поле **Путь** введите **C:\Qt\examples\TextFinder** и нажмите **Далее**.
- Откроется диалог **Управление проектом**.



4. В поле **Добавить в проект** выберите **TextFinder.pro** и нажмите **Завершить** для открытия файла в редакторе кода.
5. Выберите **Добавить > Добавить префикс**.
6. В поле **Префикс** замените префикс по умолчанию на **(/)**.
7. Выберите **Добавить > Добавить файлы** для поиска и добавления **input.txt**.



### ***Сборка и запуск вашего приложения***

Теперь, когда у вас есть все необходимые файлы, нажмите на кнопку  чтобы скомпилировать и запустить вашу программу.

## **Практическая работа № 13**

**Тема:** Механизм сигналов и слотов, позволяющий объектам взаимодействовать друг с другом.

**Цель:** Разработка программы в Qt Creator с графическим интерфейсом, использующей сигналы и слоты.

Важные классы Qt, не предназначенные для работы с графическим интерфейсом; надеюсь, что вы самостоятельно разобрались с классами, которые не обсуждались, поскольку секрет успешного изучения любой технологии заключается в самостоятельной работе. В этой статье

мы поработаем с механизмом, лежащим в основе поддержки графических интерфейсов в Qt - сигналами и слотами.

Механизм сигналов и слотов позволяет объектам взаимодействовать друг с другом. В повседневной жизни мы можем найти множество примеров подобного механизма; например, светофор - когда загорается красный свет, вы останавливаете свое транспортное средство. Красный свет является сигналом, а остановка транспортного средства - слотом. Когда светофор генерирует сигнал включения красного света, вы вызываете слот, т.е. вы останавливаете транспортное средство.

Для правильного понимания этого механизма нам необходимо попрактиковаться в работе с Qt - разработать нашу первую программу с графическим интерфейсом, использующую сигналы и слоты. Откройте Qt Creator. Перейдите в меню Файл -> Новый файл или проект -> Проект Qt Widget -> GUI приложение Qt (File -> New File or Project -> Qt Widget Project -> Qt GUI Application) (Рисунок 1). Выберите директорию для размещения и название вашего проекта (Рисунок 2). В качестве цели выберите Desktop (Рисунок 3). (картинки кликабельны)



Рис.1: Qt Creator



Рис.2: Выбор директории размещения



Рис.3: Выбор цели проекта

После этого нужно будет ответить на вопросы о классах. В поле выбора базового класса вы можете обнаружить три варианта, используемые для разных целей. Мы рассмотрим их в следующей статье. На данный момент следует выбрать QDialog в качестве базового класса (Рисунок 4). Дайте имя вашему классу, после чего нажмите "Далее", и, наконец, нажмите "Завершить".

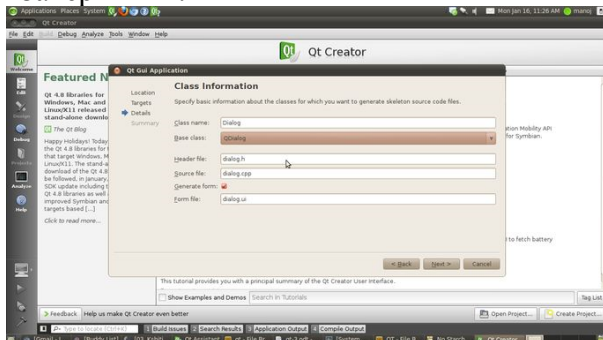


Рисунок 4: Создание класса

Сразу после этого Qt Creator откроет ваш новый проект (Рисунок 5) в окне проекта слева. В составе проекта должен присутствовать один файл проекта с названием project-name.pro, а в нашем случае 1.pro.

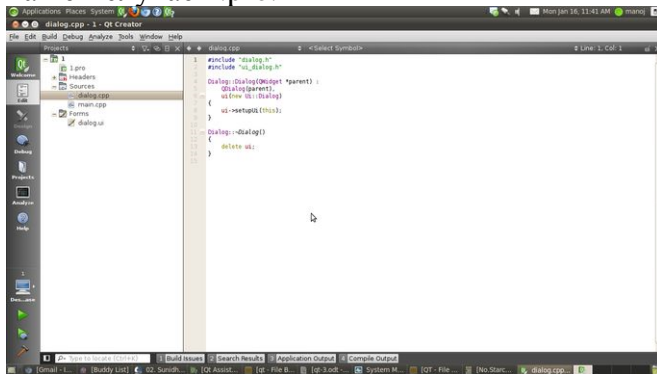


Рисунок 5: Редактор кода

Ниже расположены три директории: Заголовочные (Headers), Исходники (Sources) и Формы (Forms), в которых находятся файлы соответствующих типов. Откройте любой файл при помощи двойного клика. Класс, которому мы дали название в процессе создания проекта, состоит из трех файлов: dialog.h (заголовочный файл), dialog.cpp (исходный код C++)

и dialog.ui (файл пользовательского интерфейса или формы). Откройте файл пользовательского интерфейса в редакторе с помощью двойного клика (Рисунок 6).

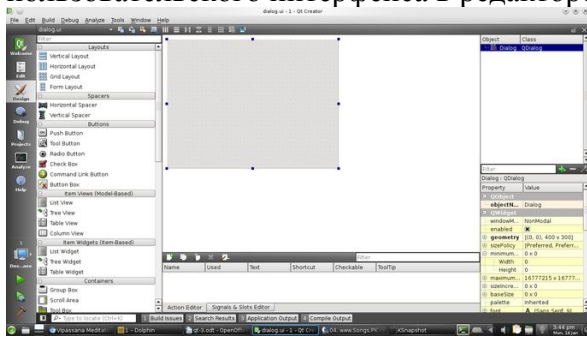
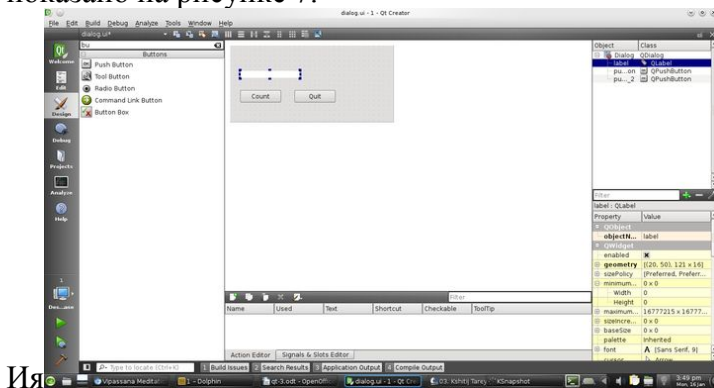


Рисунок 6: Редактор пользовательского интерфейса

Панель слева содержит список элементов графического интерфейса, которые могут быть помещены в окне диалога при помощи простого перетаскивания. Используем в нашем первом примере строку (Label) и две кнопки (Push Button). Окно диалога должно выглядеть так, как показано на рисунке 7.



Имя

Рисунок 7: Окно диалога

Для простоты не будем использовать элементы размещения (Layouts). При помощи двойных кликов измените текст кнопок на "Count" и "Quit"; используйте двойной клик по отношению к строке и удалите текст. Нажатие на кнопку "Count" должно увеличивать значение счетчика и выводить его в качестве текста строки, а нажатие на кнопку "Quit" - завершать приложение.

Теперь мы добавим слот для сигнала "clicked" кнопки "Quit". Нажмите F4 для перехода в режим редактора сигналов и слотов. Нажмите кнопку "Quit" и отпустите кнопку мыши на свободном пространстве окна диалога. После того, как вы отпустите кнопку мыши, должно появиться окно "Настройка соединения" (Configure Connection) (Рисунок 8). Слева перечислены сигналы, которые может генерировать элемент графического интерфейса; справа перечислены слоты (проще говоря, действия), которые могут быть выполнены с элементом графического интерфейса QDialog.

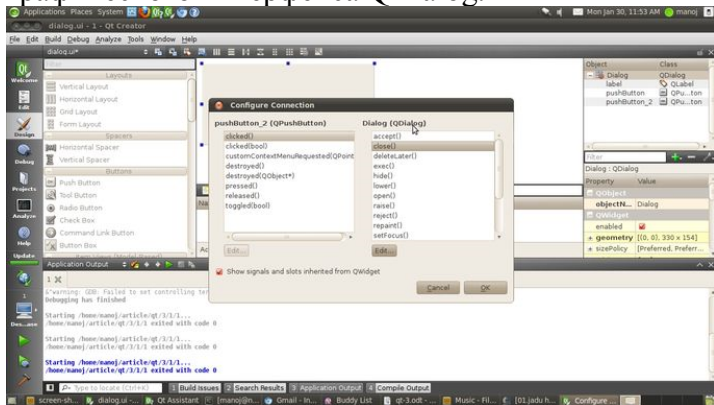


Рисунок 8: Настройка соединения

В левом списке выберите clicked(), а в правом - close(). Таким образом, при нажатии кнопки "Quit" окно будет закрыто. Нажмите F3 для перехода в режим редактирования свойств элементов графического интерфейса. Кнопка "Count" тоже должна быть соединена со слотом. Выберите эту кнопку, нажмите правой кнопкой мыши и во всплывающем меню выберите

пункт "Перейти к слоту..." ("Go to Slot"). В появившемся окне (Рисунок 9) "Переход к слоту" ("Go to Slot") будут перечислены сигналы, генерируемые данным элементом графического интерфейса.

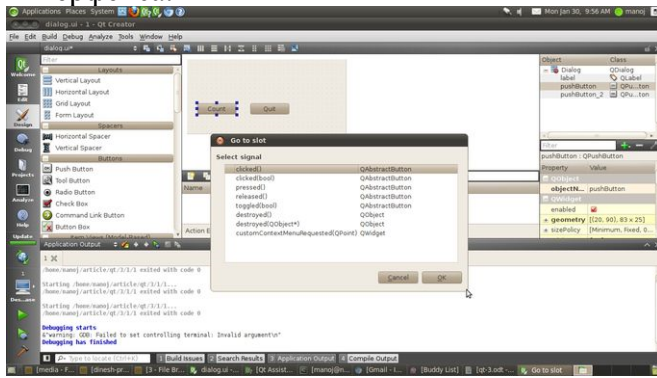


Рисунок 9: Переход к слоту

Выберите сигнал clicked() и нажмите ОК. В редакторе кода будет открыта функция void Dialog::on\_pushButton\_clicked() из файла dialog.cpp (Рисунок 10).

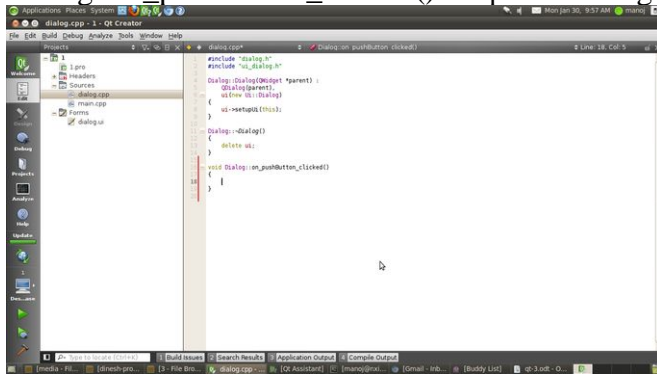


Рисунок 10: Функция слота

Отредактируйте этот код следующим образом:

```
void Dialog::on_pushButton_clicked()
{
    static int count;
    count++;
    ui->label->setText(QString::number(count));
}
```

В этом коде используется статическая переменная, значение которой увеличивается на 1 при каждом вызове функции, после чего значение преобразуется в строку QString и устанавливается в качестве текста строки. Нажмите Ctrl+R для сборки и запуска приложения (как на Рисунке 11).

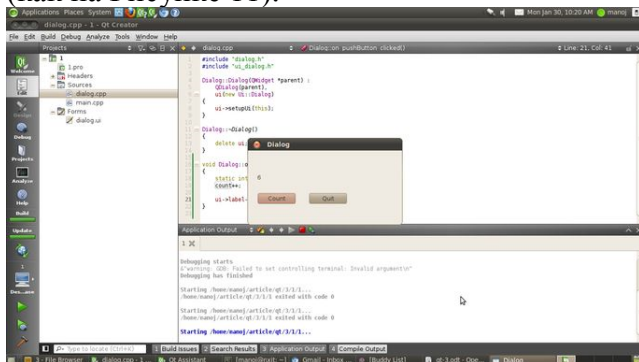


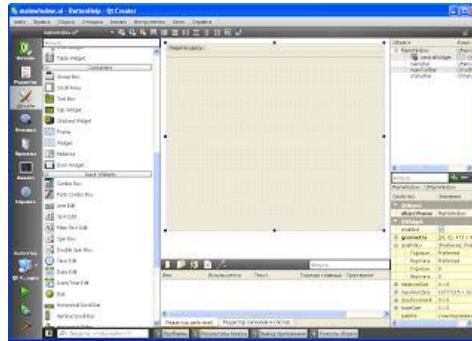
Рисунок 11: Вывод

Протестируйте приложение - нажмите кнопку "Count" и проверьте результат. Нажмите кнопку "Quit" для завершения работы приложения.

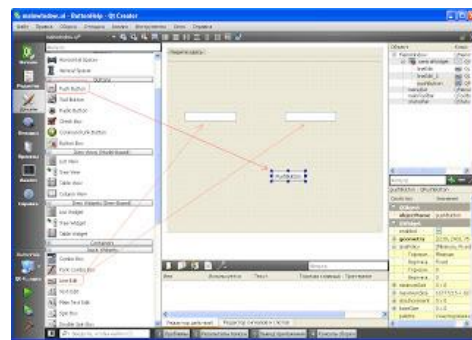
## Практическая работа № 14

Тема: Элементы управления

**Цель:** Разработать приложение с элементами управления QPushButton и QLineEdit  
Открываем файл mainwindow.ui двойным кликом и нашим глазам предстанет следующая картина:



Следующим шагом добавим два QLineEdit и один QPushButton, т.е. 2 поля для ввода текста и одно кнопку. Будем имя кнопки звать слиянием текстовых значений в текстовых полях.



Для добавленных объектов существуют имена по умолчанию (поле objectName справа во второй половине списка свойств), в нашем случае это lineEdit и lineEdit\_2 для текстовых полей и pushButton для кнопки.

В автоматически сгенерированных файлах к классу MainWindow уже есть указатель на форму: Ui::MainWindow \*ui;

В конструкторе уже прописана его инициализация:

```
MainWindow::MainWindow(QWidget *parent) :  
    QMainWindow(parent),  
    ui(new Ui::MainWindow)  
{  
    ui->setupUi(this); // инициализация указателя на форму  
}
```

В соответствии с особенностью Qt в виде сигналов и слотов ( [http://easy-qt.blogspot.com/2012/01/qt\\_31.html](http://easy-qt.blogspot.com/2012/01/qt_31.html) ) наша кнопка будет "сигналить" при клике по ней (да поможет вам справка). Для этого создадим слот, к которому в последующем и подключим наш сигнал. mainwindow.h будет выглядеть следующим образом:

```
#ifndef MAINWINDOW_H  
  
#define MAINWINDOW_H  
  
#include <QMainWindow>  
  
#include <QString> // добавим эту библиотеку для работы со строками
```

```

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

private slots:
    void nashSlot(); // это созданный нами слот

private:
    Ui::MainWindow *ui;
};

#endif // MAINWINDOW_H

```

Теперь делаем определение этого слота в mainwindow.cpp

```

void MainWindow::nashSlot()
{
    QString str1 = ui->lineEdit->text(); // получаем строку из первого QLineEdit
    QString str2 = ui->lineEdit_2->text(); // получаем строку из второго QLineEdit
    QString finalStr = str1 + str2; // объединяем эти строки в одну
    ui->pushButton->setText(finalStr); // задаем имя кнопки равным финальной строке
}

```

После этого у вас ещё ничего не заработает, не торопитесь. Остался последний штрих - подключить сигнал от кнопки к нашему слоту. Сделаем это в конструкторе следующим образом:

```

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    connect(ui->pushButton, SIGNAL(clicked()), this, SLOT(nashSlot()));
}

```

Это общее представление о том, как работать с кнопками и текстовыми полями. Если вам надо получить числовое значение из QLineEdit (а это бывает очень часто), делается это так:

```
int vall = ui->lineEdit->text().toInt();
```

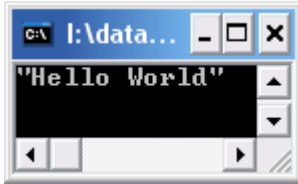
А обратно

```
QString vall2 = QString::number(vall);
```

### **Практическая работа № 15**

**Тема:** Окно с текстом

**Цель:** Разработать приложение с текстом в окне и сделать описание каждого оператора



Ниже приводится текст оконного приложения Qt программы:

```
#include <QtCore/QCoreApplication>
```

```
#include <QDebug>
```

```
int main(int argc, char *argv[])  
{  
    QCoreApplication a(argc, argv);  
  
    QString mStr = "Hello World";  
  
    qDebug () << mStr;  
  
    return a.exec();  
}
```

Ответить на вопросы:

Какие определения классов подключаются в данном приложении?

Что управляет ресурсами приложения?

Какие аргументы имеет конструктор QCoreApplication?

Какой визуальный компонент создаётся в приложении для отображения надписи "Hello World"?

Каким оператором выполняется передача управления библиотеке Qt. С этого момента программа переходит в режим ожидания, когда она ничего не делает, а просто ждет действий пользователя, например, нажатие на клавишу или кнопку мыши.

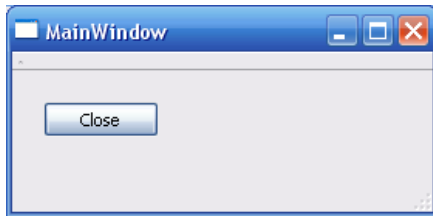
Зайти в папку 02 Hello World с исполняемым файлом данного приложения hello.exe и запустить его для выполнения.

### **Практическая работа № 16**

**Тема:** Окно с текстом на элементе управления «кнопка»

**Цель:** Разработать приложение с текстом в окне и сделать описание каждого оператора





Ниже приводится текст оконного приложения Qt программы:

Main.cpp

```
#include <QtGui/QApplication>
#include "mainwindow.h"
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}
```

mainwindow.cpp:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
```

```
MainWindow::MainWindow(QWidget *parent) :
```

```
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    ui->pushButton->setText("Close");
}
```

```
MainWindow::~MainWindow()
{
    delete ui;
}
```

Ответить на вопросы:

Какие определения классов подключаются в данном приложении?

Что управляет ресурсами приложения?

Какие аргументы имеет конструктор QApplication?

Какой визуальный компонент создаётся в приложении?

Каким оператором выполняется передача управления библиотеке Qt. С этого момента программа переходит в режим ожидания, когда она ничего не делает, а просто ждет действий пользователя, например, нажатие на клавишу или кнопку мыши.

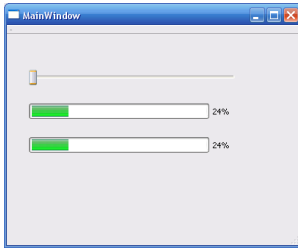
Зайти в папку 03 Intro to GUI с исполняемым файлом данного приложения gui.exe и запустить его для выполнения.

### **Практическая работа № 17**

**Тема:** Окно с элементами управления «кнопка» и «horizontalSlider»



**Цель:** Разработать приложение с элементами управления в окне и сделать описание каждого оператора.



Ниже приводится текст оконного приложения Qt программы:

Main.cpp

```
#include <QtGui/QApplication>
#include "mainwindow.h"
```

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}
```

mainwindow.cpp:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
```

```
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    connect(ui->horizontalSlider,SIGNAL(valueChanged(int)),
            ui->progressBar,SLOT(setValue(int)));

    connect(ui->horizontalSlider,SIGNAL(valueChanged(int)),
            ui->progressBar_2,SLOT(setValue(int)));
}
```

```
MainWindow::~MainWindow()
{
    delete ui;
}
```

```
void MainWindow::on_pushButton_clicked(bool checked)
{
}
}
```

```

void MainWindow::on_pushButton_released()
{

}

void MainWindow::on_pushButton_toggled(bool checked)
{

}

void MainWindow::on_pushButton_pressed()
{

}

```

```

mainwindow.h
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

namespace Ui {
    class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

private:
    Ui::MainWindow *ui;

private slots:
    void on_pushButton_pressed();
    void on_pushButton_toggled(bool checked);
    void on_pushButton_released();
    void on_pushButton_clicked(bool checked);
};

#endif // MAINWINDOW_H

```

### Ответить на вопросы:

Какие определения классов подключаются в данном приложении?

Что управляет ресурсами приложения?

Какие аргументы имеет конструктор QApplication?

Какие визуальные компоненты создаются в приложении?

Какие сигналы и слоты используются?

Какие события обрабатываются?

Каким оператором выполняется передача управления библиотеке Qt. С этого момента программа переходит в режим ожидания, когда она ничего не делает, а просто ждет действий пользователя, например, нажатие на клавишу или кнопку мыши.

### Задание:

Зайти в папку 04 Signals and Slots -> myprogress-build-desktop ->debug с исполняемым файлом данного приложения myprogress.exe и запустить его для выполнения.

Зайти в папку 04 Signals and Slots -> myprogress и запустить mainwindow.ui. Описать свойства каждого компонента.

## Практическая работа № 18

**Тема:** Графические возможности Qt

**Цель:** Разработать приложение с изображением на "графических устройствах"

**Двухмерная графика.**

Краеугольным камнем движка двухмерной графики в Qt является QPainter. Он может использоваться для рисования на поверхности виджета (на экране), во внутреннем буфере (pixmap) и на принтере. Кроме того, в состав Qt входит класс QCanvas, который позволяет создавать изображения из графических примитивов.

В качестве альтернативы QPainter и QCanvas, можно рассматривать библиотеку OpenGL. Она предоставляет механизмы создания трехмерной графики, но может использоваться и для рисования двухмерных изображений. Код, использующий OpenGL очень легко интегрируется в приложения Qt.

### Рисование средствами QPainter

Класс QPainter используется для создания изображений на "графических устройствах", таких как виджеты или карты пикселей (pixmap). Чаще всего он используется при создании нестандартных виджетов, для придания им уникального, ни на что не похожего, внешнего вида. Однако этот класс может использоваться и для вывода графики на принтер, более подробно мы коснемся этого вопроса немного ниже.

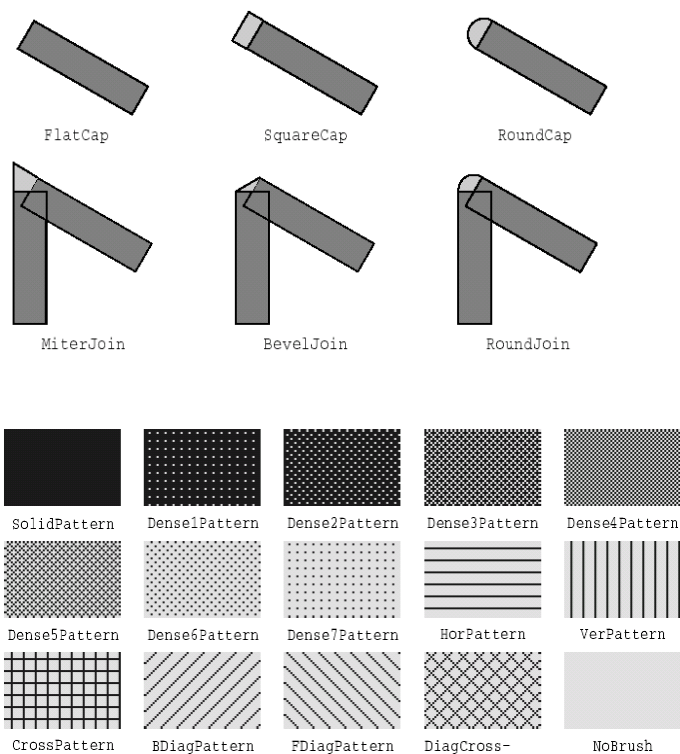
QPainter может рисовать простые геометрические фигуры: точки, линии, прямоугольники, эллипсы, дуги, сегменты круга, замкнутые ломаные (многоугольники) и кривые Безье. Он так же может отображать карты пикселей, рисунки и текст.

Тремя наиболее важными характеристиками QPainter являются **перо** (pen), **кисть** (brush) и **шрифт** (font).

- *Перо* используется для рисования линий и границ геометрических фигур. Оно характеризуется такими параметрами, как: цвет, толщина, стиль рисования линий, стиль оформления концов линий и стиль оформления углов.
- *Кисть* -- это шаблон, которым заполняются геометрические фигуры. Кисти характеризуются цветом и стилем.
- *Шрифт* используется для рисования текста. Шрифт может иметь огромное количество атрибутов, среди них: название и размер.

Настройки этих характеристик могут быть выполнены с помощью функций:

setPen(), setBrush() и setFont().



1. Итак, для того чтобы начать рисование, необходимо создать проект GUI приложение в Qt Creator, в качестве базового класса выбрать QDialog (вообще можно выбрать любой класс).
2. Войти в режим редактирования
3. Открыть заголовочный файл dialog.h и отредактировать код:

```
1. #ifndef DIALOG_H
2. #define DIALOG_H
3. #include <QDialog>
4. #include <QtGui>
5. #include <QtCore>
6. namespace Ui {
7.     class Dialog;
8. }
9. class Dialog : public QDialog
10. {
11.     Q_OBJECT
12. public:
13.     explicit Dialog(QWidget *parent = 0);
14.     ~Dialog();
15. private:
16.     Ui::Dialog *ui;
17. protected:
18.     void paintEvent(QPaintEvent *e);
19. };
20. #endif // DIALOG_H
```

3. Сохраните проект и откройте файл исходных кодов dialog.cpp. После кода:

```
Dialog::~Dialog()
{
    delete ui;
}
```

Допишите код:

```
void Dialog::paintEvent(QPaintEvent *e)
{
    //описание рисуемого объекта
}
```

Теперь можно начать описание рисуемого объекта.

**Примеры** описания рисуемых объектов:

**Объект « Линия»**

```
QPainter painter(this);
```

```
QPen pointpen(Qt::black);
pointpen.setWidth(6);
```

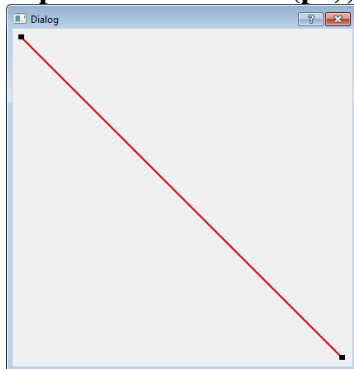
```
QPen linepen(Qt::red);
linepen.setWidth(2);
```

```
QPoint p1;
p1.setX(10);
p1.setY(10);
```

```
QPoint p2;
p2.setX(100);
p2.setY(100);
```

```
painter.setPen(linepen);
painter.drawLine(p1,p2);
```

```
painter.setPen(pointpen);
painter.drawPoint(p1);
painter.drawPoint(p2);
```



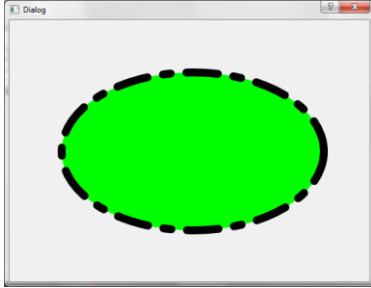
*Ниже приводится код, который рисует объект «эллипс».*

```
QPainter painter(this);
painter.setRenderHint(QPainter::Antialiasing, true);
```

```

painter.setPen(QPen(Qt::black, 12, Qt::DashDotLine, Qt::RoundCap));
painter.setBrush(QBrush(Qt::green, Qt::SolidPattern));
painter.drawEllipse(80, 80, 400, 240);

```

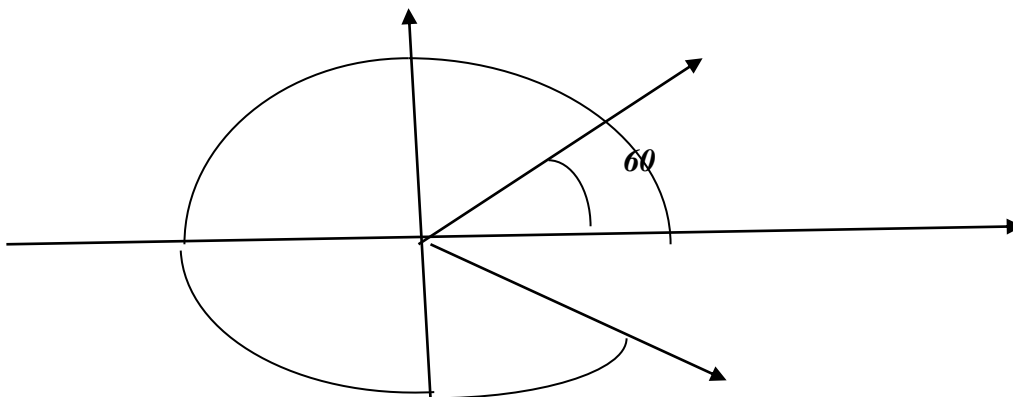
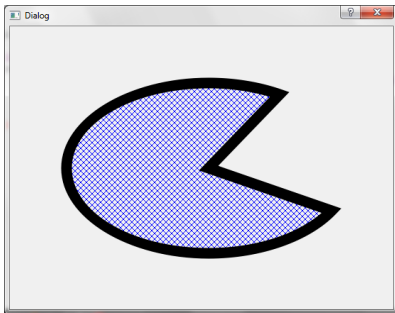


*Следующий код рисует объект «сегмент круга»*

```

QPainter painter(this);
painter.setRenderHint(QPainter::Antialiasing, true);
painter.setPen(QPen(Qt::black, 15, Qt::SolidLine, Qt::RoundCap, Qt::MiterJoin));
painter.setBrush(QBrush(Qt::blue, Qt::DiagCrossPattern));
painter.drawPie(80, 80, 400, 240, 60 * 16, 270 * 16);

```

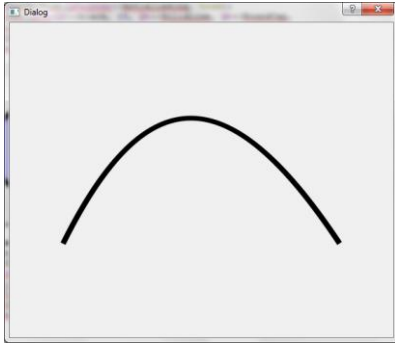


*Код, который рисует кривую Безье*

```

QPainter painter(this);
painter.setRenderHint(QPainter::Antialiasing, true);
QPainterPath path;
path.moveTo(80, 320);
path.cubicTo(200, 80, 320, 80, 480, 320);
painter.setPen(QPen(Qt::black, 8));
painter.drawPath(path);

```

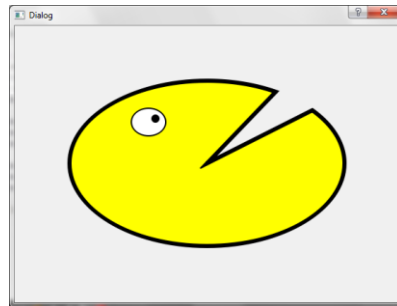


### Ещё пример

```

QPainter painter(this);
painter.setRenderHint(QPainter::Antialiasing, true);
painter.setPen(QPen(Qt::black, 6, Qt::SolidLine, Qt::RoundCap, Qt::MiterJoin));
painter.setBrush(QBrush(Qt::yellow));
painter.drawPie(80, 80, 400, 240, 60 * 16, 340 * 16);
painter.setPen(QPen(Qt::black, 2, Qt::SolidLine, Qt::RoundCap));
painter.setBrush(QBrush(Qt::white, Qt::SolidPattern));
painter.drawEllipse(170, 120, 50, 40);
painter.setPen(QPen(Qt::black, 2, Qt::SolidLine, Qt::RoundCap));
painter.setBrush(QBrush(Qt::black, Qt::SolidPattern));

```



```

painter.drawEllipse(200, 130, 10, 10)

```

### Градиенты

Для того чтобы нарисовать градиент допишите в заголовочном файле dialog.h следующие строки:

```

#include <QRadialGradient>
#include <QLinearGradient>
#include <QConicalGradient>

```

В файле исходного кода dialog.cpp в описании paintEvent вставить код описания рисуемого объекта «*Линейный градиент*»

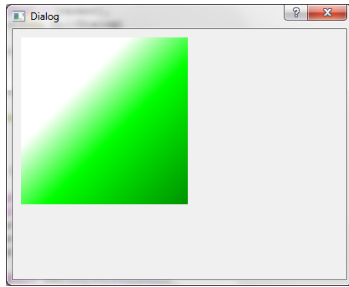
```

void Dialog::paintEvent(QPaintEvent *e)
{
//описание рисуемого объекта
}

// описание рисуемого объекта «Линейный градиент»
QPainter painter(this);
QLinearGradient gradient(50, 100, 300, 350); // создание градиента
gradient.setColorAt(0.0, Qt::white); // цвет градиента 1
gradient.setColorAt(0.2, Qt::green); // цвет градиента 2
gradient.setColorAt(1.0, Qt::black); // цвет градиента 3

```

```
QRect rec(10,10,200,200); // определение области заливки градиентом
painter.fillRect(rec, gradient); // заливка градиентом
```

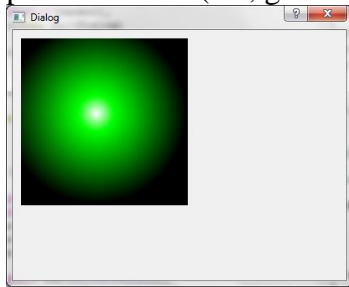


В файле исходного кода dialog.cpp в описании paintEvent вставить код описания рисуемого объекта *«Радиальный градиент»*

```
//описание рисуемого объекта «Радиальный градиент»
QRadialGradient gradient(100, 200, 130); // (центр, радиус, фокус)
```

```
gradient.setColorAt(0.0, Qt::white); // цвет градиента 1
gradient.setColorAt(0.2, Qt::green); // цвет градиента 2
gradient.setColorAt(1.0, Qt::black); // цвет градиента 3
```

```
QRect rec(10,10,200,200); // определение области заливки градиентом
painter.fillRect(rec, gradient); // заливка градиентом
```

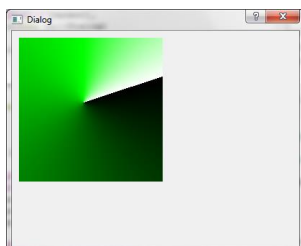


В файле исходного кода dialog.cpp в описании paintEvent вставить код описания рисуемого объекта *«Конический градиент»*

```
// описание рисуемого объекта «Конический градиент»
```

```
QPainter painter(this);
QConicalGradient gradient (100, 100, 18); //(центр1, центр2, угол)
gradient.setColorAt(0.0, Qt::white); // цвет градиента 1
gradient.setColorAt(0.2, Qt::green); // цвет градиента 2
gradient.setColorAt(1.0, Qt::black); // цвет градиента 3
```

```
QRect rec(10,10,200,200); // определение области заливки градиентом
painter.fillRect(rec, gradient); // заливка градиентом
```





## Практическая работа № 19

**Тема:** Разработка приложения с использованием элементов управления.

**Цель:** Разработать программный продукт на Qt, позволяющий решить квадратное уравнение.

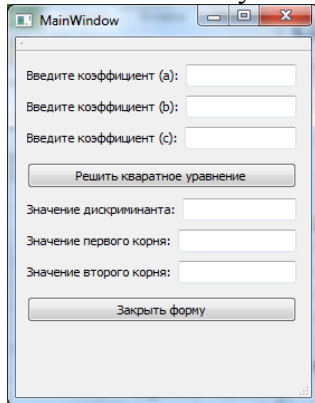
### **Задание.**

Для заданных коэффициентов  $a$ ,  $b$ ,  $c$  найти корни уравнения вида:  $ax^2+bx+c=0$ .

Создать проект в комплексной среде Qt, позволяющий решить квадратное уравнение.

Графический интерфейс представлен в виде окна с элементами управления для ввода коэффициентов и вывода корней уравнения. Кнопки используются для начала решения уравнения и выхода их приложения.

Описать используемые классы и обрабатываемые события.



### **Main.cpp**

```
#include <QtGui/QApplication>
#include "mainwindow.h"
```

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}
```

### **mainwindow.cpp**

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <math.h>
#include <sstream>
```

```
using std::string;
using std::stringstream;
```

```
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}
```

```
MainWindow::~MainWindow()
```

```

{
    delete ui;
}

void MainWindow::changeEvent(QEvent *e)
{
    QMainWindow::changeEvent(e);
    switch (e->type()) {
    case QEvent::LanguageChange:
        ui->retranslateUi(this);
        break;
    default:
        break;
    }
}

void MainWindow::on_pushButton_clicked()
{
    stringstream ss;
    float x1;
    float x2;
    float deskr;

    QString sx1;
    QString sx2;
    QString sdeskr;

    int a = ui->lineEdit->text().toInt();
    int b = ui->lineEdit_2->text().toInt();
    int c = ui->lineEdit_3->text().toInt();

    if(a == 0 && b == 0 && c == 0 && a != 0 && b != 0 && c != 0)
    {
        ui->label_7->setText("yPABHEHuE He KBADPATHOE");
    }

    deskr = (b * b) - (4 * a * c);
    sdeskr = QString::number(deskr);
    ui->lineEdit_4->setText(sdeskr);

    if(deskr < 0)
    {
        ui->label_7->setText("DECKPuMuHAHT MEHbLLIE HyJI9I");
    }

    if(deskr > 0)
    {
        x1 = ((b * (-1)) + sqrt(deskr)) / (2 * a);
        x2 = ((b * (-1)) - sqrt(deskr)) / (2 * a);
        sx1 = QString::number(x1);
        sx2 = QString::number(x2);
        ui->lineEdit_5->setText(sx1);
        ui->lineEdit_6->setText(sx2);
    }
}

```

```

if(deskr == 0)
{
    x1 = (b * (-1)) / (2 * a);
    sx1 = QString::number(x1);
    ui->lineEdit_4->setText(sx1);
}
}

void MainWindow::on_pushButton_2_clicked()
{
    close();
}

```

**mainwindow.h**

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

namespace Ui {
    class MainWindow;
}

class MainWindow : public QMainWindow {
    Q_OBJECT
public:
    MainWindow(QWidget *parent = 0);
    ~MainWindow();

protected:
    void changeEvent(QEvent *e);

private:
    Ui::MainWindow *ui;

private slots:
    void on_pushButton_2_clicked();
    void on_pushButton_clicked();
};

#endif // MAINWINDOW_H

```

## **Практическая работа № 20**

**Тема:** Методы и средства разработки технической документации программного продукта.

**Цель:** Разработать техническую документацию для программного продукта Qt.

### **Задание.**

Для готового программного модуля, разработанного в Qt, создать документ, содержащий техническую документацию.

### **Методические указания:**

Документация должна содержать необходимые сведения по установке, обеспечению надёжной работы продукта, справочное пособие для пользователя, демонстрационные версии, примеры документов, создаваемых при помощи данного программного продукта, обучающие программы.

Программная документация, включает:

- техническое задание (назначение, область применения программы, требования, предъявляемые к программе);
- текст программы (запись программы с необходимыми комментариями);
- описание программы (сведения о логической структуре и функционировании программы);
- пояснительная записка (схема алгоритма, общее описание алгоритма и/или функционирования программы, обоснование принятых решений);
- эксплуатационные документы.

Программный документ "Пояснительная записка" составляется на стадии эскизного или технического проектов программы. Как правило, на стадии рабочего проекта не используется.

К эксплуатационным документам относят:

- описание применения (сведения о назначении программы, области применения, применяемых методах, классе решаемых задач, ограничениях для применения, минимальной конфигурации технических средств);
- руководство системного программиста (сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения);
- руководство программиста (сведения для эксплуатации программы);
- руководство оператора (сведения для обеспечения общения оператора с вычислительной системой в процессе выполнения программы);
- описание языка (описание синтаксиса и семантики языка);
- руководство по техническому обслуживанию (сведения для применения тестовых и диагностических программ при обслуживании технических средств)

Основная часть программной документации составляется на стадии рабочего проекта. Необходимость того или иного документа определяется на этапе составления технического задания. Допускается объединять отдельные виды документов.

Эксплуатационный документ "Описание языка" включается в программную документацию, если разработанный программный продукт реализует некий язык программирования, управления заданиями, организации вычислительного процесса и т. п.

Эксплуатационный документ "Руководство по техническому обслуживанию" включается в программную документацию, если разработанный программный продукт требует использования тестовых или диагностических программ.

В техническое задание включают:

- введение (наименование, краткая характеристика области применения программы);
- основания для разработки (документы, на основании которых ведётся разработка, организация, утвердившая документы, дата утверждения, наименование и обозначение темы разработки);

- назначение разработки (функциональное и эксплуатационное назначение программы);
- требования к программе и программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приёмки.

Наиболее существенной частью технического задания является раздел "требования..." В этом разделе приводятся:

- требования к функциональным характеристикам (состав выполняемых функций, организация входных и выходных данных, временные характеристики);
- требования к надёжности (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа);
- требования к информационной и программной совместимости (требования к информационным структурам на входе и выходе, методам решения, исходным кодам, языкам программирования и программным средствам; требования к защите информации);
- требования к составу и параметрам технических средств;
- требования к программной документации.

Данный раздел может содержать требования к маркировке, упаковке, транспортировке и хранению, а также условия эксплуатации.

Кроме явно описанных в техническом задании требований, следует придерживаться общепринятых правил разработки программ с учётом выбранной парадигмы программирования:

1. Программа не должна содержать избыточные элементы (все элементы программы адекватны поставленной задаче: нет циклов, массивов и т. п. элементов, без которых можно обойтись).
2. Алгоритм должен быть структурирован: для функционального стиля программирования - адекватное разбиение на функции (процедуры), для объектно-ориентированного - адекватная иерархия классов. Каждая функция (метод класса) должна реализовывать ровно одно действие.
3. У функций (методов классов) должны быть параметры. Следует избегать использования в функциях глобальных переменных.
4. Программа должна аккуратно использовать память: работать с динамическими массивами, в ней не должно быть неиспользуемых блоков памяти, лишних переменных.
5. Должны проверяться диапазоны вводимых пользователем значений и параметров, передаваемых между модулями программы.
6. При использовании в программе каких-либо готовых компонент (библиотечных функций, классов) если функция или метод класса может завершиться неудачей, необходимо обязательно проверять это, не полагаясь на незначительность вероятности такого события.
7. Программа должна быть конфигурируема (важные параметры программы следует выделить в единый блок).

### **Текст программы**

Текст программы представляет собой символическую запись на исходном или промежуточном языке или символическое представление машинных кодов. Текст программы оформляется моноширинным шрифтом (Courier, Lucida Console и т. п.):

1. Количество операторов на строке должно быть равно 1.
2. Все операторы, входящие в составной оператор, должны быть сдвинуты вправо на одинаковое количество позиций, при этом операторные скобки (т. е. то, что ограничивает составной оператор), относящиеся к одному блоку, должны располагаться следующим образом: открывающая скобка должна находиться на той же строке, что и оператор, открывающий блок,

а закрывающая должна находиться в той же колонке, с которой начинается оператор, открывающий блок. Допускается располагать открывающую скобку на строке, следующей за оператором, открывающим блок, в той же колонке, с которой начинается этот оператор.

3. Строка исходного текста программы должна целиком располагаться в одной типографской строке (до 80 символов в зависимости от шрифта). Несоблюдение этого правила говорит о слишком большой вложенности блоков, что означает неудачный алгоритм или структуру программы. В таком случае рекомендуется переосмыслить структуру программы, ввести дополнительные функции, заменив какие-то большие части кода их вызовами, переделать алгоритм и т.п.
4. Если синтаксис языка позволяет, желательно отделять знаки операций пробелами от операндов. Как и в обычном тексте, после запятой должен следовать пробел.
5. Определения функций или логические части программы следует отделять друг от друга пустыми строками.
6. Идентификаторы (названия переменных, типов, подпрограмм) должны быть значимыми настолько, чтобы читающий текст программы мог понимать их смысл без присутствия рядом автора. При необходимости объявление переменной или типа может сопровождаться комментарием.
7. Текст программы должен содержать комментарии, отражающие функциональное назначение того или иного блока программы, структуру программы.

### **Описание программы**

Документ "Описание программы" содержит:

- общие сведения (обозначение наименование программы, программное обеспечение, необходимое для функционирования программы, языка программирования, на которых написана программа);
- функциональное назначение (классы решаемых задач, сведения о функциональных ограничениях на применение);
- описание логической структуры (алгоритм программы, используемые методы, структура программы с описанием составных частей и связи между ними);
- используемые технические средства (типы ЭВМ и устройств, которые используются при работе программы);
- вызов и загрузка (способ вызова программы с соответствующего носителя данных);
- входные данные (характер, организация и предварительная подготовка входных данных, а также их формат, описание и способ кодирования);
- выходные данные (характер и организация выходных данных, а также их формат, описание и способ кодирования).

Описание логической структуры программы следует сопровождать блок-схемой программы.

Документ "Описание программы" может содержать также схемы данных, схемы взаимодействия программ, схемы ресурсов системы и проч., оформленные в соответствии с ГОСТ 19.701-90.

### **Описание применения**

Документ "Описание применения" относится к эксплуатационным документам и состоит из следующих разделов:

- назначение программы (возможности, основные характеристики, ограничения области применения);
- условия применения (требования к техническим и программным средствам, общие характеристики входной и выходной информации, а также

требования и условия организационного, технического и технологического характера);

- описание задачи (указываются определения задачи и методы её решения);
- входные и выходные данные.

#### **Руководство системного программиста**

Документ "Руководство системного программиста" относится к эксплуатационным документам и включается в программную документацию, если разработанный программный продукт требует обслуживания системным программистом. Документ состоит из следующих разделов:

- общие сведения о программе (назначение и функции программы, сведения о технических и программных средствах, обеспечивающих выполнение данной программы);
- структура программы (сведения о структуре, взаимосвязи между модулями программы и с другими программами);
- настройка программы (настройка на состав технических средств, выбор функций и т. п.);
- проверка программы (способы и методики проверки, контрольные примеры, методы прогона, результаты);
- дополнительные возможности;
- сообщения системному программисту (тексты сообщений, выдаваемых в ходе выполнения настройки, проверки программы, в ходе выполнения программы и описание действий, которые необходимо предпринять по этим сообщениям).

#### **Руководство программиста**

Документ "Руководство программиста" относится к эксплуатационным документам и включается в программную документацию, если разработанный программный продукт требует обслуживания программистом. Документ состоит из следующих разделов:

- назначение и условия применения программы (назначение и функции программы, сведения о технических и программных средствах, обеспечивающих выполнение данной программы);
- характеристики программы (временные характеристики, режимы работы, средства контроля правильности выполнения и т. п.);
- обращение к программе (способы передачи управления и параметров данных);
- входные и выходные данные (формат и кодирование);
- сообщения (тексты сообщений, выдаваемых программисту или оператору в ходе выполнения программы и описание действий, которые необходимо предпринять по этим сообщениям).

#### **Руководство оператора**

Документ "Руководство оператора" относится к эксплуатационным документам и состоит из следующих разделов:

- назначение программы (информация, достаточная для понимания функций программы и её эксплуатации);
- условия выполнения программы (минимальный и/или максимальный набор технических и программных средств и т. п.);
- выполнение программы (последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы; описываются функции, форматы и возможные варианты команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды);
- сообщения оператору (тексты сообщений, выдаваемых оператору в ходе выполнения программы и описание действий, которые необходимо предпринять по этим сообщениям).

## **Практическая работа № 21**

**Тема:** Разработка пояснительной записки программного продукта приложения.

**Цель:** Разработать пояснительную записку

Задание.

Для готового программного модуля создать пояснительную записку. Согласовать терминологию предметной области. Согласовать компьютерную терминологию.

**Методические указания:**

Документация должна содержать необходимые сведения по установке, обеспечению надёжной работы продукта, справочное пособие для пользователя, демонстрационные версии, примеры документов, создаваемых при помощи данного программного продукта, обучающие программы.

Программная документация, включает:

1. техническое задание (назначение, область применения программы, требования, предъявляемые к программе);
2. текст программы (запись программы с необходимыми комментариями);
3. описание программы (сведения о логической структуре и функционировании программы);
4. пояснительная записка (схема алгоритма, общее описание алгоритма и/или функционирования программы, обоснование принятых решений);
5. эксплуатационные документы.

### **Пояснительная записка**

Программный документ "Пояснительная записка" составляется на стадии эскизного или технического проектов программы. Как правило, на стадии рабочего проекта не используется. Включает разделы:

1. Введение.
2. Назначение и область применения.
3. Технические характеристики.
4. Ожидаемые технико-экономические показатели.
5. Источники, используемые при разработке.

Введение содержит наименование программы и обозначение темы разработки, документы, на основе которых ведётся разработка.

В назначении и области применения указывают назначение программы, краткую характеристику области применения программы.

Технические характеристики содержат:

Постановка задачи на разработку программы, описание применяемых математических методов и различных ограничений, связанных с выбранным математическим аппаратом.

Описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи, возможного взаимодействия программы с другими программами.

Описание и обоснование выбора метода организации входных и выходных данных.

Описание и обоснование выбора состава технических и программных средств на основе проведённых расчётов и анализов, распределение носителей данных, которые использует программа.



#### 4. ШКАЛА ОЦЕНКИ ОБРАЗОВАТЕЛЬНЫХ ДОСТИЖЕНИЙ

Баллы	Качественная оценка	Количественная оценка
91-100	отлично	«5»
76-90	хорошо	«4»
61-75	удовлетворительно	«3»
менее 61	неудовлетворительно	«2»
более 61	зачтено	
менее 61	не зачтено	

#### 5. ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ И ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

**Основная литература:**

12. Брайен, Керниган В. Язык программирования С : учебник / Брайен Керниган В., Деннис Ричи М. — Москва : Интуит НОУ, 2016. — 313 с. — URL: <https://book.ru/book/918294>
13. Александров, Э.Э. Программирование на языке С в Microsoft Visual Studio 2010 : курс лекций / Александров Э.Э., Афонин В.В. — Москва : Интуит НОУ, 2016. — 570 с. — URL: <https://book.ru/book/918122>
14. Алексеев, Е.Р. Программирование на языке С++ в среде Qt Creator : курс лекций / Алексеев Е.Р., Злобин Г.Г., Костюк Д.А., Чеснокова О.В., Чмыхало А.С. — Москва : Интуит НОУ, 2016. — 715 с. — URL: <https://book.ru/book/918128>
15. Построение сетей на базе коммутаторов и маршрутизаторов : курс лекций / — Москва : Интуит НОУ, 2016. — 330 с. — URL: <https://book.ru/book/917981>
16. Смирнова, Е.В. Построение коммутируемых компьютерных сетей : курс лекций / Смирнова Е.В., Баскаков И.В., Пролетарский А.В., Федотов Р.А. — Москва : Интуит НОУ, 2016. — 428 с. — URL: <https://book.ru/book/917979>
17. Кумскова, И.А. Базы данных : учебник / Кумскова И.А. — Москва : КноРус, 2020. — 400 с. — (СПО). — ISBN 978-5-406-07467-1. — URL: <https://book.ru/book/932493>
18. Грошев, А.С. Основы работы с базами данных : курс лекций / Грошев А.С. — Москва : Интуит НОУ, 2016. — 255 с. — URL: <https://book.ru/book/917933>
19. Пакулин, В.Н. 1С:Бухгалтерия 8. 1 : учебное пособие / Пакулин В.Н. — Москва : Интуит НОУ, 2016. — 67 с. — URL: <https://book.ru/book/917496>
20. Заика, А.А. Практика бухгалтерского учета в 1С:Бухгалтерии 8 : курс лекций / Заика А.А. — Москва : Интуит НОУ, 2016. — 526 с. — URL: <https://book.ru/book/917985>
21. Угринович, Н.Д. Информатика : учебник / Угринович Н.Д. — Москва : КноРус, 2020. — 377 с. — (СПО). — ISBN 978-5-406-07314-8. — URL: <https://book.ru/book/932057>
22. Прохорский, Г.В. Информатика : учебное пособие / Прохорский Г.В. — Москва : КноРус, 2020. — 240 с. — (СПО). — ISBN 978-5-406-07612-5. — URL: <https://book.ru/book/936152>

**Электронные ресурсы:**

4. ЭБС ИЗДАТЕЛЬСТВА "BOOK.RU" КОЛЛЕКЦИЯ СПО <https://www.book.ru/>
5. ЭБС ИЗДАТЕЛЬСТВА "ЮРАЙТ" <https://urait.ru>
6. ЭБС ИЗДАТЕЛЬСТВА "ЛАНЬ" <https://e.lanbook.com>

**Дополнительная литература:**

12. Макарова, Н.В. Основы программирования : учебник / Макарова Н.В., Нилова Ю.Н., Зеленина С.Б., Лебедева Е.В. — Москва : КноРус, 2021. — 451 с. — ISBN 978-5-406-03394-4. — URL: <https://book.ru/book/936582>

13. Фридман, А.Л. Язык программирования С : курс лекций / Фридман А.Л. — Москва : Интуит НОУ, 2016. — 218 с. — ISBN 978-5-9556-0017-8. — URL: <https://book.ru/book/918295>
14. Златопольский, Д.М. Программирование: типовые задачи, алгоритмы, методы : учебное пособие / Златопольский Д.М. 4-е изд. — Москва : Лаборатория знаний, 2020. — 224 с. — ISBN 978-5-00101-789-9. — URL: <https://book.ru/book/936428>
15. Основы построения объединенных сетей по технологиям CISCO : курс лекций / — Москва : Интуит НОУ, 2016. — 285 с. — URL: <https://book.ru/book/917875>
16. Осипов, Д.Л. Системы управления базами данных : практикум / Осипов Д.Л., Огур М.Г., сост. — Ставрополь : Северо-Кавказский федеральный университет, 2017. — 148 с. — URL: <https://book.ru/book/929919>
17. Златопольский, Д.М. Программирование: типовые задачи, алгоритмы, методы : учебное пособие / Златопольский Д.М. 4-е изд. — Москва : Лаборатория знаний, 2020. — 224 с. — ISBN 978-5-00101-789-9. — URL: <https://book.ru/book/936428>
18. Попов, А.А. Эргономика пользовательских интерфейсов в информационных системах : учебное пособие / Попов А.А. — Москва : КноРус, 2020. — 304 с. — ISBN 978-5-406-07634-7. — URL: <https://book.ru/book/935936>
19. 1С: Бухгалтерия предприятия 8. 2 : практическое пособие / Селищев Н.В., под ред. — Москва : КноРус, 2020. — 385 с. — ISBN 978-5-406-07219-6. — URL: <https://book.ru/book/932749>
20. Операционная система Microsoft Windows XP : курс лекций / — Москва : Интуит НОУ, 2016. — 374 с. — URL: <https://book.ru/book/917813>
21. Назаров, С.В. Современные операционные системы : курс лекций / Назаров С.В., Широков А.И. — Москва : Интуит НОУ, 2016. — 351 с. — ISBN 978-5-9963-0416-5. — URL: <https://book.ru/book/918225>
22. Угринович, Н.Д. Информатика. Практикум : учебное пособие / Угринович Н.Д. — Москва : КноРус, 2020. — 264 с. — (СПО). — ISBN 978-5-406-07320-9. — URL: <https://book.ru/book/932058>

**Информационное обеспечение обучения требует наличия залов:** библиотеки, читального зала с выходом в сеть Интернет.