

МИНОБРНАУКИ РОССИИ

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



## **РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **ОП.05 Основы программирования**

#### **ПРОГРАММЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА**

по специальности

#### **09.02.03 Программирование в компьютерных системах**

Уровень подготовки: базовый

Год набора на ООП  
2020

Артем 2020



## СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ.....	21
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	23
5. ГЛОССАРИЙ.....	29
6. ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ, ВНЕСЕННЫХ В РАБОЧУЮ ПРОГРАММУ.....	133

# **1 ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ**

## **1.1 Область применения программы**

Рабочая программа учебной дисциплины «Основы программирования» вводится в соответствии с ФГОС СПО в профессиональный цикл в качестве обязательной общепрофессиональной дисциплины (ОП – «Общепрофессиональные дисциплины») программы подготовки специалистов среднего звена по специальности 09.02.03 Программирование в компьютерных системах.

Рабочая программа учебной дисциплины может быть использована в дополнительном профессиональном образовании при организации курсовой подготовки повышения квалификации кадров или их переподготовки, а также по всем направлениям профессиональной подготовки кадров.

## **1.2 Место дисциплины в структуре программы подготовки специалистов среднего звена:**

Учебная общепрофессиональная дисциплина «Основы программирования» входит в основную часть профессионального цикла ППССЗ по специальности 09.02.03 Программирование в компьютерных системах.

## **1.3 Цели и задачи дисциплины – требования к результатам освоения дисциплины:**

### **Цель дисциплины**

- побудить студента на освоение базовых понятий и терминов программирования.

### **Задачи дисциплины:**

- ознакомить студента с конструкциями языка программирования высокого уровня;
- ознакомить студента с технологией разработки программ на данном языке;
- показать основной смысл и применение структур данных и алгоритмов их обработки;
- научить базовым концепциям объектно-ориентированного.

### **В результате освоения дисциплины обучающийся должен уметь:**

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

### **В результате освоения дисциплины обучающийся должен знать:**

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования;
- принципы объектно-ориентированного программирования.

### **Требования к результатам освоения дисциплины:**

Процесс изучения дисциплины направлен на формирование общих компетенций, включающих в себя способность:

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. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

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.

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 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

#### **1.4 При изучении дисциплины рассматриваются:**

- базовые понятия программирования;
- основные конструкции языка программирования;
- теория алгоритмов;
- структуры данных;
- сортировка информации;
- поиск информации;
- объектно-ориентированное программирование (ООП).

#### **1.5 Рекомендуемое количество часов на освоение программы дисциплины:**

- максимальной учебной нагрузки обучающегося – 180 часов, в том числе:
- обязательной аудиторной учебной нагрузки обучающегося – 120 часов;
- самостоятельной работы обучающегося – 60 часов.

## 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1 Объем учебной дисциплины и виды учебной работы

Рабочая программа дисциплины построена по модульно-блочному принципу. Под модулем понимается укрупненная логико-понятийная тема, характеризующаяся общностью используемого понятийно-терминологического аппарата. Каждый модуль состоит из одного или нескольких блоков. В таблице 1 указан объем времени, запланированный на реализацию всех видов учебной работы.

Таблица 1 – Объем времени, запланированный на реализацию всех видов учебной работы

<b>Вид учебной работы</b>	<b>Объем часов</b>
<b>Максимальная учебная нагрузка (всего)</b>	<b>180</b>
<b>Обязательная аудиторная учебная нагрузка (всего), в том числе:</b>	<b>120</b>
теоретическое обучение	58
практические занятия	62
контрольные работы	-
<b>Внеаудиторная самостоятельная работа обучающегося (всего):</b>	<b>60</b>
в том числе:	
исследовательская работа	14
творческая работа	16
работа с нормативной и справочной литературой	30
<i>Проверка знаний осуществляется с применением рейтинговой технологии. Промежуточная аттестация проводится в третьем семестре в форме экзамена (компьютерное тестирование).</i>	

**2.2 Тематический план по дисциплине в разрезе модулей по специальности  
09.02.03 Программирование в компьютерных системах**

Наименование разделов модулей и тем	Максимальная учебная нагрузка студента (час)	Внеаудиторная работа студента (час)	Количество аудиторных часов		
			Всего	в том числе:	
				Теоретическое обучение	ЛПЗ, семинары
<b>Раздел 1. Базовые понятия программирования</b>	<b>18</b>	<b>8</b>	<b>10</b>	<b>8</b>	<b>2</b>
Тема 1.1. Введение в дисциплину. Понятие информации. Архитектура компьютера.	4	2	2	2	-
Тема 1.2. Программирование как научная дисциплина. Среды программирования	4	2	2	2	-
Тема 1.3. Жизненный цикл программного обеспечения	4	2	2	2	-
Тема 1.4. Среда программирования	6	2	4	2	2
<b>Раздел 2. Основные конструкции языка программирования</b>	<b>80</b>	<b>20</b>	<b>60</b>	<b>20</b>	<b>40</b>
Тема 2.1. Язык программирования.	6	2	4	2	2
Тема 2.2. Константы.	8	2	6	2	4
Тема 2.3. Выражения.	6	2	4	2	2
Тема 2.4. Типы данных.	8	2	6	2	4
Тема 2.5. Операторы.	12	2	10	2	8
Тема 2.6. Подпрограммы.	8	2	6	2	4
Тема 2.7. Структура программы.	8	2	6	2	4
Тема 2.8. Файловые типы данных.	8	2	6	2	4
Тема 2.9. Ссылочные типы данных и указатели.	8	2	6	2	4
Тема 2.10. Динамические библиотеки.	8	2	6	2	4
<b>Раздел 3. Теория алгоритмов</b>	<b>12</b>	<b>4</b>	<b>8</b>	<b>6</b>	<b>2</b>
Тема 3.1. Интуитивное определение алгоритма.	2	-	2	2	-
Тема 3.2. Формальное определение алгоритма.	4	2	2	2	-

Тема 3.3. Способы записи алгоритмов.	6	2	4	2	2
<b>Раздел 4. Структуры данных</b>	<b>16</b>	<b>6</b>	<b>10</b>	<b>6</b>	<b>4</b>
Тема 4.1. Последовательные списки.	4	2	2	2	-
Тема 4.2. Связные списки.	6	2	4	2	2
Тема 4.3. Графы.	6	2	4	2	2
<b>Раздел 5. Сортировка информации</b>	<b>16</b>	<b>8</b>	<b>8</b>	<b>4</b>	<b>4</b>
Тема 5.1. Внутренняя сортировка (сортировка массивов).	8	4	4	2	2
Тема 5.2. Внешняя сортировка (сортировка файлов).	8	4	4	2	2
<b>Раздел 6. Поиск информации</b>	<b>12</b>	<b>4</b>	<b>8</b>	<b>4</b>	<b>4</b>
Тема 6.1. Поиск в массиве.	6	2	4	2	2
Тема 6.2. Поиск в таблице.	6	2	4	2	2
<b>Раздел 7. Объектно-ориентированное программирование (ООП).</b>	<b>26</b>	<b>10</b>	<b>16</b>	<b>10</b>	<b>6</b>
Тема 7.1. Основные понятия ООП.	4	2	2	2	-
Тема 7.2. Концепции объекта и класса.	6	2	4	2	2
Тема 7.3. Инкапсуляция, наследование и полиморфизм.	4	2	2	2	-
Тема 7.4. Поля, свойства, методы. События.	6	2	4	2	2
Тема 7.5. Области видимости в классах. Шаблоны классов. Интерфейсы.	6	2	4	2	2
<b>ВСЕГО</b>	<b>180</b>	<b>60</b>	<b>120</b>	<b>58</b>	<b>62</b>



## 2.3 Тематический план и содержание учебной дисциплины «Основы программирования»

Для характеристики уровня освоения учебного материала используются следующие обозначения:

- ознакомительный (узнавание ранее изученных объектов, свойств);
- репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)
- продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

Наименование разделов модулей и тем	Содержание учебного материала, лабораторные и практические работы, самостоятельная работа обучающихся,	Объём часов	Уровень освоения
1	2	3	4
<b>Раздел 1. Базовые понятия программирования</b>			
Тема 1.1. Введение в дисциплину. Понятие информации. Архитектура компьютера.	<b>Содержание учебного материала.</b> Определение, виды, свойства, меры информации. Информационные технологии. Аппаратное, программное и алгоритмическое обеспечение информационных технологий. Структура компьютера. Принципы Фон-Неймана. Понятие архитектуры компьютера. Краткий обзор современных многоядерных и многопроцессорных вычислительных систем.	2	1,2,3
	<b>Внеаудиторная самостоятельная работа №1:</b> Подготовка конспекта по теме «Развитие программирования как науки».	2	3
Тема 1.2. Программирование как научная дисциплина. Среды программирования	<b>Содержание учебного материала</b> Пользователи, программисты, системные администраторы. Прикладное, системное, научно-исследовательское программирование. Сложность - основная проблема программирования. Основные компоненты среды программирования: редактор, компилятор, отладчик. Классификация ошибок в программе: синтаксические, семантические ошибки, ошибки времени выполнения.	2	1,2,3
	<b>Внеаудиторная самостоятельная работа №2.</b> Подготовка конспекта по теме «Современный рейтинг сред разработки».	2	3
Тема 1.3. Жизненный цикл программного обеспечения	<b>Содержание учебного материала</b> Этапы жизненного цикла программного обеспечения. Анализ и спецификация. Проектирование. Реализация. Тестирование. Сопровождение.	2	1,2,3
	<b>Внеаудиторная самостоятельная работа №3.</b> Подготовка конспекта по теме «Виды жизненных циклов разработки программного обеспечения».	2	3
Тема 1.4. Среда программирования	<b>Содержание учебного материала</b> Основные компоненты среды программирования: редактор, компилятор, отладчик. Классификация	2	1,2,3

	ошибок в программе: синтаксические, семантические ошибки, ошибки времени выполнения.		
	<b>Практическое занятие №1.</b> Знакомство со средой программирования Delphi. Создание консольного приложения.	2	2,3
	<b>Внеаудиторная самостоятельная работа №4.</b> Подготовка конспекта по теме «История развития среды программирования Delphi».	2	3
<b>Раздел 2. Основные конструкции языка программирования</b>			
Тема 2.1. Язык программирования.	<b>Содержание учебного материала.</b> Алфавит, синтаксис и семантика языка программирования. Средства определения синтаксиса: расширенные формулы Бэкуса-Наура (РБНФ), синтаксические диаграммы. Классификация языков программирования по уровню абстракции. Обзор основных элементов языка программирования высокого уровня.	2	1,2,3
	<b>Практическое занятие №2.</b> Создание синтаксических диаграмм.	2	2,3
	<b>Внеаудиторная самостоятельная работа №5.</b> Выполнение дифференцированных заданий по теме «Формулы Бэкуса-Наура».	2	3
Тема 2.2. Константы.	<b>Содержание учебного материала.</b> Объявление и использование констант. Типизированные константы.	2	1,2,3
	<b>Практическое занятие №3.</b> «Создание программ с использованием констант»	2	2,3
	<b>Практическое занятие №4.</b> «Создание программ с применением типизированных констант»	2	2,3
	<b>Внеаудиторная самостоятельная работа №6.</b> Выполнение дифференцированных заданий по теме «Типизированные константы».	2	3
Тема 2.3. Выражения.	<b>Содержание учебного материала.</b> Классификация и приоритет операций. Построение и вычисление выражений.	2	1,2,3
	<b>Практическое занятие №5.</b> «Вычисление арифметических и логических выражений»	2	2,3
	<b>Внеаудиторная самостоятельная работа №7.</b> Выполнение дифференцированных заданий по теме «Построение и вычисление выражений»	2	3
Тема 2.4. Типы данных.	<b>Содержание учебного материала.</b> Классификация типов данных, примеры. Эквивалентность типов. Совместимость типов, совместимость по присваиванию. Преобразование типов.	2	1,2,3
	<b>Практическое занятие №6.</b> «Работа с переменными»	2	2,3
	<b>Практическое занятие №7.</b> «Применение методов приведения типов данных»	2	2,3
	<b>Внеаудиторная самостоятельная работа №8:</b> Подготовка конспекта по теме «Преобразование	2	3

	типов»		
Тема 2.5. Операторы.	<b>Содержание учебного материала.</b> Классификация операторов. Примеры операторов. Понятие структурного программирования. Теорема о структурном программировании.	2	1,2,3
	<b>Практическое занятие №8.</b> «Реализация программы с использованием структурных операторов»	2	2,3
	<b>Практическое занятие №9.</b> «Реализация программы с использованием операторов ветвления»	2	2,3
	<b>Практическое занятие №10.</b> «Реализация программы с использованием операторов циклов»	2	2,3
	<b>Практическое занятие №11.</b> «Реализация программы с использованием операторов безусловного перехода»	2	2,3
	<b>Внеаудиторная самостоятельная работа №9.</b> Выполнение дифференцированного задания по теме «Теорема о структурном программировании»	2	3
Тема 2.6. Подпрограммы.	<b>Содержание учебного материала.</b> Процедуры и функции. Спецификация подпрограммы. Формальные и фактические параметры подпрограммы. Способы передачи фактических параметров в подпрограмму. Понятие модульного программирования, модульная структура программы.	2	1,2,3
	<b>Практическое занятие №12.</b> «Создание и использование подпрограмм»	2	2,3
	<b>Практическое занятие №13.</b> «Передача параметров подпрограммам»	2	2,3
	<b>Внеаудиторная самостоятельная работа №10.</b> Выполнение дифференцированного задания по теме «Способы передачи фактических параметров в подпрограмму»	2	3
Тема 2.7. Структура программы.	<b>Содержание учебного материала.</b> Определяющее и использующее вхождения идентификатора. Области действия и видимости декларации. Локальные и глобальные переменные, побочный эффект подпрограммы. Распределение памяти программы: сегмент данных, сегмент стека, куча.	2	1,2,3
	<b>Практическое занятие №14.</b> «Реализация программы с использованием механизма областей видимости»	2	2,3
	<b>Практическое занятие №15.</b> Создание примеров «побочного эффекта»	2	2,3
	<b>Внеаудиторная самостоятельная работа №11.</b> Выполнение дифференцированного задания по теме «Области действия и видимости декларации»	2	3

Тема 2.8. Файловые типы данных.	<b>Содержание учебного материала</b> Классификация файловых типов. Стандартные подпрограммы работы с файлами. Типизированные файлы. Нетипизированные файлы.	2	1,2,3
	<b>Практическое занятие №16.</b> Чтение и запись информации в текстовые файлы	2	2,3
	<b>Практическое занятие №17.</b> Чтение и запись информации в двоичные файлы	2	2,3
	<b>Внеаудиторная самостоятельная работа №12.</b> Выполнение дифференцированного задания по теме «Стандартные подпрограммы работы с файлами»	2	3
Тема 2.9. Ссылочные типы данных и указатели.	<b>Содержание учебного материала.</b> Статические и динамические переменные программы. Стандартные подпрограммы работы с указателями.	2	1,2,3
	<b>Практическое занятие №18.</b> Работа с динамически выделяемой памятью	2	2,3
	<b>Практическое занятие №19.</b> Работа с динамическими массивами	2	2,3
	<b>Внеаудиторная самостоятельная работа №13.</b> Выполнение дифференцированного задания по теме «Статические и динамические переменные программы»	2	3
Тема 2.10. Динамические библиотеки.	<b>Содержание учебного материала.</b> Структура и назначение компонент библиотеки. Использование библиотек в разработке больших программных систем. Раннее связывание. Позднее связывание. Процедурный тип данных.	2	1,2,3
	<b>Практическое занятие №20.</b> Создание и использование динамических библиотек. Раннее связывание	2	2,3
	<b>Практическое занятие №21.</b> Создание и использование динамических библиотек. Позднее связывание	2	2,3
	<b>Внеаудиторная самостоятельная работа №14.</b> Подготовить конспект по теме «Динамические библиотеки»	2	3
<b>Раздел 3. Теория алгоритмов</b>			
Тема 3.1. Интуитивное определение алгоритма.	<b>Содержание учебного материала.</b> Основные свойства алгоритмов. Понятие исполнителя алгоритма. Основные методы разработки алгоритмов.	2	1,2,3
Тема 3.2. Формальное определение алгоритма.	<b>Содержание учебного материала.</b> Машина Тьюринга. Операции над машинами Тьюринга. Тезис Тьюринга. Нормальные алгоритмы Маркова. Принцип нормализации Маркова. Понятия самоприменимости алгоритма и алгоритмической неразрешимости.	2	1,2,3
	<b>Внеаудиторная самостоятельная работа №15.</b> Подготовить конспект по теме «Тезис Тьюринга»	2	3

Тема 3.3. Способы записи алгоритмов.	<b>Содержание учебного материала.</b> Понятие блок-схем. Основные элементы блок-схем. Понятие псевдокода.	2	1,2,3
	<b>Практическое занятие №22.</b> Построение блок-схем	2	2,3
	<b>Внеаудиторная самостоятельная работа №16.</b> Подготовить конспект по теме «Основные элементы блок-схем»	2	3
<b>Раздел 4. Структуры данных</b>			
Тема 4.1. Последовательные списки.	<b>Содержание учебного материала.</b> Понятие абстрактного типа данных. Классификация структур данных. Стек, очередь, дек.	2	1,2,3
	<b>Внеаудиторная самостоятельная работа №17.</b> Подготовить конспект по теме «Классификация структур данных»	2	3
Тема 4.2. Связные списки.	<b>Содержание учебного материала.</b> Однонаправленный список, двунаправленный список, циклический список.	2	1,2,3
	<b>Практическое занятие №23.</b> Создание связных списков и работа с ними.	2	2,3
	<b>Внеаудиторная самостоятельная работа №18.</b> Подготовить конспект по теме «Двунаправленный список и методы работы с ним»	2	3
Тема 4.3. Графы.	<b>Содержание учебного материала.</b> Методы реализации графов: матрица смежности, матрица инцидентности, списки смежных вершин, список ребер.	2	1,2,3
	<b>Практическое занятие №24.</b> Работа с графами.	2	2,3
	<b>Внеаудиторная самостоятельная работа №19.</b> Подготовить конспект по теме «Матрица смежности, матрица инцидентности»	2	3
<b>Раздел 5. Сортировка информации</b>			
Тема 5.1. Внутренняя сортировка (сортировка массивов).	<b>Содержание учебного материала.</b> Понятие сложности алгоритма сортировки. Основные алгоритмы внутренней сортировки: пузырьковая сортировка, шейкерная сортировка, сортировка простым выбором, сортировка простыми вставками, сортировка Шелла, сортировка слиянием, быстрая сортировка.	2	1,2,3
	<b>Внеаудиторная самостоятельная работа №20.</b> Подготовка конспекта по теме «Пузырьковая сортировка, шейкерная сортировка, сортировка простым выбором»	2	3
	<b>Практическое занятие №25.</b> «Реализация методов внутренней сортировки данных».	2	2,3
	<b>Внеаудиторная самостоятельная работа №21.</b> Подготовка конспекта по теме «Быстрая сортировка»	2	3
Тема 5.2. Внешняя сортировка	<b>Содержание учебного материала.</b> Отрезки файла. Операции разделения и слияния файлов.	2	1,2,3

(сортировка файлов).	Некоторые алгоритмы внешней сортировки: многофазная сортировка, сбалансированное слияние. Использование алгоритмов внутренней сортировки в сортировке последовательных файлов.		
	<b>Внеаудиторная самостоятельная работа №22.</b> Подготовка конспекта по теме: «Операции разделения и слияния файлов»	2	3
	<b>Практическое занятие №26.</b> «Реализация методов внешней сортировки».	2	2,3
	<b>Внеаудиторная самостоятельная работа №23.</b> Подготовить конспект по теме «Использование алгоритмов внутренней сортировки»	2	3
<b>Раздел 6. Поиск информации</b>			
Тема 6.1. Поиск в массиве.	<b>Содержание учебного материала.</b> Линейный поиск. Бинарный поиск.	2	1,2,3
	<b>Практическое занятие №27.</b> «Организация поиска в массиве».	2	2,3
	<b>Внеаудиторная самостоятельная работа №24.</b> Подготовка конспекта по теме «Линейный поиск»	2	3
Тема 6.2. Поиск в таблице.	<b>Содержание учебного материала.</b> Хеширование. Выбор хеш-функции. Разрешение коллизий: метод открытой адресации, метод цепочек.	2	1,2,3
	<b>Практическое занятие №28.</b> Использование хеш-функций для поиска в таблице.	2	2,3
	<b>Внеаудиторная самостоятельная работа №25.</b> Подготовка конспекта по теме «Хеш-функции»	2	
<b>Раздел 7. Объектно-ориентированное программирование (ООП)</b>			
Тема 7.1. Основные понятия ООП.	<b>Содержание учебного материала.</b> Парадигма объектно-ориентированного программирования. Сходства и различия ООП с другими парадигмами программирования: структурное, процедурное, модульное, функциональное программирование. Концепции объекта и класса. Инкапсуляция, наследование и полиморфизм. Поля, свойства, методы. События.	2	1,2,3
	<b>Внеаудиторная самостоятельная работа №26.</b> Подготовка конспекта по теме «Структурное, процедурное, модульное, функциональное программирование»	2	3
Тема 7.2. Концепции объекта и класса.	<b>Содержание учебного материала.</b> Различие в понятиях объект и класс. Описание класса. Объявление экземпляров класса. Создание объектов. Разрушение объектов. Объявление массива объектов.	2	1,2,3
	<b>Практическое занятие №29.</b> Создание класса THuman.	2	2,3
	<b>Внеаудиторная самостоятельная работа №27.</b> Подготовка конспекта по теме «Различие в понятиях объект и класс»	2	3

Тема 7.3. Инкапсуляция, наследование и полиморфизм.	<b>Содержание учебного материала.</b> Понятие инкапсуляции. Основное отличие объектного типа данных от типа данных запись. Использование механизма инкапсуляции. Понятие наследования. Класс TObject. Иерархия классов. Способы наследования членов класса. Понятие полиморфизма. Примеры использования механизма полиморфизма объектов. Обращение к членам полиморфных классов.	2	1,2,3
	<b>Внеаудиторная самостоятельная работа №28.</b> Подготовка конспекта по теме «Иерархия классов. Способы наследования членов класса»	2	3
Тема 7.4. Поля, свойства, методы. События.	<b>Содержание учебного материала.</b> Поля данных объекта. Область применения полей класса. Понятие метода. Типы методов. Статические и виртуальные методы. Абстрактные методы. Перегружаемые методы. Способы описания методов класса. Способы вызова методов. Основное отличие методов от подпрограмм. Обращение к элементам класса из методов. Основное назначение неявной переменной self и объектной переменной sender. Понятие свойства объекта. Основное отличие свойств от полей. Способы описания свойств объектов. Соглашение об использовании префиксов методов Set и Get. Операции присваивания с использованием свойств. Описание и применение свойства на чтение, запись и на обе операции. Обработка событий. Назначение методов на событие.	2	1,2,3
	<b>Практическое занятие №30.</b> Создание класса TCircle.	2	2,3
	<b>Внеаудиторная самостоятельная работа №29.</b> Подготовка конспекта по теме «Основное назначение неявной переменной self и объектной переменной sender»	2	3
Тема 7.5.Области видимости в классах. Шаблоны классов. Интерфейсы.	<b>Содержание учебного материала.</b> Понятие степени видимости элементов класса. Скрытие критических элементов класса. Пять уровней видимости. Директивы видимости: private, public, protected, published, automated. Создание и использование шаблона класса. Понятие абстрактного класса. Понятие интерфейса. Основное отличие интерфейса от класса. Назначение интерфейсов.	2	1,2,3
	<b>Внеаудиторная самостоятельная работа №30.</b> Подготовка конспекта по теме «Директивы видимости: private, public, protected, published, automated»	2	3
	<b>Практическое занятие №31.</b> Создание абстрактного класса TFigure.	2	2,3
<b>Итого по дисциплине часов</b>		<b>180</b>	

	<b>в том числе:</b>		
	<b>теоретическое обучение</b>	<b>58</b>	
	<b>практические занятия</b>	<b>62</b>	
	<b>Внеаудиторная самостоятельная работа</b>	<b>60</b>	

#### 2.4 Тематика практических занятий

В программу по дисциплине введены практические занятия, которые являются формой индивидуально-группового и практико-ориентированного обучения на основе реальных или модельных ситуаций применительно к виду и профилю профессиональной деятельности обучающегося. Занятия проводятся в диалоговом режиме, основными субъектами которых являются студенты. Практические занятия проводятся по 4 и 5 модулям.

Тематика обучающихся занятий представлена в таблице 4.

Таблица 4 – Тематика практических занятий

№ п/п	Учебно-образовательный модуль. Цели лабораторного практикума	Перечень и наименование лабораторных работ	Рекомендуется для области знаний (семестр)	
			3	4
1.	Практическая работа №1	Знакомство со средой программирования Delphi. Создание консольного приложения.	*	
	Цель: научить основным навыкам работы в среде программирования.			
2	Практическая работа №2	Создание синтаксических диаграмм	*	
	Цель: закрепить понятие синтаксических диаграмм.			
3	Практическая работа № 3	Создание программ с использованием констант	*	
	Цель: научить эффективно использовать константы в языке программирования.			
4	Практическая работа №4	Создание программ с применением типизированных констант	*	
	Цель: научить эффективно использовать типизированные константы и их инициализации в языке программирования.			
5.	Практическая работа №5	Вычисление арифметических и логических выражений	*	
	Цель: закрепить знания арифметических операций их приоритетов.			
6	Практическая работа №6	Работа с переменными	*	
	Цель: научить эффективно использовать переменные и типы данных.			
7	Практическая работа №7	Применение методов приведения типов данных	*	



	Цель: закрепить основные методы приведения различных типов данных.			
8	Практическая работа №8	Реализация программы с использованием структурных операторов	*	
	Цель: изучить операторы управления ходом работы программы.			
9	Практическая работа №9	Реализация программы с использованием операторов ветвления	*	
	Цель: изучить понятие ветвления			
10	Практическая работа №10	Реализация программы с использованием операторов циклов	*	
	Цель: научить эффективно использовать операторы циклов.			
11	Практическая работа №11	Реализация программы с использованием операторов безусловного перехода	*	
	Цель: изучить ошибки при использовании операторов безусловного перехода.			
12	Практическая работа №12	Создание и использование подпрограмм	*	
	Цель: изучить преимущества процедурного программирования.			
13	Практическая работа №13	Передача параметров подпрограммам	*	
	Цель: изучить основные методы передачи параметров подпрограмме.			
14	Практическая работа №14	Реализация программы с использованием механизма областей видимости	*	
	Цель: изучить понятие областей видимости.			
15	Практическая работа №15	Создание примеров «побочного эффекта»	*	
	Цель: изучить понятие «побочного эффекта».			
16	Практическая работа №16	Чтение и запись информации в текстовые файлы	*	
	Цель: изучить способы записи и чтения текстовых файлов.			
17	Практическая работа №17	Чтение и запись информации в двоичные файлы	*	
	Цель: изучить способы записи и чтения двоичных файлов.			
18	Практическая работа №18	Работа с динамически выделяемой памятью	*	
	Цель: изучить группу понятий: указатель,			

	динамическая память, разыменовывание указателя.			
19	Практическая работа №19	Работа с динамическими массивами	*	
	Цель: научить основным методам работы с динамическими массивами.			
20	Практическая работа №20	Создание и использование динамических библиотек. Раннее связывание.	*	
	Цель: изучить методы создания и раннего связывания динамических библиотек.			
21	Практическая работа №21	Создание и использование динамических библиотек. Позднее связывание.	*	
	Цель: изучить метод позднего связывания динамических библиотек.			
22	Практическая работа №22	Построение блок-схем.	*	
	Цель: повторить основные элементы блок-схем.			
23	Практическая работа №23	Создание связанных списков и работа с ними	*	
	Цель: закрепить основные методы работы со связанными списками.			
24	Практическая работа №24	Работа с графами	*	
	Цель: закрепить основные операции над графами.			
25	Практическая работа №25	Реализация методов внутренней сортировки данных	*	
	Цель: изучить основные способы сортировки данных в оперативной памяти.			
26	Практическая работа №26	Реализация методов внешней сортировки	*	
	Цель: изучить основные способы сортировки данных в текстовых файлах.			
27	Практическая работа №27	Организация поиска в массиве	*	
	Цель: изучить основные методы поиска информации в массиве.			
28	Практическая работа №28	Использование хеш-функций для поиска в таблице	*	
	Цель: показать преимущество поиска с помощью хэш-функций.			
29	Практическая работа №29	Создание класса THuman	*	
	Цель: изучить простой пример описания класса.			
30	Практическая работа №30	Создание класса TCircle	*	

	Цель: научить принципу инкапсуляции.			
31	Практическая работа №31	Создание абстрактного класса TFigure	*	
	Цель: углубление знаний по объектно-ориентированному программированию.			

## 2.5 Внеаудиторная самостоятельная работа

Программой определен объем самостоятельной работы студента, аудиторной и внеаудиторной самостоятельная работа студентов, - не менее 50% от общей обязательной нагрузки студента – и является важнейшим компонентом образовательного процесса, формирующего личность студента, его мировоззрение и культуру поведения, развивающим его способности к самообучению и повышению своего профессионального уровня.

Цели самостоятельной работы – формирование способностей к самостоятельному познанию и обучению, поиску литературы, обобщению, оформлению и представлению полученных результатов, их критическому анализу, поиску новых и неординарных решений, аргументированному отстаиванию своих предложений, умений подготовки выступлений и ведения дискуссий.

Самостоятельная работа проводится в период изучения отдельных тем курса по заданию преподавателя по рекомендуемой им учебной литературе, подготовки практическим занятиям, к рубежному контролю, экзамену или зачету, контрольной работе, к выполнению домашнего задания, предусмотренного рабочей учебной программой, к написанию рефератов, презентаций и доклада по ним.

Тематика СР носит профессионально-ориентированный характер и непосредственно связана с вопросами, изучаемыми по дисциплине. Тематика внеаудиторных самостоятельных работ представлена в таблице 5.

Таблица 5 – Тематика внеаудиторных самостоятельных работ

№ п/п	Учебно-образовательный модуль.	Тематика внеаудиторных самостоятельных работ	Рекомендуется для области знаний (семестры)	
			3	4
1	Базовые понятия программирования	<i>Развитие программирования как науки</i>	*	
2	Базовые понятия программирования	<i>Современный рейтинг сред разработки</i>	*	
3	Базовые понятия программирования	<i>Виды жизненных циклов разработки программного обеспечения</i>	*	
4	Базовые понятия программирования	<i>История развития среды программирования Delphi</i>	*	
5	Основные конструкции языка программирования	<i>Формулы Бэкуса-Наура</i>	*	
6	Основные конструкции языка программирования	<i>Типизированные константы</i>	*	
7	Основные конструкции языка программирования	<i>Построение и вычисление выражений</i>	*	
8	Основные конструкции языка программирования	<i>Преобразование типов</i>	*	
9	Основные конструкции языка программирования	<i>Теорема о структурном программировании</i>	*	
10	Основные конструкции языка программирования	<i>Способы передачи фактических параметров в подпрограмму</i>	*	
11	Основные конструкции языка программирования	<i>Области действия и видимости декларации</i>	*	
12	Основные конструкции языка программирования	<i>Стандартные подпрограммы работы с файлами</i>	*	
13	Основные конструкции языка программирования	<i>Статические и динамические переменные программы</i>	*	
14	Основные конструкции языка программирования	<i>Динамические библиотеки</i>	*	
15	Теория алгоритмов	<i>Тезис Тьюринга</i>	*	
16	Теория алгоритмов	<i>Основные элементы блок-схем</i>	*	
17	Структуры данных	<i>Классификация структур данных</i>	*	
18	Структуры данных	<i>Двунаправленный список и методы работы с ним</i>	*	
19	Структуры данных	<i>Матрица смежности, матрица инцидентности</i>	*	
20	Сортировка информации	<i>Пузырьковая сортировка, шейкерная сортировка, сортировка простым выбором</i>	*	
21	Сортировка информации	<i>Быстрая сортировка</i>	*	
22	Сортировка информации	<i>Операции разделения и слияния файлов</i>	*	
23	Сортировка информации	<i>Использование алгоритмов внутренней сортировки</i>	*	
24	Поиск информации	<i>Линейный поиск</i>	*	
25	Поиск информации	<i>Использование хеш-функций для</i>	*	

		<i>поиска в таблице</i>		
26	Объектно-ориентированное программирование (ООП)	<i>Структурное, процедурное, модульное, функциональное программирование</i>	*	
27	Объектно-ориентированное программирование (ООП)	<i>Различие в понятиях объект и класс</i>	*	
28	Объектно-ориентированное программирование (ООП)	<i>Иерархия классов. Способы наследования членов класса</i>	*	
29	Объектно-ориентированное программирование (ООП)	<i>Основное назначение неявной переменной <i>self</i> и объектной переменной <i>sender</i></i>	*	
30	Объектно-ориентированное программирование (ООП)	<i>Директивы видимости: <i>private</i>, <i>public</i>, <i>protected</i>, <i>published</i>, <i>automated</i></i>	*	

### **3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ**

#### **3.1 Требования к минимальному материально-техническому обеспечению**

Реализация программы дисциплины предполагает наличие лабораторий системного и прикладного программирования, информационно-коммуникационных систем.

##### **1. Лаборатория системного и прикладного программирования, оснащённая оборудованием:**

- посадочные места по количеству обучающихся (столы, стулья по числу посадочных мест);
- рабочее место преподавателя (стол, стул);
- учебно-методический комплекс по дисциплине (рабочие программы, календарно-тематические планы, разработки уроков по дисциплине, учебно-методическое обеспечение к каждому уроку, в т.ч. презентации к урокам, комплект видеоуроков, комплект контрольно-оценочных средств и др.);

- программное обеспечение общего назначения.

- локальная сеть.

##### **с техническими средствами обучения:**

- персональные компьютеры;
- ОС семейства Windows;
- пакет программных продуктов Microsoft Office;
- программы-архиваторы;
- интерактивная доска или мультимедиа проектор.

##### **2. Лаборатория информационно-коммуникационных технологий, оснащённая оборудованием:**

- посадочные места по количеству обучающихся (столы, стулья по числу посадочных мест);
- рабочее место преподавателя (стол, стул);
- комплект учебно-методической документации;
- программное обеспечение общего назначения.
- локальная сеть.

##### **с техническими средствами обучения:**

- персональные компьютеры;
- ОС семейства Windows;
- пакет программных продуктов Microsoft Office;
- программы-архиваторы;
- интерактивная доска или мультимедиа проектор.

#### **3.2 Информационное обеспечение реализации программы**

Для реализации программы библиотечный фонд филиала имеет печатные и /или электронные образовательные ресурсы, рекомендуемые для использования в образовательном процессе

##### **3.2.1 Основная литература:**

1. Баженова, И.Ю. Введение в программирование : курс лекций / Баженова И.Ю., Сухомлин В.А. — Москва : Интуит НОУ, 2016. — 411 с. — ISBN 978-5-9556-0077-9. — URL: <https://book.ru/book/917645>

2. Основы программирования : курс лекций / Мейер Бертран — Москва : Интуит НОУ, 2016. — 324 с. — ISBN 978-5-9963-0573-5. — URL: <https://book.ru/book/917903>

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>

##### **3.2.2 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>

### 3.3 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. 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>

### 3.4 Общие требования к организации образовательного процесса

Освоение обучающимися дисциплины проходит в условиях созданной образовательной среды в учебном заведении.

Общепрофессиональная дисциплина «Основы программирования» входит в профессиональный цикл ППССЗ по специальности **09.02.03 Программирование в компьютерных системах** и изучается одновременно с дисциплинами «Архитектура компьютерных систем», «Теория алгоритмов», с профессиональным модулем ПМ.04 Выполнение работ по рабочей профессии «Оператор электронно-вычислительных и вычислительных машин»

Изучение программы дисциплины завершается промежуточной аттестацией в форме экзамена, результаты которого оцениваются на основании выполнения студентами всех зачетных мероприятий по дисциплине.

### 3.5. Кадровое обеспечение образовательного процесса:

Требования к квалификации педагогических кадров, обеспечивающих обучение по дисциплине:

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

Таблица 6. - Кадровое обеспечение образовательного процесса

№ п/п	Характеристика педагогических работников					
	Фамилия, имя, отчество, должность по штатному расписанию	Какое образовательное учреждение окончил, специальность	Ученая степень и ученое (почетное) звание, квалификационная категория	Стаж педагогической (научно-педагогической) работы	Основное место работы, должность	Условия привлечения к педагогической деятельности
1	Волошин Евгений Владимирович	ФГБОУ ВО «ВГУЭС» в г. Артеме, бакалавр по направлению 09.03.03 Прикладная информатика	Преподаватель	2 г.9 мес	Филиал ВГУЭС в г. Артеме	штатный

## 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**4.1 Контроль и оценка результатов освоения** дисциплины осуществляется в процессе проведения практических занятий, тестирования, а также выполнения обучающимися индивидуальных заданий, исследований.

Таблица 7 – Формы и методы контроля и оценки результатов освоения учебной дисциплины по результатам текущего контроля и промежуточной аттестации

<b>Результаты обучения (освоенные умения, усвоенные знания)</b>	<b>Формы и методы контроля и оценки результатов обучения</b>
<b>Умения:</b>	
работать в среде программирования	Текущий контроль в форме выполнения и защиты практического задания и внеаудиторной самостоятельной работы
реализовывать построенные алгоритмы в виде программ на конкретном языке программирования	Текущий контроль в форме выполнения и защиты практического задания и внеаудиторной самостоятельной работы
<b>Знания:</b>	
этапы решения задачи на компьютере	Текущий контроль в форме выполнения и защиты практического задания и внеаудиторной самостоятельной работы
типы данных	Текущий контроль в форме выполнения и защиты практического задания и внеаудиторной самостоятельной работы
базовые конструкции изучаемых языков программирования	Текущий контроль в форме выполнения и защиты практического задания и внеаудиторной самостоятельной работы
принципы структурного и модульного программирования	Текущий контроль в форме выполнения и защиты практического задания и внеаудиторной самостоятельной работы
принципы объектно-ориентированного программирования	Текущий контроль в форме выполнения и защиты практического задания и внеаудиторной самостоятельной работы



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

Таблица 8 – Формы и методы контроля и оценки результатов развития общих компетенций и обеспечивающих их умений.

<b>Результаты (освоенные общие компетенции)</b>	<b>Основные показатели результатов подготовки</b>	<b>Формы и методы контроля</b>
ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес	понимание сущности и социальной значимости своей будущей профессии, проявление к ней устойчивого интереса	Наблюдение и оценка деятельности учащихся при проведении учебно-воспитательных мероприятий профессиональной направленности
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 выполнения поставленной задачи, объективная оценка своей работы.	Наблюдение и оценка активности учащихся при проведении учебно-воспитательных мероприятий профессиональной направленности.
ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	умение принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	Наблюдение и оценка деятельности учащихся при проведении учебно-воспитательных мероприятий профессиональной направленности
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	Наблюдение и оценка результатов деятельности обучающегося в процессе освоения образовательной программы на практических и семинарских занятиях, при выполнении внеаудиторных самостоятельных работ, рефератов.
ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.	демонстрация умения оперативно осуществлять операции, предлагаемые преподавателем, делать анализ и давать оценку полученной информации, в т.ч. и с использованием программного обеспечения	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.
ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.	коммуникабельность при взаимодействии с обучающимися и преподавателями в ходе обучения.	Экспертное наблюдение и оценка коммуникативной деятельности обучающегося в процессе освоения образовательной программы на практических занятиях, при

		<p>выполнении индивидуальных домашних заданий.</p> <p>Наблюдение и оценка использования учащихся коммуникативных методов и приемов при подготовке и проведении учебно-воспитательных мероприятий различной тематики.</p>
<p>ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.</p>	<p>умение брать ответственность за работу членов команды (подчиненных), за результат выполнения 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>
<p>ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.</p>	<p>способность к организации и планированию самостоятельных занятий при изучении дисциплины.</p> <p>демонстрация потребности в получении дополнительных знаний, возможностей самореализации</p>	<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>
<p>ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.</p>	<p>проявление интереса к инновациям в области профессиональной деятельности</p>	<p>Наблюдение и оценка результатов деятельности обучающегося в процессе освоения образовательной программы на практических и семинарских занятиях, при выполнении внеаудиторных самостоятельных работ, рефератов</p>

### 4.3 Контроль и оценка сформированности профессиональных компетенций обучающихся

Таблица 9 – Формы и методы контроля и оценки результатов сформированности профессиональных компетенций обучающихся

<b>Результаты (освоенные профессиональные компетенции)</b>	<b>Основные показатели результатов подготовки</b>	<b>Формы и методы контроля</b>
ПК 1.1. Выполнять разработку спецификаций отдельных компонент.	- демонстрировать знание методов составления спецификаций отдельных компонент	Оценка в рамках текущего контроля: - результатов выполнения практических работ на практических занятиях; - результатов выполнения индивидуальных домашних заданий; - результатов тестирования; - результатов участия в семинарских занятиях
ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.	-демонстрировать умение разрабатывать код программного продукта на основе готовых спецификаций на уровне модуля	Оценка в рамках текущего контроля: - результатов выполнения практических работ на практических занятиях; - результатов выполнения индивидуальных домашних заданий; - результатов тестирования; - результатов участия в семинарских занятиях
ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.	-демонстрировать умение выполнять отладку программных модулей с использованием специализированных программных средств	Оценка в рамках текущего контроля: - результатов выполнения практических работ на практических занятиях; - результатов выполнения индивидуальных домашних заданий; - результатов тестирования; - результатов участия в семинарских занятиях
ПК 1.4. Выполнять тестирование программных модулей	- демонстрировать умение выполнять тестирование программных модулей.	Оценка в рамках текущего контроля: - результатов выполнения практических работ на практических занятиях; - результатов выполнения индивидуальных домашних заданий; - результатов тестирования; - результатов участия в семинарских занятиях
ПК 1.5. Осуществлять оптимизацию программного кода модуля.	- демонстрировать умение осуществлять оптимизацию программного кода модуля.	Оценка в рамках текущего контроля: - результатов выполнения практических работ на практических занятиях; - результатов выполнения индивидуальных домашних заданий; - результатов тестирования; - результатов участия в семинарских занятиях

		занятиях
ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.	- демонстрировать умение анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения	Оценка в рамках текущего контроля: - результатов выполнения практических работ на практических занятиях; - результатов выполнения индивидуальных домашних заданий; - результатов тестирования; - результатов участия в семинарских занятиях

Таблица 10 – Соответствие содержания дисциплины требуемым результатам обучения

№ п/п	Результаты обучения	Учебно-образовательные модули					
		1	2	3	4	5	6
1	<b>Обобщенные общекультурные и профессиональные компетенции</b>						
1.1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.	*	*	*			
1.2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.				*	*	
1.3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.						*
1.4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.		*	*	*		
1,5	Использовать информационно-коммуникационные технологии в профессиональной деятельности.			*		*	
1.6	Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями		*			*	
1.7	Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.		*				*
1.8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.	*		*		*	
1.9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	*			*		
1.10	Выполнять разработку спецификаций отдельных компонент				*		*
1.11	Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.					*	
1.12	Выполнять отладку программных модулей с использованием специализированных программных средств.			*		*	
1.13	Выполнять тестирование программных модулей		*				
1.14	Осуществлять оптимизацию программного кода модуля		*		*	*	
1.15	Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения		*		*		*
2.	<b>Дисциплинарные компетенции (знания, умения)</b>						
	знания:						

2.1	этапы решения задачи на компьютере	*		*	*		*
2.2	типы данных	*	*				
2.3	базовые конструкции изучаемых языков программирования		*				
2.4	принципы структурного и модульного программирования	*				*	*
2.5	принципы объектно-ориентированного программирования				*	*	*
	умения:						
2.6	работать в среде программирования			*	*		*
2.7	реализовывать построенные алгоритмы в виде программ на конкретном языке программирования					*	*

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

Таблица 11 - Оценка индивидуальных образовательных достижений и компетенций по результатам текущего контроля и промежуточной аттестации с применением рейтинговой технологии

№ п/п	Наименование работ	Всего баллов 100			
		Текущая аттестация от 0 до 40 баллов (1-8 неделя)		Семестровая аттестация от 41 до 100 баллов (9-16 неделя)	
		Оценка уровня освоения дисциплины	Оценка компетенций обучающихся	Оценка уровня освоения дисциплины	Оценка компетенций обучающихся
1	Теоретический материал	5	5	5	6
2	Конспект лекции	1	1	1	1
3	ВСР	5	7	5	7
5	Практические работы	4	7	4	6
6	Промежуточная аттестация			10	10
7	Посещаемость	5		5	
<b>Итого:</b>		<b>40</b>		<b>60</b>	

Таблица 12 - Перевод баллов в традиционную систему оценивания

Баллы	Качественная оценка	Количественная оценка
91-100	отлично	«5»
76-90	хорошо	«4»
61-75	удовлетворительно	«3»
менее 61	неудовлетворительно	«2»
более 60	зачтено	
менее 61	не зачтено	

## 5 ГЛОССАРИЙ

Таблица 12 – Глоссарий основных терминов и определений, изучаемых в дисциплине «Основы программирования»

<b>Алгоритм</b>	точное предписание исполнителю совершить определенную последовательность действий для достижения поставленной цели за конечное число шагов
<b>Язык программирования</b>	искусственный (формальный) язык, предназначенный для записи алгоритмов. Язык программирования задается своим описанием и реализуется в виде специальной программы: компилятора или интерпретатора.
<b>Интероперабельность</b>	способность к взаимодействию различных аппаратных и программных платформ.
<b>Масштабируемость программного обеспечения</b>	способность программного обеспечения корректно работать на малых и на больших системах с производительностью, которая увеличивается пропорционально вычислительной мощности системы.
<b>Self</b>	дополнительный параметр, доступный в любом методе класса. Позволяет обратиться к тому объекту, который вызвал данный метод. Поскольку все упоминаемые в методе поля, методы и свойства по умолчанию связываются с тем объектом, который вызвал данный метод, параметр Self явно указывается обычно лишь в операторах with для устранения возможной неоднозначности. Часто используется также в качестве параметра Sender при вызове обработчиков событий.
<b>Деструктор (destructor)</b>	особый метод класса, обеспечивающий разрушение объектов этого класса (и, в частности, освобождение занимаемой ими области в динамической памяти). Обычно имеет имя Destroy. Вызов деструктора, как правило, выполняется с помощью вспомогательного метода Free. Особенностью метода Free является то, что он не выполняет никаких действий, если значение объекта равно nil (вызов в такой ситуации деструктора Destroy приводит к исключительной ситуации EAccessViolation). Заметим, что автоматического присваивания объекту значения nil при разрушении данного объекта не происходит. Формы и компоненты, размещаемые на формах в процессе проектирования, разрушаются автоматически при завершении работы приложения.
<b>Инспектор объектов (object inspector)</b>	элемент интегрированной среды Delphi, позволяющий настраивать свойства и события форм и компонентов на этапе проектирования программы.
<b>Исключение (исключительная ситуация, exсeption)</b>	особый класс, а также объект этого класса, используемый при обработке ошибок или особых ситуаций, возникающих при выполнении приложения. С каждым видом ошибки связывается свое исключение. Общим предком исключений является класс Exсeption. Имена классов-исключений, в отличие от имен прочих стандартных классов, начинаются с префикса «E». По умолчанию любое исключение обрабатывается путем вывода на экран сообщения о возникновении данного исключения (после этого консольное приложение завершает работу, а графическое приложение продолжает выполняться). Для явной обработки исключений в программе надо использовать конструкции try – exсept или try – finally, называемые защищенными блоками, или try-блоками (try-блоки второго типа в примерах из данной книги не используются). Try-блоки могут быть вложенными. В разделе exсept внешнего try-блока обрабатываются исключения, для которых не предусмотрено обработчика во внутреннем try-блоке. Если ни в одном try-блоке не предусмотрена



	<p>обработка исключения данного типа, то для него выполняется обработка по умолчанию (см. выше). Уже обработанное исключение можно возбудить повторно с помощью оператора raise. Это приведет к выполнению обработчиков данного исключения из внешних try-блоков, а при их отсутствии — к обработке данного исключения по умолчанию.</p>
<b>Класс (классовый тип, class)</b>	<p>структурный тип, состоящий из фиксированного числа элементов — полей, методов и свойств. Описывается с помощью ключевого слова class. При описании допускает наследование от уже определенных классов. Имена стандартных классов (кроме классов-исключений) начинаются с префикса «T», однако это лишь соглашение, следовать которому необязательно.</p>
<b>Компонент (component)</b>	<p>класс, порожденный от базового компонента — класса TComponent, а также объект этого класса. Почти все компоненты, за исключением тех, которые «встроены» в другие компоненты (например, ToolButton или TabSheet), располагаются в палитре компонентов и могут быть перенесены на форму при проектировании программы. Форма также считается особым компонентом. Многие свойства компонентов могут быть настроены в инспекторе объектов на этапе проектирования программы. В названии компонента обычно опускают префикс «T», входящий в имя соответствующего класса: так, компонент типа TEdit называется просто Edit. Среди компонентов выделяют визуальные компоненты (потомки класса TControl), которые отображаются на форме при выполнении программы, а среди визуальных — оконные компоненты (потомки класса TWinControl), которые могут получать фокус ввода и обрабатывать события от клавиатуры.</p>
<b>Конструктор (constructor)</b>	<p>особый метод класса, обеспечивающий инициализацию объектов этого класса. Обычно имеет имя Create и вызывается следующим образом:</p> <pre>&lt;имя инициализируемого объекта&gt; := &lt;имя класса&gt;.Create(&lt;список параметров&gt;);</pre> <p>Попытка доступа к методам или свойствам неинициализированного объекта приводит к исключительной ситуации EAccessViolation с сообщением «Read of address FFFFFFFF». Инициализация форм обычно проводится в файле проекта. Компоненты, размещаемые на форме в процессе проектирования, инициализируются автоматически при инициализации формы.</p>
<b>Метод (method)</b>	<p>процедура или функция, включенная в описание класса. Совокупность методов определяет «действия», которые могут выполнять объекты данного класса.</p>
<b>Модуль (unit)</b>	<p>текстовый файл с расширением PAS, содержащий фрагмент программного кода. Играет роль «библиотеки», которую можно подключать к файлам проекта и другим модулям. Модули могут компилироваться независимо друг от друга; откомпилированные модули имеют расширение DCU. Каждая форма, входящая в приложение, описывается в отдельном модуле. Текст модуля начинается с ключевого слова unit и состоит из четырех секций: interface (содержит описания имен, доступных для внешних программ и других модулей), implementation (секция реализации), initialization, finalization. Две последние секции являются необязательными.</p>
<b>Наследование (inheritance)</b>	<p>механизм, позволяющий определять новые классы на основе уже существующих. При этом новый класс («непосредственный потомок») автоматически получает все поля, методы и свойства</p>

	<p>существующего класса («непосредственного предка»). Методы и свойства предка в классе-потомке могут быть изменены. Кроме того, в классе-потомке могут быть описаны новые поля, методы и свойства. Все классы в Object Pascal наследуются от базового класса TObject, передающего своим потомкам, в частности, конструктор Create, деструктор Destroy и связанный с ним метод Free. Все компоненты наследуются от класса TComponent, а все исключения — от класса Exception.</p>
<b>Объект (object)</b>	<p>переменная классового типа. Перед использованием объекта необходимо его инициализировать с помощью конструктора, а после использования — разрушить с помощью деструктора. В Object Pascal все объекты распределяются динамически, поэтому имя объекта фактически представляет собой указатель на ту область памяти, которая содержит данный объект. Для доступа к полям, методам и свойствам объекта используется составное обозначение: &lt;имя объекта&gt;.&lt;имя поля, метода или свойства&gt;</p>
<b>Операция as</b>	<p>обеспечивает приведение объекта к новому классовому типу. Используется в выражении &lt;имя объекта&gt; as &lt;имя класса&gt; которое позволяет обращаться к указанному объекту как к объекту указанного класса. Если фактический тип объекта (тип времени выполнения) не может быть приведен к указанному классу, то возникает исключение EInvalidCast. Приведение типа часто применяется к параметру Sender в обработчиках событий.</p>
<b>Операция is</b>	<p>позволяет определить тип времени выполнения для данного объекта. Выражение &lt;имя объекта&gt; is &lt;имя класса&gt; возвращает True, если тип объекта совпадает с указанным классом или наследуется от указанного класса. В противном случае возвращается False.</p>
<b>Палитра компонентов (component palette)</b>	<p>элемент интегрированной среды Delphi, позволяющий добавлять в приложение компоненты на этапе проектирования программы (компоненты при этом размещаются на форме).</p>
<b>Поле (field)</b>	<p>элемент класса, содержащий данные определенного типа. Для различных объектов данного класса поля могут принимать различные значения, определяя тем самым «текущее состояние» объектов. Из внешних программ поля, как правило, не вызываются. Поля используются при реализации методов и свойств класса.</p>
<b>Приложение (application)</b>	<p>синоним исполняемой программы. Различают графические приложения (GUI-приложения), реализованные с использованием графического интерфейса Windows (GUI), и консольные приложения, выполняемые в особом консольном окне (консольное окно имитирует дисплей, находящийся в текстовом режиме). Специальным видом графического приложения является MDI-приложение, в котором подчиненные формы (дочерние окна) располагаются внутри главной формы (родительского окна).</p>
<b>Свойство (property)</b>	<p>элемент класса, снабженный особыми методами для своего чтения и записи. Обращение к свойству из программы выглядит как обращение к полю, однако фактически приводит к вызову связанных с ним методов. Это обеспечивает, в частности, выполнение необходимых побочных действий при изменении свойства (например, при изменении свойства Width для некоторого визуального компонента автоматически производится перерисовка этого компонента). Многие</p>

	свойства компонентов отображаются в инспекторе объектов и могут изменяться на этапе проектирования. Доступ к другим свойствам возможен только во время выполнения программы. Среди этих последних свойств могут быть свойства только для чтения, непосредственное изменение которых в программе запрещено.
<b>Событие (event)</b>	особое свойство компонента, позволяющее связывать с компонентом обработчик события — процедуру, которая вызывается в определенной ситуации (например, при получении компонентом фокуса или щелчке на нем мышью). Имена событий начинаются с префикса «On». События компонента и связанные с ними обработчики указываются на вкладке «Events» инспектора объектов.
<b>Тип времени выполнения (run-time type)</b>	фактический тип объекта, который указывается при его инициализации с помощью конструктора. Может не совпадать с типом, использованным при описании данного объекта. Требуется лишь, чтобы класс, использованный при описании объекта, содержался в цепочке предков для класса, использованного при инициализации объекта. Например, объект, описанный как TObject, можно инициализировать как TBitmap (с помощью конструктора TBitmap.Create), но не наоборот. Определить тип времени выполнения инициализированного объекта можно с помощью операции is.
<b>Файл проекта (project file)</b>	файл с расширением DPR, содержащий исходный текст программы (этот текст начинается с ключевого слова program). При создании нового проекта генерируется автоматически. В случае графических приложений, как правило, не редактируется, поскольку содержательная часть программного кода обычно связывается с формами и располагается в модулях (которые подключаются к DPR-файлу с помощью конструкции uses). Имя откомпилированного файла приложения (EXE-файла) совпадает с именем DPR-файла.
<b>Файл ресурсов (resource file)</b>	двоичный файл, содержащий пиктограммы, растровые изображения, курсоры и другие данные, используемые в приложении. Файл ресурсов приложения имеет расширение RES и имя, совпадающее с именем файла проекта. Файл ресурсов формы имеет расширение DFM и имя, совпадающее с именем модуля, в котором форма описана (начиная с Delphi версии 6, DFM-файлы по умолчанию имеют текстовый формат). Эти файлы ресурсов всегда входят в проект и создаются автоматически. Кроме них в приложении могут использоваться дополнительные файлы ресурсов.
<b>Форма (form)</b>	класс, порожденный от класса TForm, а также объект этого класса. Форма представляет собой объектную реализацию окна операционной системы Windows. Графическое приложение может включать несколько форм, одна из которых считается главной (главная форма отображается на экране при запуске приложения; закрытие главной формы приводит к завершению работы приложения). Каждая форма описывается в виде нового класса (потомка TForm) в отдельном модуле. Свойства и события формы, как и обычных компонентов, можно задать с помощью инспектора объектов.

**6 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ, ВНЕСЕННЫХ В РАБОЧУЮ ПРОГРАММУ**

№ изменения, дата внесения изменения, № страницы с изменением:

**БЫЛО:**

**СТАЛО:**

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 рабочей программы учебной дисциплины ОП.05 Основы программирования по специальности 09.02.03  
Программирование в компьютерных системах, представленной преподавателем кафедры экономики, управления и информационных технологий филиала ФГБОУ ВО «ВГУЭС» в г. Артеме Ематиной Н.И..**

**ЭКСПЕРТНОЕ ЗАКЛЮЧЕНИЕ**

№	Наименование экспертного показателя	Экспертная оценка	
		да	Нет
<b>Экспертиза оформления титульного листа и оглавления</b>			
1.	Наименование программы учебной дисциплины на титульном листе совпадает с наименованием дисциплины в тексте ФГОС и УП	да	
2.	Название филиала соответствует названию по Уставу	да	
3.	На титульном листе указан учебный цикл, код и наименование специальности	да	
4.	Оборотная сторона титульного листа заполнена	да	
5.	Нумерация страниц в «Содержании» верна	да	
<b>Экспертиза раздела 1 «Паспорт программы учебной дисциплины»</b>			
6.	Раздел 1 «Паспорт программы учебной дисциплины» имеется	да	
7.	Наименование программы дисциплины совпадает с наименованием на титульном листе	да	
8.	Пункт 1.1. «Область применения программы» заполнен	да	
9.	Пункт 1.2. «Место дисциплины в структуре основной профессиональной образовательной программы» заполнен	да	
10.	Пункт 1.3. «Цели и задачи дисциплины – требования к результатам освоения дисциплины» заполнен	да	
11.	Требования к умениям и знаниям соответствуют перечисленным в тексте ФГОС	да	
12.	Вариативная часть отражена (при наличии)	не предусмотрена	
13.	ПК, на которые ориентировано содержание дисциплины, указаны	да	
14.	ОК, формируемые в процессе изучения дисциплины, указаны	да	
15.	Подстрочные надписи удалены	да	
16.	Пункт 1.4. «Рекомендуемое количество часов на освоение программы дисциплины» заполнен	да	
17.	Перечислены виды самостоятельной работы	да	
18.	Указанное количество часов в графе «Итого» соответствует учебному плану	да	
<b>Экспертиза раздела 2 «Структура и содержание учебной дисциплины»</b>			
19.	Раздел 2. «Структура и содержание учебной дисциплины» имеется	да	
20.	Пункт 2.1. «Объем учебной дисциплины и виды учебной работы» заполнен	да	
21.	Таблица 2.3. «Тематический план и содержание учебной дисциплины» заполнена	да	

22.	Объем максимальной учебной нагрузки обучающегося в паспорте программы в таблицах 2.1, 2.2 и 2.3. совпадает	да	
23.	Объем обязательной аудиторной нагрузки в паспорте программы в таблицах 2.1., 2.2. и 2.3.совпадает	да	
24.	Объем времени, отведенного на самостоятельную работу обучающихся, в паспорте программы, таблицах 2.1., 2.2. и 2.3. совпадает	да	
25.	Объем в часах имеется во всех ячейках	да	
26.	Перечислены виды самостоятельной работы студентов, сформулированные через деятельность	да	
27.	Сумма по каждому столбцу равна максимальной нагрузке	да	
28.	В таблицах 2.2. и 2.3. все графы и строки заполнены	да	
<b>Экспертиза раздела 3 «Условия реализации программы дисциплины»</b>			
29.	Раздел 3 «Условия реализации программы дисциплины» имеется	да	
30.	Пункт 3.1. «Требования к минимальному материально-техническому обеспечению» заполнен	да	
31.	Пункт 3.2. «Информационное обеспечение обучения» заполнен в соответствии с требованиями ГОСТ по оформлению литературы	да	
32.	В пункте 3.2. указаны информационные основные и дополнительные источники для студентов и преподавателя	да	
33.	В списке основной литературы отсутствуют издания, выпущенные более 5 лет назад	да	
<b>Экспертиза раздела 4 «Контроль и оценка результатов освоения дисциплины»</b>			
34.	Раздел 4. «Контроль и оценка результатов освоения дисциплины» имеется	да	
35.	Наименования знаний и умений совпадают с указанными в п. 1.3	да	
<b>ИТОГОВОЕ ЗАКЛЮЧЕНИЕ</b>			
Программа дисциплины может быть направлена на содержательную экспертизу		да	

Разработчик программы: \_\_\_\_\_ Н.И. Ематина  
28 апреля 2020 г.

И.о. Зав. кафедрой \_\_\_\_\_ А.А.Власенко  
Зав. отделением \_\_\_\_\_ М.С.Словилова  
Методист УМЧ \_\_\_\_\_ Т.И.Теплякова  
30 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 дисциплины ОП.05 Основы программирования по специальности 09.02.03 Программирование в компьютерных системах, представленной преподавателем кафедры экономики, управления и информационных технологий филиала ФГБОУ ВО «ВГУЭС» в г. Артеме Ематиной Н.И..**

**ЭКСПЕРТНОЕ ЗАКЛЮЧЕНИЕ**

№	Наименование экспертного показателя	Экспертная оценка			Примечание
		да	нет	заключение отсутствует	
<b>Экспертиза раздела 1 «Паспорт программы учебной дисциплины»</b>					
1.	Требования к умениям и знаниям соответствуют перечисленным в ФГОС СПО (в т. ч. конкретизируют и/или расширяют требования ФГОС)	да			
2.	В пункте 1.3. указаны ПК и ОК, на формирование которых ориентировано содержание дисциплины	да			
3.	Вариативная часть содержит требования к результатам освоения дисциплины (при наличии)	не предусмотрена			
<b>Экспертиза раздела 2 «Структура и содержание учебной дисциплины»</b>					
4.	Содержание видов учебной деятельности соответствует требованиям, предъявляемым к результатам освоения дисциплины («уметь», «знать»).	да			
5.	Содержание учебной дисциплины разработано с ориентацией на формирование указанных в разделе 1 ПК и ОК	да			
6.	Структура программы учебной дисциплины соответствует принципу единства теоретического и практического обучения	да			
7.	Тематика лабораторных и/или практических работ соответствует формируемым умениям и ориентирована на подготовку к овладению ПК	да			
8.	Тематический план и содержание учебной дисциплины соответствует содержанию материала, указанного в разделе 1.	да			
9.	Уровни освоения соответствуют видам учебной деятельности в	да			



	разделе				
10.	Содержание самостоятельной работы студентов, в т.ч. внеаудиторной, направлено на выполнение требований к результатам освоения дисциплины	да			
11.	Формулировки самостоятельной работы понимаются однозначно	да			
12.	Разделы программы учебной дисциплины выделены дидактически целесообразно	да			
13.	Содержание учебного материала соответствует требованиям к формированию знаний и умений.	да			
14.	Объем времени достаточен для освоения указанного в содержании учебного материала	да			
15.	Объем и содержание лабораторных и практических работ определены дидактически целесообразно и соответствуют требованиям к умениям и знаниям	да			
16.	Примерная тематика курсовых работ соответствует целям и задачам освоения учебной дисциплины <i>(пункт заполняется, если в программе дисциплины предусмотрена курсовая работа)</i>	не предусмотрена			
<b>Экспертиза раздела 3 «Условия реализации программы дисциплины»</b>					
17.	Перечень учебных кабинетов (мастерских, лабораторий и др.) обеспечивает проведение всех видов лабораторных и практических работ, предусмотренных программой учебной дисциплины	да			
18.	Перечисленное оборудование обеспечивает проведение всех видов практических занятий, предусмотренных программой учебной дисциплины	да			
19.	Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники	да			
20.	Перечисленные Интернет-ресурсы актуальны и достоверны	да			
21.	Перечисленные источники соответствуют структуре и содержанию программы учебной дисциплины	да			
22.	Информационные источники указаны с учетом содержания дисциплины	да			
<b>Экспертиза раздела 4 «Контроль и оценка результатов освоения дисциплины»</b>					
23.	Основные показатели оценки результатов обучения позволяют однозначно диагностировать уровень освоения	да			

24.	Наименование форм и методов контроля и оценки освоенных умений и усвоенных знаний точно и однозначно описывает процедуру аттестации	да			
25.	Формы и методы контроля позволяют оценивать степень освоения умений и усвоения знаний	да			

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

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

Разработчик программы: \_\_\_\_\_ Н.И. Ематина  
28 апреля 2020 г.

И.о.Зав. кафедрой \_\_\_\_\_ А.А.Власенко  
Зав. отделением \_\_\_\_\_ М.С.Словилова  
Методист УМЧ \_\_\_\_\_ Т.И.Теплякова  
30 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 дисциплины ОП.05 Основы программирования по специальности 09.02.03  
Программирование в компьютерных системах, представленной преподавателем кафедры экономики, управления и информационных  
технологий филиала ФГБОУ ВО «ВГУЭС» в г. Артеме Ематиной Н.И..**

**ЭКСПЕРТНОЕ ЗАКЛЮЧЕНИЕ**

№	Наименование экспертного показателя	Экспертная оценка			Примечание
		да	нет	заключение отсутствует	
<b>Экспертиза раздела 1 «Паспорт программы учебной дисциплины»</b>					
1.	Требования к умениям и знаниям соответствуют перечисленным в ФГОС СПО (в т. ч. конкретизируют и/или расширяют требования ФГОС)	да			
2.	В пункте 1.3. указаны ПК и ОК, на формирование которых ориентировано содержание дисциплины	да			
3.	Вариативная часть содержит требования к результатам освоения дисциплины (при наличии)	не предусмотрена			
<b>Экспертиза раздела 2 «Структура и содержание учебной дисциплины»</b>					
4.	Содержание видов учебной деятельности соответствует требованиям, предъявляемым к результатам освоения дисциплины («уметь», «знать»).	да			
5.	Содержание учебной дисциплины разработано с ориентацией на формирование указанных в разделе 1 ПК и ОК	да			
6.	Структура программы учебной дисциплины соответствует принципу единства теоретического и практического обучения	да			
7.	Тематика лабораторных и/или практических работ соответствует формируемым умениям и ориентирована на подготовку к овладению ПК	да			
8.	Тематический план и содержание учебной дисциплины соответствует содержанию материала, указанного в разделе 1.	да			
9.	Уровни освоения соответствуют видам учебной деятельности в	да			

	разделе				
10.	Содержание самостоятельной работы студентов, в т.ч. внеаудиторной, направлено на выполнение требований к результатам освоения дисциплины	да			
11.	Формулировки самостоятельной работы понимаются однозначно	да			
12.	Разделы программы учебной дисциплины выделены дидактически целесообразно	да			
13.	Содержание учебного материала соответствует требованиям к формированию знаний и умений.	да			
14.	Объем времени достаточен для освоения указанного в содержании учебного материала	да			
15.	Объем и содержание лабораторных и практических работ определены дидактически целесообразно и соответствуют требованиям к умениям и знаниям	да			
16.	Примерная тематика курсовых работ соответствует целям и задачам освоения учебной дисциплины <i>(пункт заполняется, если в программе дисциплины предусмотрена курсовая работа)</i>	не предусмотрена			
<b>Экспертиза раздела 3 «Условия реализации программы дисциплины»</b>					
17.	Перечень учебных кабинетов (мастерских, лабораторий и др.) обеспечивает проведение всех видов лабораторных и практических работ, предусмотренных программой учебной дисциплины	да			
18.	Перечисленное оборудование обеспечивает проведение всех видов практических занятий, предусмотренных программой учебной дисциплины	да			
19.	Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники	да			
20.	Перечисленные Интернет-ресурсы актуальны и достоверны	да			
21.	Перечисленные источники соответствуют структуре и содержанию программы учебной дисциплины	да			
22.	Информационные источники указаны с учетом содержания дисциплины	да			
<b>Экспертиза раздела 4 «Контроль и оценка результатов освоения дисциплины»</b>					
23.	Основные показатели оценки результатов обучения позволяют однозначно диагностировать уровень освоения	да			

24.	Наименование форм и методов контроля и оценки освоенных умений и усвоенных знаний точно и однозначно описывает процедуру аттестации	да			
25.	Формы и методы контроля позволяют оценивать степень освоения умений и усвоения знаний	да			

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

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

Разработчик программы: \_\_\_\_\_ Н.И. Ематина  
28 апреля 2020 г.

Эксперты:  
\_\_\_\_\_ В.В.Неслюзов  
\_\_\_\_\_ О.В. Бажин  
30 апреля 2020 г.

МИНОБРНАУКИ РОССИИ

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



## **КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **ОП.05 Основы программирования**

**ПРОГРАММЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА**

по специальности

**09.02.03 Программирование в компьютерных системах**

Уровень подготовки: базовый

Год набора на ООП  
2020

Артем 2020

**СОГЛАСОВАНО**

Протокол заседания научно-методического совета  
от 18 мая 2020г. №7

Председатель  О.И. Иванюга

**РАССМОТРЕНО И ОДОБРЕНО**

на заседании кафедры ЭУИТ

Протокол № 14 от 06 мая 2020г.

И.о.зав.кафедрой  А.А. Власенко

Разработчик:  Н.И. Ематина

Преподаватель кафедры ЭУИТ

## СОДЕРЖАНИЕ

1. Общие положения .....	26
2. Результаты освоения дисциплины, подлежащие проверке.....	27
3. Распределение оценивания результатов обучения по видам контроля <b>Ошибка! Закладка не определена.</b>	
4. Распределение типов контрольных заданий по элементам знаний и умений <b>Ошибка! Закладка не определена.</b>	
5. Распределение типов и количества контрольных заданий по элементам знаний и умений, контролируемых на промежуточной аттестации .....	<b>Ошибка!</b>
<b>Закладка не определена.</b>	
6. Структура контрольных заданий .....	29
7. Шкала оценки образовательных достижений .....	66
8. Перечень используемых материалов, оборудования и информационных источников .....	66



## **1. Общие положения**

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу подготовки специалистов среднего звена ОП.05 Основы программирования.

КОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации в форме контрольной работы.

КОС разработаны в соответствии с:

– программой подготовки специалистов среднего звена (далее – ППССЗ) по специальности 09.02.03 Программирование в компьютерных системах, реализуемой в колледже;

– программой подготовки специалистов среднего звена ОП.05 Основы программирования.

## 2. Результаты освоения дисциплины, подлежащие проверке

Результаты обучения (освоенные умения, усвоенные знания)	Основные показатели оценки результатов
ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.	– демонстрация интереса к будущей профессии.
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 в области разработки и администрирования баз данных; – оценка эффективности и качества выполнения.
ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	– решение стандартных и нестандартных профессиональных задач в области разработки и администрирования баз данных.
ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.	– эффективный поиск необходимой информации; – использование различных источников, включая электронные.
ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.	– разрабатывать, программировать и администрировать базы данных
ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.	– взаимодействие с обучающимися, преподавателями и мастерами в ходе обучения
ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.	– самоанализ и коррекция результатов собственной работы
ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.	– организация самостоятельных занятий при изучении профессионального модуля
ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	– анализ инноваций в области разработки и администрирования баз данных
ПК 1.1. Выполнять разработку спецификаций отдельных компонент.	– создание отдельных компонент – выполнение спецификаций компонент
ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых	– выполнение создания кода программного продукта на уровне модуля в соответствии с требованиями к готовому программному продукту
ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.	– использование специализированных программных средств отладки программных модулей – разработка компонент программных модулей с использованием современных инструментальных средств и технологий.
ПК 1.4. Выполнять тестирование программных модулей.	– выполнение тестирования качества разработки 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 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.	– анализ проектной и технической документацию на уровне взаимодействия компонент программного обеспечения

### **3. Структура контрольных заданий**

#### **3.1. Задания текущего контроля**

##### **3.1.1 Практические задания**

#### **Тема 1.4. Среда программирования**

##### **Практическое задание № 1.**

**Тема:** Знакомство со средой программирования Delphi. Создание консольного приложения.

**Задание:** научиться основным навыкам работы в среде программирования.

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2

##### **Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

#### **Тема 2.1. Язык программирования.**

##### **Практическое задание № 2.**

**Тема:** Создание синтаксических диаграмм.

**Задание:** закрепить понятие синтаксических диаграмм.

**Проверяемые результаты обучения:** ПК 1.2, ПК 3.1

##### **Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

#### **Тема 2.2. Константы**

##### **Практическое задание № 3.**

**Тема:** Создание программ с использованием констант.

**Задание:** научиться эффективно использовать константы в языке программирования.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.5

##### **Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

##### **Практическое задание № 4.**

**Тема:** Создание программ с применением типизированных констант.

**Задание:** научиться эффективно использовать типизированные константы и их инициализации в языке программирования.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.5

##### **Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

#### **Тема 2.3. Выражения**

**Практическое задание № 5.**

**Тема:** Вычисление арифметических и логических выражений.

**Задание:** закрепить знания арифметических операций их приоритетов.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Тема 2.4. Типы данных****Практическое задание № 6.**

**Тема:** Работа с переменными.

**Задание:** научить эффективно использовать переменные и типы данных.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Практическое задание № 7.**

**Тема:** Применение методов приведения типов данных.

**Задание:** закрепить основные методы приведения различных типов данных.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Тема 2.5. Операторы.****Практическое задание № 8.**

**Тема:** Реализация программы с использованием структурных операторов.

**Задание:** изучить операторы управления ходом работы программы.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Практическое задание № 9.**

**Тема:** Реализация программы с использованием операторов ветвления.

**Задание:** изучить понятие ветвления.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Практическое задание № 10.**

**Тема:** Реализация программы с использованием операторов циклов.

**Задание:** научить эффективно использовать операторы циклов.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Практическое задание № 11.**

**Тема:** Реализация программы с использованием операторов безусловного перехода.

**Задание:** изучить ошибки при использовании операторов безусловного перехода.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Тема 2.6. Подпрограммы.****Практическое задание № 12.**

**Тема:** Создание и использование подпрограмм.

**Задание:** изучить преимущества процедурного программирования.

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Практическое задание № 13.**

**Тема:** Передача параметров подпрограммам.

**Задание:** изучить основные методы передачи параметров подпрограмме.

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Тема 2.7. Структура программы****Практическое задание № 14.**

**Тема:** Реализация программы с использованием механизма областей видимости

**Задание:** изучить понятие областей видимости.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Практическое задание № 15.**

**Тема:** Создание примеров «побочного эффекта».

**Задание:** изучить понятие «побочного эффекта».

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Тема 2.8. Файловые типы данных**

**Практическое задание № 16.**

**Тема:** Чтение и запись информации в текстовые файлы.

**Задание:** изучить способы записи и чтения текстовых файлов.

**Проверяемые результаты обучения:** ПК 1.2

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Практическое задание № 17.**

**Тема:** Чтение и запись информации в двоичные файлы.

**Задание:** изучить способы записи и чтения двоичных файлов.

**Проверяемые результаты обучения:** ПК 1.2

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Тема 2.9. Ссылочные типы данных и указатели**

**Практическое задание № 18.**

**Тема:** Работа с динамически выделяемой памятью.

**Задание:** изучить группу понятии: указатель, динамическая память, разыменовывание указателя.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Практическое задание № 19.**

**Тема:** Работа с динамическими массивами.

**Задание:** научить основным методам работы с динамическими массивами.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

### **Тема 2.10. Динамические библиотеки**

#### **Практическое задание № 20.**

**Тема:** Создание и использование динамических библиотек. Раннее связывание..

**Задание:** изучить методы создания и раннего связывания динамических библиотек.

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.5

#### **Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

#### **Практическое задание № 21.**

**Тема:** Создание и использование динамических библиотек. Позднее связывание..

**Задание:** изучить метод позднего связывания динамических библиотек.

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.5

#### **Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

### **Тема 3.3. Способы записи алгоритмов**

#### **Практическое задание № 22.**

**Тема:** Построение блок-схем..

**Задание:** повторить основные элементы блок-схем.

**Проверяемые результаты обучения:** ПК 1.2, ПК 3.1

#### **Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

### **Тема 4.2. Связные списки**

#### **Практическое задание № 23.**

**Тема:** Создание связных списков и работа с ними..

**Задание:** закрепить основные методы работы со связными списками.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

#### **Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

### **Тема 4.3. Графы**

#### **Практическое задание № 24.**

**Тема:** Работа с графами



**Задание:** закрепить основные операции над графами.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

### **Тема 5.1. Внутренняя сортировка (сортировка массивов)**

#### **Практическое задание № 25.**

**Тема:** Реализация методов внутренней сортировки данных.

**Задание:** изучить основные способы сортировки данных в оперативной памяти.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

### **Тема 5.2. Внешняя сортировка (сортировка файлов).**

#### **Практическое задание № 26.**

**Тема:** Реализация методов внешней сортировки.

**Задание:** изучить основные способы сортировки данных в текстовых файлах.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

### **Тема 6.1. Поиск в массиве.**

#### **Практическое задание № 27.**

**Тема:** Организация поиска в массиве.

**Задание:** изучить основные методы поиска информации в массиве.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

### **Тема 6.2. Поиск в таблице.**

#### **Практическое задание № 28.**

**Тема:** Использование хеш-функций для поиска в таблице.

**Задание:** показать преимущество поиска с помощью хэш-функций.

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Тема 7.2. Концепции объекта и класса.**

**Практическое задание № 29.**

**Тема:** Создание класса Thuman.

**Задание:** изучить простой пример описания класса.

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Тема 7.4. Поля, свойства, методы. События.**

**Практическое задание № 30.**

**Тема:** Создание класса Tcircle.

**Задание:** научить принципу инкапсуляции.

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**Тема 7.5. Области видимости в классах. Шаблоны классов. Интерфейсы.**

**Практическое задание № 31.**

**Тема:** Создание абстрактного класса Tfigure.

**Задание:** углубление знаний по объектно-ориентированному программированию.

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 4 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 80 минут

**3.1.2 Внеаудиторные самостоятельные работы**

**Внеаудиторная самостоятельная работа №1.** Подготовка конспекта по теме «Развитие программирования как науки».

**Проверяемые результаты обучения:** ПК 1.2

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №2.** Подготовка конспекта по теме «Современный рейтинг сред разработки».

**Проверяемые результаты обучения:** ПК 1.2

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №3.** Подготовка конспекта по теме «Виды жизненных циклов разработки программного обеспечения».

**Проверяемые результаты обучения:** ПК 1.2

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №4.** Подготовка конспекта по теме «История развития среды программирования Delphi».

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №5.**

Выполнение дифференцированных заданий по теме «Формулы Бэкуса-Наура».

**Проверяемые результаты обучения:** ПК 1.2, ПК 3.1

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №6.**

Выполнение дифференцированных заданий по теме «Типизированные константы».

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №7.** Выполнение дифференцированных заданий по теме «Построение и вычисление выражений»

Выполнение дифференцированных заданий по теме «Типизированные константы».

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №8:** Подготовка конспекта по теме «Преобразование типов»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №9.**

Выполнение дифференцированного задания по теме «Теорема о структурном программировании»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №10.** Выполнение дифференцированного задания по теме «Способы передачи фактических параметров в подпрограмму»

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №11.** Выполнение дифференцированного задания по теме «Области действия и видимости декларации»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №12.** Выполнение дифференцированного задания по теме «Стандартные подпрограммы работы с файлами»

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №13.** Выполнение дифференцированного задания по теме «Статические и динамические переменные программы»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №14.** Подготовка конспекта по теме «Динамические библиотеки»

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №15.** Подготовка конспекта по теме «Тезис Тьюринга»

**Проверяемые результаты обучения:** ПК 1.2, ПК 3.1

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №16.** Подготовка конспекта по теме «Основные элементы блок-схем»

**Проверяемые результаты обучения:** ПК 1.2, ПК 3.1

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №17.** Подготовка конспекта по теме «Классификация структур данных»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №18.** Подготовка конспекта по теме «Двунаправленный список и методы работы с ним»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №19.** Подготовка конспекта по теме «Матрица смежности, матрица инцидентности»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.  
 За не полностью выполненную работу выставляется – 1 балл.  
 За невыполненную работу выставляется – 0 баллов.  
**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №20.** Подготовка конспекта по теме «Пузырьковая сортировка, шейкерная сортировка, сортировка простым выбором»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.  
 За не полностью выполненную работу выставляется – 1 балл.  
 За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №21.** Подготовка конспекта по теме «Быстрая сортировка»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.  
 За не полностью выполненную работу выставляется – 1 балл.  
 За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №22.** Подготовка конспекта по теме: «Операции разделения и слияния файлов»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.  
 За не полностью выполненную работу выставляется – 1 балл.  
 За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №23.** Подготовка конспекта по теме «Использование алгоритмов внутренней сортировки»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.  
 За не полностью выполненную работу выставляется – 1 балл.  
 За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №24.** Подготовка конспекта по теме «Линейный поиск»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.  
 За не полностью выполненную работу выставляется – 1 балл.  
 За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №25.** Подготовка конспекта по теме «Хеш-функции»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №26.** Подготовка конспекта по теме «Структурное, процедурное, модульное, функциональное программирование»

**Проверяемые результаты обучения:** ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №27.** Подготовка конспекта по теме «Различие в понятиях объект и класс»

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №28.** Подготовка конспекта по теме «Иерархия классов. Способы наследования членов класса»

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №29.** Подготовка конспекта по теме «Основное назначение неявной переменной self и объектной переменной sender»

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

**Внеаудиторная самостоятельная работа №30.** Подготовка конспекта по теме «Директивы видимости: private, public, protected, published, automated»

**Проверяемые результаты обучения:** ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5

**Критерии оценки:**

За верное выполнение работы выставляется – 3 балла.

За не полностью выполненную работу выставляется – 1 балл.

За невыполненную работу выставляется – 0 баллов.

**Время выполнения:** 120 минут

## 6. Задания промежуточной аттестации

### 6.1 Тестовое задание

#### Раздел 1. Базовые понятия программирования

1. Какими знаками заканчивается большинство строк кода в Си++?
  - 1) ; (точка с запятой)
  - 2) , (запятая)
  - 3) . (точка)
  - 4) : (двоеточие)
2. Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?
  - 1) 1
  - 2) Программа не возвращает значение.
  - 3) 0
  - 4) -1
3. Какая из следующих записей - правильный комментарий в C++?
  - 1) /\* комментарий \*/
  - 2) {комментарий}
  - 3) \*\* комментарий \*\*
  - 4) \*/ комментарий \*/
4. Какая из следующих записей - правильный комментарий в C++?
  - 1) // комментарий
  - 2) /-- комментарий
  - 3) /\* комментарий
  - 4) # комментарий
5. Какая строка содержит только зарезервированные слова языка программирования C++?
  - 1) **if, else, for, while do, switch, continue, break**
  - 2) defaulted, goto, return, extern, private, public, protected
  - 3) sizeof, const, typedef, static, voided, enum, struct, union
  - 4) char, int, float, doubled, short, long, unsigned, signed
6. Какое подключение заголовочного файла библиотеки будет неверным?
  - 1) #include<iostream>
  - 2) #include "iostream"
  - 3) **#include <main>**
  - 4) #include "main"
7. Какие среды программирования (IDE) предназначены для разработки программных средств?
  - 1) **MS Visual Studio, NetBeans, QT Creator, RAD Studio, Dev-C++**
  - 2) MS Visual Studio, Code::Blocks, QT Creator, RAD Studio, MathCAD
  - 3) MS Visual Studio, Code::Blocks, QT Creator, AutoCAD, Eclipse
8. Какие скобки используются для подключения заголовочных файлов системных библиотек
  - 1) Квадратные
  - 2) Фигурные
  - 3) Круглые
  - 4) **Угловые**
9. Какие скобки определяют блок программного кода
  - 1) Квадратные
  - 2) **Фигурные**
  - 3) Круглые
  - 4) Угловые
10. Какие служебные символы используются для обозначения начала и конца блока кода?
  - 1) ( )
  - 2) [ ]
  - 3) <>
  - 4) { }
11. Какой заголовочный файл необходимо подключить, чтобы вызвать функцию isalpha()?
  - 1) ifstream.h



- 2) cstring  
 3) conio.h  
 4) **cctype**
12. Какую функцию должны содержать все программы на C++?  
 1) program()  
 2) **main()**  
 3) start()  
 4) system()
13. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:  
 1) **транслятор**  
 2) компилятор  
 3) интерпретатор  
 4) сканер
14. Укажите объектно-ориентированный язык программирования  
 1) C++  
 2) Java  
 3) **Все варианты ответов**  
 4) Delphi
15. Чтобы подключить заголовочный файл в программу на C++, например iostream необходимо написать:  
 1) include #iostream,h;  
 2) #include<>; с iostream.h внутри скобок  
 3) include (iostreamh)  
 4) **#include <> с iostreamвнутрискобок**
16. Язык программирования C++ разработал  
 1) **БьериСтрауструп**  
 2) Дональд Кнут  
 3) Никлаус Вирт  
 4) Кен Томпсон
17. Что означает std?  
 1) **Это название пространства имен, в котором находится функция cout**  
 2) Это название класса, у которого вызывается не статический метод cout  
 3) Это название класса, у которого вызывается статический метод cout  
 4) Это название класса, в котором находится функция cout
18. Библиотека, отвечающая за взаимодействие с системой (определяет функцию system() ), а также для функции для работы с генератором случайных чисел srand() и rand(), называется:  
 Ответ: cstdlib
19. Библиотека, отвечающая за работу с математическими функциями, называется:  
 Ответ: cmath
20. Библиотека, отвечающая за работу с потоками ввода-вывода, называется:  
 Ответ: iostream
21. Директива препроцессора, позволяющая подключать заголовочные файлы определяется ключевым словом:  
 Ответ: include
22. Какими символами начинается однострочный комментарий в языке C++ - это  
 Ответ: //
23. Какой файл имеет расширение \*.h в языке C++ (1 слово):  
 Ответ: заголовочный
24. Какой файл имеет расширение \*.cpp в языке C++ (1 слово):  
 Ответ: исходный
25. Название пространства имен, подключаемого в соответствии с новым стандартом языка:  
 Ответ: std
26. Программа, преобразующая исходный код (\*.cpp) на языке C++ в объектный код (\*.o)  
 Ответ: компилятор

27. Программа, преобразующая объектный код (\*.o) файлов проекта в исполняемый код (\*.exe)

Ответ: компоновщик

28. Программа, преобразующая исходный код с учетом всех директив (например #include )

Ответ: препроцессор

29. Смысл любого оператора в любом языке программирования определяет...

Ответ: семантика

30. Верно ли, что в языке C++ регистр символов важен при определении переменных?

**Верно**

Неверно

31. Верно ли, что на языке C++ написано большинство современных операционных систем?

**Верно**

Неверно

## Раздел 2. Основные конструкции языка программирования

32. Как обозначается оператор взятия остатка деления в языке C++:

Ответ: %

33. Как обозначается оператор деления в языке C++:

Ответ: /

34. Как обозначается оператор присваивания в языке C++:

Ответ: =

35. Как обозначается оператор присваивания языка C++, выполняющий вычитание:

Ответ: -=

36. Как обозначается оператор присваивания языка C++, выполняющий деление:

Ответ: /=

37. Как обозначается оператор присваивания языка C++, выполняющий умножение:

Ответ: \*=

38. Как обозначается оператор присваивания языка C++, выполняющий сложение:

Ответ: +=

39. Как обозначается оператор сдвига влево в языке C++:

Ответ: <<

40. Как обозначается оператор сдвига вправо в языке C++:

Ответ: >>

41. Какая функция в языке C++ определяет возведение в степень?

Ответ: pow

42. Какая функция в языке C++ определяет квадратный корень:

Ответ: sqrt

43. Какая функция в языке C++ определяет модуль числа (абсолютное значение)?

Ответ: abs

44. Какая функция в языке C++ определяет округление вещественного числа в большую сторону?

Ответ: floor

45. Какой заголовочный файл C++ содержит инструкции файлового ввода/вывода?

- 1) **fstream**
- 2) ifstream
- 3) ifstream
- 4) iostream

46. Какой из следующих классов обрабатывает процесс записи в файл?

- 1) **ofstream**
- 2) другое
- 3) ifstream
- 4) input\_file

47. Какой из следующих классов обрабатывает процесс чтения из файла?

- 1) ofstream
- 2) другое
- 3) **ifstream**

4) `input_file`

48. Как организовать запись в файл?

- 1) `a_file<< "запись";`
- 2) `a_file="запись";`
- 3) `a_file.printf("запись");`
- 4) `a_file.out("запись");`

49. Как организовать считывание из файла?

- 1) `a_file>> "считывание";`
- 2) `a_file=" считывание ";`
- 3) `a_file scanf("запись");`
- 4) `a_file.in("запись");`

50. Простые типы данных в C++.

- 1) целые - `int`, вещественные - `float` или `real`, символьные - `char`
- 2) целые - `int`, вещественные - `float` или `double`, символьные - `string`
- 3) целые - `bool`, вещественные - `float` или `double`, символьные - `string`
- 4) **целые - `int`, вещественные - `float` или `double`, символьные - `char`**

51. Поставьте в соответствие типам данных их описание

<code>long</code>	целый
<code>float</code>	вещественный
<code>bool</code>	логический
<code>char</code>	символьный

52. Поставьте в соответствие типам данных их описание

<code>int</code>	целый
<code>double</code>	вещественный
<code>bool</code>	логический
<code>char</code>	символьный

53. Укажите последовательность выполнения операторов в цикле `for`, если цикл обозначить следующим образом: `for(Оператор1; Оператор 2; Оператор 3) Блок_кода`

Оператор1	1
Оператор2	2
Оператор3	4
Блок_кода	3

54. В каком случае лучше всего использовать приведение типов данных?

- 1) чтобы изменить тип возвращаемого значения функции
- 2) чтобы разрешить программе использовать только целые числа
- 3) **при делении двух целых чисел, для того, чтобы вернуть результат с плавающей точкой**
- 4) во всех выше указанных случаях

55. Вывод данных в C++

- 1) `cout<< выражение1, выражение2, ...;`
- 2) `cout>> выражение1 << выражение2 <<endl<< ...;`
- 3) `cout>> выражение1, выражение2, ...;`
- 4) **`cout<< выражение1 << выражение2 ...;`**

56. Ввод данных в C++

- 1) `cin>> выражение1, выражение2, ...;`
- 2) **`cin>> выражение1>>выражение2...;`**

- 3) `cin >> выражение1 >> выражение2 >> endl >> ...;`
- 4) `cin << выражение1 << выражение2...;`

57. Выберите правильный вариант объявления константной переменной в C++, где `type` - тип данных в C++  
`variable` - имя переменной `value` - константное значение

- 1) `constvariable = value;`
- 2) **`const type variable = value;`**
- 3) `consttypevariable := value;`

58. Какое значение будет напечатано?

```

1 int main(int argc, char** argv)
2 {
3     int x = 0;
4     int y = 0;
5
6     if (x++ && y++)
7     {
8         y += 2;
9     }
10
11     std::cout << x + y << std::endl;
12
13     return 0;
14 }

```

- 1) 4
- 2) 2
- 3) **1**
- 4) 3

59. К вещественному типу данных относится:

- 1) **float**
- 2) `bool`
- 3) `long`
- 4) `char`

60. К вещественному типу данных относится:

- 1) `longint`
- 2) `int`
- 3) **double**
- 4) `bool`

61. Какое из следующих значений эквивалентно зарезервированному слову `true`?

- 1) 1
- 2) 0.1
- 3) -1
- 4) **Все варианты ответов**
- 5) 66

62. Какое ключевое слово указывает, что целая переменная не может принимать отрицательные значения?

- 1) `positive`
- 2) `long`
- 3) **unsigned**
- 4) нет такого зарезервированного слова

63. К логическому типу данных относится:

- 1) `long`
- 2) `float`
- 3) **bool**
- 4) `double`

64. К логическому типу данных относится:

- 1) `char`
- 2) **bool**
- 3) `int`
- 4) `longint`

65. К символьному типу данных относится:

- 1) `int`

- 2) float
- 3) **char**
- 4) longint

66. К символьному типу данных относится:

- 1) **char**
- 2) long
- 3) double
- 4) bool

67. К целому типу данных относятся. Выберите один или несколько ответов:

- 1) **int**
- 2) **long**
- 3) float
- 4) char

68. К целому типу данных относятся. Выберите один или несколько ответов:

- 1) char
- 2) double
- 3) **long int**
- 4) bool

69. Какие преобразования типов данных не возможны без потери данных?

- 1) char ->float
- 2) все перечисленные преобразования не возможны
- 3) int ->float
- 4) **float ->int**

70. Какие преобразования типов данных не возможны без потери данных?

- 1) int -> char
- 2) float -> double
- 3) **double -> float**
- 4) все перечисленные преобразования не возможны

71. Какие преобразования типов данных не возможны без потери данных?

- 1) double ->int
- 2) float ->int
- 3) double -> float
- 4) **все перечисленные преобразования не возможны**

72. Какие преобразования типов данных не возможны без потери данных?

- 1) int -> char
- 2) float -> double
- 3) int ->float
- 4) **нет ответа**

73. Какой заголовочный файл следует подключить, чтобы можно было пользоваться приведением типов данных?

- 1) cctype
- 2) **Никакого**
- 3) cmath
- 4) iostream

74. Какой из перечисленных типов данных не является типом данных в C++?

- 1) int
- 2) float
- 3) double
- 4) **const**

75. Объявлена переменная char a; Какое из следующих выражений не верно?

- 1) a = '3';
- 2) **a = "3";**
- 3) a = 3;
- 4) a = (3);

76. Оператор присваивания в C++:

- 1) **=**

- 2) :=
- 3) ===
- 4) ==

77. Оператор вывода cout может печатать несколько значений или переменных в одной команде, используя следующий синтаксис:

- 1) cout<< ("Привет" & name & "n");
- 2) cout<< "Привет", name, "n";
- 3) cout<< "Привет" + name + "n";
- 4) **cout<< "Привет" <<name<< "n";**

78. Оператор ввода cin может принимать несколько значений переменных в одной команде, используя следующий синтаксис:

- 1) cin>> "Введите var1" >>var1 >>"Введите var2" >>var2;
- 2) **cin>> var1 >> var2;**
- 3) cin>> "Введите var1" + var1 >>"Введите var2" + var2;
- 4) cin<< var1 << var2;

79. Почему приведение типов данных может быть не безопасно.

- 1) **Вы можете временно потерять часть данных - таких, как отсечение десятичной части чисел с плавающей точкой.**
- 2) Некоторые преобразования не определены компилятором, такие как - преобразование символа в целое.
- 3) Вы можете навсегда изменить значение переменной.
- 4) нет никаких опасностей.

80. Преобразование целочисленной переменной в вещественный тип данных:

- 1) float a = 56;
- 2) float a = float (56);
- 3) float a = static\_cast<float>56;
- 4) Все варианты ответов

81. Выберите вариант, где используется явное преобразование:

- 1) float a = 56;
- 2) float a = float (56);
- 3) **float a = static\_cast<float>56;**
- 4) 56/1.0;

82. Выберите вариант, где используется неявное преобразование:

- 1) float a = 56;
- 2) float a = float (56);
- 3) 56/1.0;
- 4) **Все варианты ответов**

83. Преобразование целочисленной переменной value в ASCII эквивалент

- 1) **(char) value**
- 2) cout<< value
- 3) atoi(value)
- 4) char (value)

84. Результат выполнения следующего фрагмента кода: 54 << 3?

- 1) 440
- 2) 556
- 3) 623
- 4) **432**

85. Результат выполнения следующего фрагмента кода: 168 >> 3?

- 1) 56
- 2) **21**
- 3) 504
- 4) 1344

86. Результат выполнения следующего фрагмента кода: cout<< 22 / 5 \* 3;

- 1) **12**
- 2) 1.47
- 3) 13.2
- 4) 13

87. Результат выполнения следующего фрагмента кода: `cout<< 22 * 3 / 5;`

- 1) 12
- 2) 1.47
- 3) 13.2
- 4) **13**

88. Укажите верные способы объявить переменную `a` вещественного типа данных, равную 0. Выберите один или несколько ответов:

- 1) **`float a=0;`**
- 2) **`float a(0);`**
- 3) `int a = 0;`
- 4) `int a(0);`

89. Укажите верные способы объявить переменную `a` целочисленного типа данных, равную 0. Выберите один или несколько ответов:

- 1) `float a=0;`
- 2) `float a(0);`
- 3) **`int a = 0;`**
- 4) **`int a(0);`**

90. Укажите операцию, приоритет выполнения которой больше остальных

- 1) `+`
- 2) **`()`**
- 3) `++`
- 4) `*`

91. Укажите операцию, приоритет выполнения которой больше остальных

- 1) `+`
- 2) **`++`**
- 3) `*`
- 4) `/`

92. Укажите правильное приведение типа данных

- 1) `char:a;`
- 2) `a(char);`
- 3) `to(char, a);`
- 4) **`(char)a;`**

93. Какое значение примет переменная `c` в результате выполнения следующей операции: `c = (a = 3, b = 4)`  
Ответ: 4

94. Какое значение примет переменная `c` в результате выполнения следующей операции: `c = (a = 3, b = 4, d = 18)`  
Ответ: 18

95. Какое количество байт занимает тип данных `bool` в 32х разрядной операционной системе  
Ответ: 1

96. Какое количество байт занимает тип данных `char` в 32х разрядной операционной системе  
Ответ: 1

97. Какое количество байт занимает тип данных `double` в 32х разрядной операционной системе  
Ответ: 8

98. Какое количество байт занимает тип данных `float` в 32х разрядной операционной системе  
Ответ: 4

99. Какое количество байт занимает тип данных `int` в 32х разрядной операционной системе  
Ответ: 4

100. Какое количество байт занимает тип данных `long` в 32х разрядной операционной системе  
Ответ: 4

101. Какое количество байт занимает тип данных `wchar_t` в 32х разрядной операционной системе  
Ответ: 2

102. Чему будет равно значение `c` в следующем выражении: `double c = 4 / 3;`  
1) **1**

- 2) 2
- 3) Возникнет ошибка компиляции
- 4) 1.333333

103. Какое значение вернет следующее выражение ( $a = 3, b = 4$ ):

Ответ: 4

104. Какое значение вернет следующее выражение ( $a = 3, b = 4, c = 5, d = 6$ ):

Ответ: 6

105. Какое значение вернет следующее выражение:  $a = b = c = 10$ ;

Ответ: 10

106. Чему в результате вычисления будет равно значение переменной  $a$ , если  $a = 1/4$

Ответ: 0

107. Чему в результате вычисления будет равно значение переменной  $a$ , если  $a = 3/4$

Ответ: 0

108. Чему в результате вычисления будет равно значение переменной  $a$ , если  $a = 9/4$

Ответ: 2

109. Чему равен результат выполнения следующего выражения:  $1000 / 100 \% 7 * 2$

- 1) 250
- 2) 1000
- 3) **6**
- 4) 10

110. Что будет напечатано, после выполнения этого кода: `cout<< (5 << 3);` ?

- 1) 53
- 2) **40**
- 3) 35
- 4) 15

111. Что будет напечатано, после выполнения этого кода: `cout<< (120 >> 3);` ?

- 5) 53
- 6) 40
- 7) 35
- 8) **15**

112. Это значение  $5.9875e17$  может быть сохранено в переменной, типа:

- 1) short
- 2) **float**
- 3) int
- 4) long

113. Это значение  $5.9875e39$  может быть сохранено в переменной, типа:

- 1) float
- 2) int
- 3) long
- 4) **double**

114. Операция декремента (уменьшение значения на единицу) в языке C++ записывается:

Ответ: --

115. Операция инкремента (увеличение значения на единицу) в языке C++ записывается:

Ответ: ++

116. Поток ввода в языке C++

Ответ: cin

117. Поток вывода в языке C++

Ответ: cout

118. Пустой тип данных в языке программирования C++ записывается как:

Ответ: void

### Раздел 3. Теория алгоритмов

119. В каком случае можно не использовать фигурные скобки в операторе выбора if?

- 1) если в теле оператора if нет ни одного оператора



- 2) нет правильного ответа
- 3) если в теле оператора if два и более операторов
- 4) **если в теле оператора if всего один оператор**

120. Выберите верный алгоритм поиска максимума

- 1) **int a[10];**  
**int max = a[0];**

```
for(int i = 1; i < 10; i++)
{
    if(a[i] > max )
    max = a[i];
}
```

- 2) int a[10];  
int max = a[0];

```
for(int i = 1; i < 10; i++)
{
    if(a[i] == max )
    max = a[i];
}
```

- 3) int a[10];  
int max = a[0];

```
for(int i = 1; i < 10; i++)
{
    if(a[i] < max )
    max = a[i];
}
```

121. Выберите верный алгоритм поиска минимума

- 1) int a[10];  
int min = a[0];

```
for(int i = 1; i < 10; i++)
{
    if(a[i] > min )
    max = a[i];
}
```

- 2) int a[10];  
int min = a[0];

```
for(int i = 1; i < 10; i++)
{
    if(a[i] == min )
    max = a[i];
}
```

- 3) **int a[10];**  
**int max = a[0];**

```
for(int i = 1; i < 10; i++)
{
    if(a[i] < min )
    max = a[i];
}
```

122. Если условие оператора выбора if ложное, то:

- 1) **выполняется следующий оператор, сразу после оператора if**
- 2) программа завершает работу

3) выполняется тело оператора выбора if

123. Какое значение вернет следующее выражение (true или false):

`!(2<3) || (3>4) || !(4<7)`

Ответ: false

124. Какое значение вернет следующее выражение (true или false):

`!(2<3)||((3>4))`

Ответ: false

125. Какое значение вернет следующее выражение (true или false):

`!(2<3)&&(3>4)`

Ответ: true

126. Какое значение вернет следующее выражение (true или false):

`((2<3)||((3>4)&&(4<7)))`

Ответ: true

127. Какое значение вернет следующее выражение (true или false):

`((2<3)&&!(3>4)&&(4<7))`

Ответ: true

128. Какое значение вернет следующее выражение (true или false):

`((2<3)||((3>4)))`

Ответ: true

129. Какое значение вернет следующее выражение (true или false):

`((2<3)&&(3>4))`

Ответ: false

130. Какое значение вернет следующее выражение (true или false):

`((2<3),(3>4))`

Ответ: false

131. Какое значение вернет следующее выражение (true или false):

`((2<3)||!(3>4))`

Ответ: true

132. Какое значение вернет следующее выражение (true или false):

`((2<3)&&!(3>4))`

Ответ: true

133. Каков будет результат выражения `!(1 && !(0 || 1))?`

- 1) false
- 2) неоднозначность
- 3) **true**

134. Результат выполнения следующего фрагмента кода: `!((1 || 0) && 0)`

- 1) **1**
- 2) результат не может быть заранее определен
- 3) 0
- 4) Другое

135. Результат выполнения следующего фрагмента кода: `!((1 && 0) && 0)`

- 1) **1**
- 2) результат не может быть заранее определен
- 3) 0
- 4) Другое

136. Результат выполнения следующего фрагмента кода: `!((1 || 0) || 0)`

- 1) 1
- 2) результат не может быть заранее определен
- 3) **0**
- 4) Другое

137. Какие скобки определяют условия

- 1) Квадратные
- 2) Фигурные
- 3) **Круглые**
- 4) Угловые

138. Какой из ниже перечисленных вариантов ответа, показывает правильно записанный оператор выбора if
- 1) if{ условное выражение}
  - 2) **if( условное выражение )**
  - 3) условное выражение if
  - 4) if условное выражение
139. Какой из ниже перечисленных операторов, не является циклом в C++?
- 1) dowhile
  - 2) **switch**
  - 3) while
  - 4) for
140. Какой из следующих логических операторов - логический оператор И?
- 1) ||
  - 2) **&&**
  - 3) |
  - 4) &
141. Какой из следующих логических операторов - логический оператор ИЛИ?
- 1) ||
  - 2) &&
  - 3) |
  - 4) &
142. Какой из следующих логических операторов - логический оператор НЕ?
- 1) -
  - 2) **!**
  - 3) !!
  - 4) no
143. Какой из следующих операторов - оператор сравнения двух переменных?
- 1) :=
  - 2) ==
  - 3) =
  - 4) ===
144. Ключевое слово, обозначающее начало оператора цикла с предусловием:  
Ответ: while
145. Ключевое слово, обозначающее начало условного оператора:  
Ответ: if
146. Ключевое слово, обозначающее начало цикла с постусловием:  
Ответ: do while
147. Ключевое слово, обозначающее начало цикла с параметром:  
Ответ: for
148. Логический оператор И в языке C++  
Ответ: &&
149. Логический оператор ИЛИ в языке C++  
Ответ: ||
150. Логический оператор НЕ в языке C++  
Ответ: !
151. Логическое значение ИСТИНА в языке C++ обозначается как  
Ответ: true
152. Логическое значение ЛОЖЬ в языке C++ обозначается как  
Ответ: false
153. Оператор сравнения НЕРАВНО в языке C++  
Ответ: !=
154. Оператор сравнения РАВНО в языке C++  
Ответ: ==
155. Какой оператор не допускает перехода от одного константного выражения к другому?
- 1) stop;
  - 2) **break;**

- 3) end;  
4) точка с запятой
156. Логическая операция с большим приоритетом выполнения.  
1) !  
2) &&  
3) ||  
4) \*
157. Оператор ifelse позволяет определить действие ...  
1) только для ложного условия  
2) только для истинного условия  
3) **для истинного и ложного условий**  
4) другое
158. Тело любого цикла выполняется до тех пор, пока его условие ...  
1) ложно  
2) у цикла нет условия  
3) **истинно**  
4) в любом случае
159. Тело оператора if, будет выполняться, если его условие:  
1) ложно (false)  
2) **истинно (true)**
160. Укажите блок кода, в котором переменная у доступна

```

1 int main(int argc, char** argv)
2 {
3
4     if ( argc > 10 )
5     {
6     }
7     else if (int y = argc - 1 )
8     {
9
10    }
11    else
12    {
13
14    }
15
16    return 0;
17 }

```

- 1) строки 8 -11  
2) строки 4 -17  
3) строки 4 -15  
4) **строки 8 -14**
161. Укажите верно написанный оператор if:  
1) **if (условие) {Оператор;}**  
2) if (условие) {Оператор}  
3) if условие {Оператор;}  
4) if (условие); {Оператор}  
5) if (условие) {Оператор};
162. Укажите верно написанный оператор while:  
1) **while (условие) {Оператор;}**  
2) while (условие) {Оператор}  
3) while условие {Оператор;}  
4) while (условие); {Оператор}  
5) while (условие) {Оператор};

163. Укажите верный условный оператор, выводящий на экран true или false в зависимости от значения логического выражения.

- 1) if a<b  
cout<< "true";  
else  
cout<< "false";
- 2) **if (a<b)**  
**cout<< "true";**  
**else**  
**cout<< "false";**
- 3) if (a<b)  
cout<< "true"  
else  
cout<< "false";
- 4) if (a<b);  
cout<< "true";  
else  
cout<< "false";

164. Укажите неправильно записанную операцию отношения

- 1) <=
- 2) >=
- 3) все операторы записаны правильно
- 4) !=

165. Укажите правильную форму записи цикла dowhile

- 1) 

```
1 // форма записи оператора цикла do while:
2 do // начало цикла do while
3 {
4 /*блок операторов*/;
```
- 2) 

```
1 // форма записи оператора цикла do while:
2 do // начало цикла do while
3 {
```
- 3) 

```
1 // форма записи оператора цикла do while:
2 do // начало цикла do while
3 {
4 /*блок операторов*/;
5 }
6 while (/*условие выполнения цикла*/); // конец цикла do while
```

- 3) **for**
- 4) switch

167. Цикл с постусловием, это?

- 1) **do while**
- 2) while
- 3) for
- 4) switch
- 5)

168. Цикл с предусловием, это?

- 1) do while
- 2) **while**
- 3) for
- 4) switch

169. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?

- 1) other
- 2) **default**
- 3) contingency
- 4) all

166. Цикл с параметром, это?

- 1) do while
- 2) while

170. Что появится на экране, после выполнения этого фрагмента кода:

```
1 int a = 1, b = 2;
2 if (a == b);
3 cout << a << " = " << b << endl;
```

- 1) a = b  
2) синтаксическая ошибка

- 3) вывод на экран не выполнится  
4) **1 = 2**

171. Оператор множественного выбора, это?

- 1) multiple  
2) case  
3) if  
4) **switch**

172. Что будет напечатано, после выполнения следующего фрагмента кода:

```
1 int main()
2 {
3     for (int i = 0; i < 4; ++i)
4     {
5         switch (i)
6         {
7             case 0 : std::cout << "0";
8             case 1 : std::cout << "1"; continue;
9             case 2 : std::cout << "2"; break;
10            default : std::cout << "D"; break;
11        }
12        std::cout << ".";
13    }
14    return 0;
15 }
```

- 1) Ошибка компиляции в строке 10  
2) 01.2.D.  
3) 0.1.2.  
4) **0112.D.**  
5) 011.2.D

173. Каков результат работы следующего фрагмента кода?

```
1 int x = 0;
2
3 switch(x)
4 {
5
6     case 1: cout << "Один";
7
8     case 0: cout << "Ноль";
9
10    case 2: cout << "Привет мир";
11
12 }
```

- 1) Привет мир  
2) Один  
3) Ноль  
4) **НольПривет мир**

174. Какое значение будет напечатано, в результате выполнения программы?

```

1  #include <iostream>
2
3  int main()
4  {
5      int x = 3;
6
7      switch(x)
8      {
9          case 0:
10             int x = 1;
11             std::cout << x << std::endl;
12             break;
13          case 3:
14             std::cout << x << std::endl;
15             break;
16          default:
17             x = 2;
18             std::cout << x << std::endl;
19         }
20
21     return 0;
22 }

```

- 1) 1
- 2) 2
- 3) ничего не напечатается, программа вообще не будет работать
- 4) 3
- 5) 0

175. В каком фрагменте кода будут напечатаны 20 звездочек - \*?

- 1) `inti, N = 20;`  
`for(i = 20; i < N; i--)`  
`cout << "*";`
- 2) `inti, N = 20;`  
`for(i = 19; i < N; i--)`  
`cout << "*";`
- 3) **`inti, N = 20;`**  
**`for(i = 0; i < N; N--)`**  
**`cout << "*";`**
- 4) `inti, N = 40;`  
`for(i = 0; i < N; i--)`  
`cout << "*";`

176. Какой служебный знак ставится после оператора case ?

- 1) -
- 2) .
- 3) :
- 4) ;

177. Сколько раз выполнится блок кода в цикле for? (бесконечность обозначать inf)

`for(i = 0; i <= 4; i++) Блок_кода`

Ответ: 5

178. Сколько раз выполнится блок кода в цикле for? (бесконечность обозначать inf)

`for(i = 0; i <= 1024; i++) Блок_кода`

Ответ: 1025

179. Сколько раз выполнится блок кода в цикле for? (бесконечность обозначать inf)

`for(i = 0; i < 1024; i++) Блок_кода`

Ответ: 1024

180. Сколько раз выполнится блок кода в цикле for? (бесконечность обозначать inf)

`for(i = 0; i < 4; i++) Блок_кода`

Ответ: 4

181. Сколько раз выполнится блок кода в цикле for? (бесконечность обозначать inf)

`for(i = 0; i = 4; i++) Блок_кода`

Ответ: 0

182. Сколько раз выполнится блок кода в цикле for? (бесконечность обозначать inf)  
for(i = 0; i == 1024; i++) Блок\_кода

Ответ: 0

183. Сколько раз выполнится блок кода в цикле for? (бесконечность обозначать inf)  
for(i = 0; i <= 4; i--) Блок\_кода;

Ответ: inf

184. Сколько раз выполнится блок кода в цикле for? (бесконечность обозначать inf)  
for(i = 0; i < 4; i--) Блок\_кода;

Ответ: inf

185. Вид алгоритма, в котором в зависимости от условия операторы языка выполняются по нескольким независимым сценариям, называется:

Ответ: разветвляющийся

186. Вид алгоритма, в котором в зависимости от условия операторы языка выполняются несколько раз, называется:

Ответ: циклический

187. Вид алгоритма, в котором операторы языка выполняются последовательно сверху вниз без ответвлений и циклов, называется:

Ответ: линейный

188. Чему будет равна переменная a, после выполнения этого кода: int a; for(a = 0; a < 10; a++) {}

- 1) 10
- 2) 1
- 3) **9**
- 4) 0

189. Переменная x может быть доступна в другом блоке программы?

```

1  int main(int argc, char** argv)
2  {
3      if ( argc > 2 )
4      {
5          int x = 5;
6      }
7      else
8      {
9      }
10     }
11
12     return 0;
13 }
```

Верно

**Неверно**

190. Может ли переменная x быть доступна в блоке else оператора if?

**Верно**

Неверно

#### Раздел 4. Структуры данных

191. В каком из следующих вариантов ответов выполнен корректный доступ к полю var структуры b? (Структура объявлена статически)

- 1) var.b
- 2) b-var;
- 3) var-b;
- 4) **b.var;**

192. В каком из следующих вариантов ответов выполнен корректный доступ к полю var структуры b? (Структура объявлена динамически)

- 1) b.var;
- 2) var->b
- 3) **b->var;**
- 4) b>var;



193. Правильное объявление переменной, типа структуры foo
- 1) foo;
  - 2) struct foo;
  - 3) int foo;
  - 4) **foovar;**
194. Правильное определение структуры в C++
- 1) structa\_structint a;
  - 2) **structa\_struct {int a};**
  - 3) structa\_struct {int a;}
  - 4) struct {int a;}
195. При определении структуры необходимо использовать следующее ключевое слово
- 1) record
  - 2) **struct**
  - 3) object
  - 4) structure
196. Укажите правильный доступ к переменной структуры
- 1) **b.var;**
  - 2) b-var;
  - 3) b->var;
  - 4) b>var;

197. Будет ли напечатано сообщение не равны?

```

1 struct Foo
2 {
3 };
4
5 struct Bar
6 {
7 };
8
9 int main(int argc, char** argv)
10 {
11     Foo* f = new Foo;
12     Bar* b = new Bar;
13
14     if ( f == b )
15         std::cout << "равны" << std::endl;
16     else
17         std::cout << "не равны" << std::endl;
18
19     return 0;
20 }

```

Верно

Неверно

## Раздел 5. Сортировка информации

198. Массив - это ...
- 1) Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и адресу
  - 2) **Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и индексу**
  - 3) Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие общий адрес. Доступ к отдельным элементам массива осуществляется по адресу и индексу
199. В какой из следующих строк выполняется обращение к седьмому элементу массива?
- 1) mas+7;
  - 2) mas(7);
  - 3) **mas[6];**
  - 4) mas[7];
  - 5) mas(6);
200. В каком из вариантов ответов объявлен двумерный массив целых чисел?
- 1) chararray[20];
  - 2) arrayanarray[20][20];
  - 3) intarray[20, 20];
  - 4) **intanarray[20][20];**

201. Выберите верный цикл for, который выводит на экран числа в диапазоне от 0 до 9

- 1) `for(int i = 0; i < 9; i++)`  
`{`  
`cout << i << endl;`  
`}`
- 2) **`for(int i = 0; i < 10; i++)`**  
`{`  
`cout << i << endl;`  
`}`
- 3) `for(int i = 1; i < 10; i++);`  
`{`  
`cout << i << endl;`  
`}`
- 4) `for(int i = 1; i < 9; i++);`  
`{`  
`cout << i << endl;`  
`}`

202. До каких пор будут выполняться операторы в теле цикла while (x < 100)?

- 1) Пока x больше ста
- 2) **Пока x строго меньше ста**
- 3) Пока x равен ста
- 4) Пока x меньше или равен ста

203. Какое значение будет напечатано, в результате выполнения следующего кода?

```

1 #include <iostream>
2
3 int main(void)
4 {
5     int sum = 0;
6
7     int array[3][3] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}};
8
9     for (int i = 0; i < 3; ++i)
10    {
11        for (int j = 2; j < 3; j++)
12        {
13            sum += array[i][j];
14        }
15    }
16
17    std::cout << sum << std::endl;
18
19    return 0;
20 }

```

- 1) **15**
- 2) синтаксическая ошибка
- 3) 21
- 4) 9

204. Какие служебные символы используются для работы с индексом?

- 1) ( )
- 2) [ ]
- 3) <
- 4) { }

205. Какой порядковый номер последнего элемента массива, размер массива 19?

- 1) 19
- 2) порядковый номер определяется программистом
- 3) **18**
- 4) 20

206. Какие скобки используются для работы с массивом
- 1) **Квадратные**
  - 2) Фигурные
  - 3) Круглые
  - 4) Угловые
207. Укажите правильное объявление массива
- 1) `arrayarray[10];`
  - 2) `array{10};`
  - 3) **`intanarray[10];`**
  - 4) `intanarray;`
208. Укажите строку, которая возвращает адрес первого элемента в массиве `arr`:
- 1) **`arr[0];`**
  - 2) `arr[1];`
  - 3) `arr;`
  - 4) `&arr;`

### Раздел 6. Поиск информации

209. Допустим у нас есть массив `char arr[8];` И в массив `arr` мы попытались с помощью оператора `cin` записать следующий набор символов `HelloWorld`. Что в действительности будет содержать массив `arr`?
- 1) **`HelloWo`**
  - 2) `Hello W`
  - 3) `HelloWorld`
  - 4) `Hello`
210. Каким символом завершается Си-строка?
- 1) `"`
  - 2) `\n`
  - 3) `'`
  - 4) `;`
211. Словосочетание `"HelloWorld!"` может быть сохранено в символьном массиве размером `n` элементов. Укажите чему равно `n`?
- 1) 10
  - 2) 11
  - 3) 12
  - 4) **13**
212. Строковый типы данных в C++
- 1) строки в C++ представляются как массивы элементов типа `char`, заканчивающиеся терминатором строки - символом с нулевым значением `'0'`.
  - 2) строки в C++ представляются как массивы элементов типа `char`, заканчивающиеся терминатором строки - символом с нулевым значением `"`.
  - 3) **строки в C++ представляются как массивы элементов типа `char`, заканчивающиеся терминатором строки - символом с нулевым значением `'\0'`.**
213. Укажите корректное определение строковой переменной. Выберите один или несколько ответов:
- 1) `string[20] mystr;`
  - 2) `string mystr[20];`
  - 3) **`string mystr;`**
  - 4) **`char mystr[20];`**
214. Укажите литерал (константу) строки в языке C++
- 1) **`"Статическая строка"`**
  - 2) `'Статическая строка'`
  - 3) `charstring[100];`
  - 4) `constcharstring;`
215. Какая из следующих функций считывает 100 символов из входного потока в строку `x`?
- 1) `read(x);`
  - 2) `cin.getline(100, x, 'n');`
  - 3) `getline(x, 100, 'n');`
  - 4) **`cin.getline(x, 100, 'n');`**

216. Какая из следующих функций добавляет одну строку в конец другой?
- 1) **strcat** ();
  - 2) strcpy();
  - 3) strncpy();
  - 4) strlen ();
217. Какая из следующих функций копирует полностью одну строку в другую?
- 1) strcat ();
  - 2) **strcpy**();
  - 3) strncpy();
  - 4) strlen ();
218. Какая из следующих функций копирует фиксированное количество символов из одной строки в другую?
- 1) strcat ();
  - 2) strcpy();
  - 3) **strncpy**();
  - 4) strlen ();
219. Какая из следующих функций определяет размер строки?
- 1) strcat ();
  - 2) strcpy();
  - 3) strncpy();
  - 4) **strlen** ();
220. Какая из следующих функций ищет символ в строке?
- 1) strcmp ();
  - 2) strstr();
  - 3) **strchr** ();
  - 4) strlen ();
221. Какая из следующих функций сравнивает две строки?
- 1) compare();
  - 2) stringcompare();
  - 3) **strcmp**();
  - 4) cmp();
222. Каков результат работы следующей программы?
- ```
char s[] = "Hello\0Hi";
cout << strlen(s) << " " << sizeof(s);
```
- 1) **5 9**
  - 2) 7 20
  - 3) 5 20
  - 4) 8 20

### Раздел 7. Объектно-ориентированное программирование (ООП).

223. Выберите верный способ объявить функцию в языке C++:
- 1) int имя\_функции, параметры;
  - 2) параметры имя\_функции (тип\_результата);
  - 3) **тип\_результата имя\_функции(параметры);**
  - 4) имя\_функции параметры (тип\_результата);
224. Правильное определение функции в C++
- 1) **void name { }**
  - 2) funct name { }
  - 3) void name { };
  - 4) funct name { };
225. В какой из следующих записей используется операция взятия адреса?
- 1) \*a;
  - 2) **&a;**
  - 3) address(a);

4) a&;

226. Выберите правильное (полное) определение функции

1) 

```
1 int funct(int x)
2 {
3     return x = x + 1;
4 }
```

```
1 int funct();
```

2)

3) 

```
1 void funct(int)
2 {
3     cout << "Hello"
4 }
```

227. Укажите правильный вызов функции. Предполагается, что функция была объявлена ранее.

- 1) **funct();**
- 2) funct;
- 3) intfunct();
- 4) funct x, y;

228. Указанный код объявляет массив ссылок: int&x[50];

**Верно**

Неверно

229. Поименованный набор описаний и операторов, выполняющих определенную задачу:

Ответ: функция

230. Можно ли гарантировать, что объявленная встроенная функция действительно является встроенной?

Верно

**Неверно**

231. Можно ли перегрузить функцию main()?

Верно

**Неверно**

232. Что будет напечатано на экране, после выполнения этого кода?

```
1 #include <iostream>
2
3 using namespace std;
4
5 int foo(int y);
6 int foo(int x)
7 {
8     return x+1;
9 }
10
11 int main(int argc, char** argv)
12 {
13     int x = 3;
14     int y = 6;
15
16     cout << foo(x) << endl;
17
18     return 0;
19 }
```

- 1) 3
- 2) ошибка компиляции
- 3) **4**
- 4) 9

233. Укажите тип возвращаемого значения следующей функции:  
intfunc(char x, float v, double t);

1) **int**

- 2) float
- 3) double
- 4) char
- 5) Функция не возвращаемого типа

234. Укажите тип возвращаемого значения следующей функции: voidfunc(char x, float v, double t);

- 1) void
- 2) float
- 3) double
- 4) char
- 5) **Функция не возвращаемого типа**

235. Для чего используются встроенные функции?

- 1) Для удаления ненужных функций
- 2) Чтобы уменьшить размер программы
- 3) Для упрощения файла с исходным кодом

#### 4) Для увеличения скорости работы программы

236. Какие из следующих функций являются встроенными?

- 1) **inlinevoidfoo() {}**
- 2) нет правильного ответа
- 3) inline:void foo() {}
- 4) void foo() inline {}

237. Что значит ключевое слово inline?

- 1) препроцессор хранит в памяти код функции
- 2) **все вызовы встроенных функции заменяются кодом этой функции**
- 3) сообщает компилятору использовать функцию только в пределах одного файла с исходным кодом
- 4) нет правильного ответа

238. Какая из следующих записей возвращает значение переменной a, хранящейся в памяти по адресу на который указывает указатель?

- 1) &a;
- 2) **\*a;**
- 3) val(a);
- 4) a

239. Какие из следующих функций являются встроенными?

- 1) **inlinevoidfoo() {}**
- 2) нет правильного ответа
- 3) voidfoo() inline {}
- 4) inline:voidfoo() {}

240. Какие скобки определяют получение параметров в функциях

- 1) **Круглые**
- 2) Квадратные
- 3) Фигурные
- 4) Угловые

241. Корректное выделение памяти

- 1) int \*a = new 20;
- 2) int a = new sizeof(int \* 20);
- 3) **int \*a = new int[20];**
- 4) int a = new int(20);

242. Корректное выделение памяти

- 1) int a = new int[20];
- 2) int \*a = new int(20);
- 3) int \*a = new sizeof(int \* 20);
- 4) **int \*a = new int;**

243. Какие скобки используются для работы с индексом и для выделения памяти

- 1) **Квадратные**
- 2) Фигурные
- 3) Круглые
- 4) Угловые

244. После выполнения ряда операций с указателем, что будет выведено на экран?

```

1 int main(int argc, char** argv)
2 {
3     // предположим, int занимает 4 байта
4     cout << sizeof(int) << endl;
5
6     int *x = new int;
7
8     // предположим адрес равен 0x60450000
9     cout << x << endl;
10
11     cout << x + 3 << endl;
12
13     return 0;
14 }

```

- 1) некорректное определение
- 2) нельзя заранее сказать, каково будет значение адреса
- 3) **0x60450003**
- 4) 0x60450000
- 5) 0x6045000C

245. Укажите правильное объявление указателя в C++

- 1) ptr x;
- 2) int&x;
- 3) **int \*x;**
- 4) int x;

246. Укажите зарезервированное ключевое слово для динамического выделения памяти?

- 1) value
- 2) **new**
- 3) malloc
- 4) create

247. Укажите зарезервированное ключевое слово для высвобождения выделенной памяти!

- 1) clear
- 2) remove
- 3) free
- 4) **delete**

248. Строка Привет Мир будет показана на экране или нет?

```

1  int main(int argc, char** argv)
2  {
3      int array[33];
4
5      if ( &array[4] < &array[23] )
6      {
7          std::cout << "Привет мир" << std::endl;
8      }
9
10     return 0;
11 }

```

- 1) синтаксическая ошибка
- 2) нет
- 3) **да**

249. Что из нижеперечисленного не является прототипом функции?

- 1) void funct();
- 2) intfunct(char x, char y);
- 3) **double funct(char x)**
- 4) char x();

250. Что такое ссылка?

- 1) **ссылка является псевдонимом для имени переменной**
- 2) оператор
- 3) используется для переименования объектов
- 4) нет правильного ответа

**Время выполнения: 40 минут**

**Перечень объектов контроля и оценки**

| Наименование объектов контроля и оценки                                                                                                                        | Основные показатели оценки результата | Оценка (кол-во баллов) |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|------------------------|
| 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. Использовать информационно-коммуникационные технологии в профессиональной деятельности.                                                                  |                                       |                        |
| 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.                                                   | Электронный тест                      | 20 баллов              |
| 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.                                                                      |                                       |                        |
| ПК 1.1. Выполнять разработку спецификаций отдельных компонент.                                                                                                 |                                       |                        |
| ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых                                                                                   |                                       |                        |
| 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 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.                                        |                                       |                        |



#### 4. Шкала оценки образовательных достижений

| Процент результативности (правильных ответов) | Качественная оценка индивидуальных образовательных достижений |                     |
|-----------------------------------------------|---------------------------------------------------------------|---------------------|
|                                               | балл (отметка)                                                | вербальный аналог   |
| 91 - 100                                      | 5                                                             | отлично             |
| 76 -90                                        | 4                                                             | хорошо              |
| 61 -75                                        | 3                                                             | удовлетворительно   |
| менее 61                                      | 2                                                             | неудовлетворительно |
| более 61                                      | зачтено                                                       |                     |
| менее 61                                      | не зачтено                                                    |                     |

#### 5. Перечень используемых материалов, оборудования и информационных источников

##### Основная литература:

4. Баженова, И.Ю. Введение в программирование : курс лекций / Баженова И.Ю., Сухомлин В.А. — Москва : Интуит НОУ, 2016. — 411 с. — ISBN 978-5-9556-0077-9. — URL: <https://book.ru/book/917645>

5. Основы программирования : курс лекций / Мейер Бертран — Москва : Интуит НОУ, 2016. — 324 с. — ISBN 978-5-9963-0573-5. — URL: <https://book.ru/book/917903>

6. 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>

##### 3.2.2 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>

##### 3.3 Дополнительная литература:

3. 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>

4. 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>

#### Оборудование:

| № пп | Материально-техническое обеспечение лаборатории и практикума по дисциплине |
|------|----------------------------------------------------------------------------|
| 1.   | Компьютерное и программное обеспечение:                                    |
| 1.1. | Компьютеры – 20 шт                                                         |
| 1.2. | Мультимедийное оборудование -1 шт                                          |
| 1.3. | Операционная система Windows'XP и выше                                     |
| 1.4. | Пакет Microsoft Office 2007 и выше.                                        |
| 1.5. | Интерактивная доска                                                        |