МИНОБРНАУКИ РОССИИ

Филиал федерального государственного бюджетного образовательного учреждения высшего образования
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
в г. Артеме
(ФИЛИАЛ ФГБОУ ВО «ВВГУ» В Г. АРТЕМЕ)

УТВЕРЖДАЮ
Зам. директора филиала
ФГБОУ ВО «ВВГУ» в г. Артеме

_____ В.В. Неслюзов

РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

ПМ.01 Осуществление интеграции программных модулей

программы подготовки специалистов среднего звена
09.02.07 Информационные системы и программирование

Форма обучения: *очная*

Артем 2022 г.

Рабочая программа профессионального модуля ПМ.01 «Осуществление интеграции программных модулей» разработана на основе Федерального государственного образовательного стандарта для специальности среднего профессионального образования 09.02.07 «Информационные системы и программирование»

Разработчик(и): *Волошин Е.В., Ематина Н.И., преподаватели*

Утверждена на заседании цикловой методической комиссии по профилю математических и информационных дисциплин, протокол № 1 от 01.10.2022 г

Председатель ЦМК_____*А.С.Бажина*

подпись

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	**4**
2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	**6**
3. УСЛОВИЯ И РЕАЛИЗАЦИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	**12**
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	**14**

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРИМЕРНОЙ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

ПМ.01. Осуществление интеграции программных модулей

1.1 Место дисциплины в структуре основной образовательной программы

Рабочая программа профессионального модуля (далее программа ПМ) является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности СПО 09.02.07 Информационные системы и программирование.

Рабочая программа составляется для очной формы обучения.

В результате изучения профессионального модуля студент должен освоить основной вид деятельности ПМ.01 Осуществление интеграции программных модулей и соответствующие ему общие компетенции, и профессиональные компетенции:

Код	Наименование общих компетенций
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 02	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
ОК 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 09	Использовать информационные технологии в профессиональной деятельности.
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языке
ОК 11	Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере

Перечень общих компетенций

Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ПК 2.1.	Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент
ПК 2.2.	Выполнять интеграцию модулей в программное обеспечение
ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5.	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

В результате освоения профессионального модуля студент должен:

иметь практический опыт	в разработке программного обеспечения; в интегрировании программных модулей; в верификации и аттестации программного обеспечения
уметь	использовать выбранную систему контроля версий; использовать методы для получения кода с заданной функциональностью и степенью качества
знать	модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения

1.3. Количество часов, отводимое на освоение профессионального модуля

Вид учебной деятельности	Объем часов
Максимальная учебная нагрузка (всего)	590
Обязательная аудиторная учебная нагрузка (всего)	322
Учебная практика	110
Производственная практика	146
Экзамен по модулю	12
Итоговая аттестация в форме: МДК.01.01 – контрольная работа 1 семестр, дифференцированный зачет 2 семестр МДК.01.02 – контрольная работа 1 семестр, дифференцированный зачет 2 семестр МДК.01.03 – дифференцированный зачет 2 семестр УП.01.01 – дифференцированный зачет 2 семестр ПП.01.01 – дифференцированный й зачет 2 семестр ПМ.01.ЭК квалификационный экзамен – 2 семестр	

2. СТРУКТУРА и содержание профессионального модуля

2.1. Структура профессионального модуля «ПМ.01. Осуществление интеграции программных модулей»

Коды профессиональных общих компетенций	Наименования разделов профессионального модуля	Суммарный объем нагрузки, час.	Объем времени, отведенный на освоение междисциплинарного курса					
			Обязательная аудиторная учебная нагрузка обучающегося				Самостоятельная работа обучающегося	
			Всего	в т.ч. лабораторные работы и практические занятия, часов	в т.ч. теоретическое обучение, часов	в т.ч. курсовая работа, часов	Всего, часов	в т.ч. консультации
ОК 01 – ОК 11 ПК 2.1 – ПК 2.5	МДК 01.01. Технология разработки программного обеспечения	136	136	83	49		4	
	МДК 01.02. Инструментальные средства разработки программного обеспечения	120	120	83	33		4	
	МДК.01.03 Математическое моделирование	66	66	32	32		2	
	УП.01.01 Учебная практика	110						2
	ПП.01.01 Производственная практика	146						2
	Экзамен по модулю	12						
	Всего:	**590**						

2.2. Тематический план и содержание профессионального модуля (ПМ.01 Осуществление интеграции программных модулей)

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, курсовая работ (проект) (если предусмотрены)	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Раздел 1		**136**	
МДК.01.01 Технология разработки программного обеспечения			
Тема 1.1. Основные понятия и стандартизация требований к программному обеспечению	**Содержание учебного материала:**		ОК 01 – ОК 11 ПК 2.1 – ПК 2.5
	Понятия требований, классификация, уровни требований. Методологии и стандарты, регламентирующие работу с требованиями. Современные принципы и методы разработки программных приложений. Методы организации работы в команде разработчиков. Системы контроля версий. Основные подходы к интегрированию программных модулей. Стандарты кодирования.	10	
	Практические работы:	20	
	«Анализ предметной области» «Разработка и оформление технического задания» «Построение архитектуры программного средства» «Изучение работы в системе контроля версий»		
Тема 1.2. Описание и анализ требований. Диаграммы IDEF	**Содержание учебного материала:**	20	ОК 01 – ОК 11 ПК 2.1 – ПК 2.5
	Описание требований: унифицированный язык моделирования - краткий словарь. Диаграммы UML. Описание и оформление требований (спецификация). Анализ требований и стратегии выбора решения		
	Практические работы:	40	
	«Построение диаграммы Вариантов использования и диаграммы. Последовательности» «Построение диаграммы Кооперации и диаграммы Развертывания» «Построение диаграммы Деятельности, диаграммы Состояний и диаграммы Классов» «Построение диаграммы компонентов» «Построение диаграмм потоков данных»		
Тема1.3. Оценка качества программных	**Содержание учебного материала:**	19	ОК 01 – ОК 11 ПК 2.1 – ПК 2.5
	Цели и задачи, и виды тестирования. Стандарты качества программной документации. Меры и метрики. Тестовое покрытие. Тестовый сценарий, тестовый пакет. Ана-		

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, курсовая работ (проект) (если предусмотрены)	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
средств	лиз спецификаций. Верификация и аттестация программного обеспечения.		
	Практические работы:		
	«Разработка тестового сценария» «Оценка необходимого количества тестов» «Разработка тестовых пакетов» «Оценка программных средств с помощью метрик» «Инспекция программного кода на предмет соответствия стандартам кодирования»	23	
Самостоятельная работа		**4**	
Раздел 2.		**120**	
МДК.01.02 Инструментальные средства разработки программного обеспечения			
Тема 2.1. Современные технологии и инструменты интеграции.	**Содержание учебного материала:**	16	ОК 01 – ОК 11 ПК 2.1 – ПК 2.5
	Понятие репозитория проекта, структура проекта. Виды, цели и уровни интеграции программных модулей. Автоматизация бизнес-процессов. Выбор источников и приемников данных, сопоставление объектов данных. Транспортные протоколы. Стандарты форматирования сообщений. Организация работы команды в системе контроля версий.		
	Практические работы:	40	
	«Разработка структуры проекта» «Разработка модульной структуры проекта (диаграммы модулей)» «Разработка перечня артефактов и протоколов проекта» «Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)» «Разработка и интеграция модулей проекта (командная работа)» «Отладка отдельных модулей программного проекта» «Организация обработки исключений»		
Тема 2.2. Инструментарий тестирования и анализа качества программных средств	**Содержание учебного материала:**	17	ОК 01 – ОК 11 ПК 2.1 – ПК 2.5
	Отладка программных продуктов. Инструменты отладки. Отладочные классы. Ручное и автоматизированное тестирование. Методы и средства организации тестирования. Инструментарии анализа качества программных продуктов в среде разработке. Обработка исключительных ситуаций. Методы и способы идентификации сбоев и ошибок. Выявление ошибок системных компонентов.		

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, курсовая работ (проект) (если предусмотрены)	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
	Практические работы:	43	
	«Применение отладочных классов в проекте» «Отладка проекта» «Инспекция кода модулей проекта» «Тестирование интерфейса пользователя средствами инструментальной среды разработки» «Разработка тестовых модулей проекта для тестирования отдельных модулей» «Выполнение функционального тестирования» «Тестирование интеграции» «Документирование результатов тестирования»		
Самостоятельная работа		**4**	
Раздел 3.		**66**	
МДК.01.03 Математическое моделирование			
Тема 3.1. Основы моделирования. Детерминированные задачи	**Содержание учебного материала:**	16	ОК 01 – ОК 11 ПК 2.1 – ПК 2.5
	Понятие решения. Множество решений, оптимальное решение. Показатель эффективности решения. Математические модели, принципы их построения, виды моделей. Задачи: классификация, методы решения, граничные условия. Общий вид и основная задача линейного программирования. Симплекс – метод. Транспортная задача. Методы нахождения начального решения транспортной задачи. Метод потенциалов. Общий вид задач нелинейного программирования. Графический метод решения задач нелинейного программирования. Метод множителей Лагранжа. Основные понятия динамического программирования: шаговое управление, управление операцией в целом, оптимальное управление, выигрыш на данном шаге, выигрыш за всю операцию, аддитивный критерий, мультипликативный критерий. Простейшие задачи, решаемые методом динамического программирования. Методы хранения графов в памяти ЭВМ. Задача о нахождении кратчайших путей в графе и методы ее решения. Задача о максимальном потоке и алгоритм Форда–Фалкерсона.		
	Практические работы:	16	
	«Построение простейших математических моделей. Построение простейших статистических моделей» «Решение простейших однокритериальных задач»		

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, курсовая работ (проект) (если предусмотрены)	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
	«Задача Коши для уравнения теплопроводности» «Сведение произвольной задачи линейного программирования к основной задаче линейного программирования» «Решение задач линейного программирования симплекс–методом» «Нахождение начального решения транспортной задачи. Решение транспортной задачи методом потенциалов» «Применение метода стрельбы для решения линейной краевой задачи» «Задача о распределении средств между предприятиями» «Задача о замене оборудования» «Нахождение кратчайших путей в графе.		
Тема 3.2. Задачи в условиях неопределенности	**Содержание учебного материала:**		ОК 01 – ОК 11 ПК 2.1 – ПК 2.5
	Системы массового обслуживания: понятия, примеры, модели. Основные понятия теории марковских процессов: случайный процесс, марковский процесс, граф состояний, поток событий, вероятность состояния, уравнения Колмогорова, финальные вероятности состояний. Схема гибели и размножения. Метод имитационного моделирования. Единичный жребий и формы его организации. Примеры задач. Понятие прогноза. Количественные методы прогнозирования: скользящие средние, экспоненциальное сглаживание, проектирование тренда. Качественные методы прогноза. Предмет и задачи теории игр. Основные понятия теории игр: игра, игроки, партия, выигрыш, проигрыш, ход, личные и случайные ходы, стратегические игры, стратегия, оптимальная стратегия. Антагонистические матричные игры: чистые и смешанные стратегии. Методы решения конечных игр: сведение игры mxn к задаче линейного программирования, численный метод – метод итераций. Область применимости теории принятия решений. Принятие решений в условиях определенности, в условиях риска, в условиях неопределенности. Критерии принятия решений в условиях неопределенности. Дерево решений.	16	
	Практические работы:		
	«Составление систем уравнений Колмогорова. Нахождение финальных вероятностей. Нахождение характеристик простейших систем массового обслуживания.» «Решение задач массового обслуживания методами имитационного моделирования» «Построение прогнозов»	16	

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, курсовая работ (проект) (если предусмотрены)	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
	«Решение матричной игры методом итераций» «Моделирование прогноза» «Выбор оптимального решения с помощью дерева решений»		
Самостоятельная работа		**2**	
Всего часов по МДК		**322**	
Учебная практика		**110**	
Производственная практика (по профилю специальности)		**146**	
Экзамен по модулю		**12**	
Всего:		**590**	

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

«ПМ.01. Осуществление интеграции программных модулей»

3.1. Для реализации программы профессионального модуля должны быть предусмотрены следующие специальные помещения:

Учебная аудитория для проведения учебных занятий (урок, практическое занятие, лабораторное занятие, консультация, лекция), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Лаборатория информационных ресурсов (ауд 1410)

компьютерные столы 33 шт.

стулья 35 шт.

автоматизированные рабочие места обучающихся - сетевые терминалы LG 23CAV42K 32 шт.

стол для преподавателя 1 шт.

автоматизированное рабочее место преподавателя

экран

мультимедийный проектор

маркерная доска 1 шт.

многофункциональное устройство (МФУ) формата А4

Помещение для самостоятельной работы обучающихся (ауд 1406)

Рабочие места на базе вычислительной техники с установленным офисным пакетом с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду организации, а также комплектом оборудования для печати:

персональные компьютеры

посадочных мест – 30 шт.

стол преподавателя - 1 шт.

стул преподавателя - 1 шт.

доска маркерная - 1 шт.

мультимедийный проектор с экраном

3.2. Информационное обеспечение реализации программы

Основная литература

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2021. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/1189951
2. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2020. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/453640
3. Зализняк, В. Е. Введение в математическое моделирование: учебное пособие для среднего профессионального образования / В. Е. Зализняк, О. А. Золотов. — Москва: Издательство Юрайт, 2020. — 133 с. — (Профессиональное образование). — ISBN 978-5-534-13307-3. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/457484

4. Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие для СПО/ Т.М. Зубкова. — Москва: Лань, 2021. — 252 с. — (Среднее профессиональное образование). - ISBN 978-5-8114-5938-4
5. Математическое моделирование и проектирование: учебное пособие / А.С. Коломейченко, И.Н. Кравченко, А.Н. Ставцев, А.А. Полухин; под ред. А.С. Коломейченко. — Москва: ИНФРА-М, 2020. — 181 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015651-4. - Текст: электронный. - URL: https://znanium.com/catalog/product/1044912 (дата обращения: 17.02.2021). – Режим доступа: по подписке.

Дополнительная литература

1. Винокурский, Д.Л. Инструментальные средства информационных систем [Электронный ресурс]: учеб. пособие (курс лекций) / Е.В. Крахоткина, Д.Л. Винокурский .— Ставрополь : изд-во СКФУ, 2018 .— 166 с. : ил. — Режим доступа: https://lib.rucont.ru/efd/687956
2. Древс, Ю. Г. Имитационное моделирование: учебное пособие для среднего профессионального образования / Ю. Г. Древс, В. В. Золотарёв. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 142 с. — (Профессиональное образование). — ISBN 978-5-534-11951-0. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/456617
3. Инструментальные средства разработки мультимедийных приложений [Электронный ресурс]: учеб. пособие (лаб. практикум) / Т.А. Куликова, Н.А. Поддубная .— Ставрополь : изд-во СКФУ, 2019 .— 148 с. : ил. — Режим доступа: https://lib.rucont.ru/efd/726883
4. Соколова, В. В. Разработка мобильных приложений: учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва: Издательство Юрайт, 2020. — 175 с. — (Профессиональное образование). — ISBN 978-5-534-10680-0. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/456795
5. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем: учебник для среднего профессионального образования / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 147 с. — (Профессиональное образование). — ISBN 978-5-534-09823-5. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/454414

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
Раздел 1. Технология разработки программного обеспечения		
ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент	**Оценка «отлично»** - разработан и обоснован вариант интеграционного решения с помощью графических средств среды разработки, указано хотя бы одно альтернативное решение; бизнес-процессы учтены в полном объеме; вариант оформлен в полном соответствии с требованиями стандартов; результаты верно сохранены в системе контроля версий. Оценка «**хорошо**» - разработана и прокомментирована архитектура варианта интеграционного решения с помощью графических средств, учтены основные бизнес-процессы; вариант оформлен в соответствии с требованиями стандартов; результаты сохранены в системе контроля версий. Оценка «**удовлетворительно**» - разработана и архитектура варианта интеграционного решения с помощью графических средств, учтены основные бизнес-процессы с незначительными упущениями; вариант оформлен в соответствии с требованиями стандартов с некоторыми отклонениями; результат сохранен в системе контроля версий.	Экзамен/зачет в форме собеседования: - практическое задание по формированию требований к программным модулям в соответствии с техническим заданием. Защита отчетов по практическим работам. Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	Оценка «**отлично**» - обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием в соответствии с минимальным размером тестового покрытия, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, выявлены ошибки системных компонент (при наличии), заполнены протоколы тестирования. Оценка «**хорошо**»- обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, заполнены протоколы тестирования. Оценка «**удовлетворительно**»- определен	Экзамен/зачет в форме собеседования: практическое задание по разработке тестовых сценариев и наборов для заданных видов тестирования и выполнение тестирования. Защита отчетов по практическим работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики

	размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты, выполнено тестирование интеграции и ручное тестирование, частично выполнено тестирование с применением инструментальных средств, частично заполнены протоколы тестирования.	
ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Оценка «**отлично**» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены все имеющиеся несоответствия стандартам в предложенном коде. Оценка «**хорошо**» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены существенные имеющиеся несоответствия стандартам в предложенном коде. Оценка «**удовлетворительно**» - продемонстрировано знание стандартов кодирования языка программирования, выявлены некоторые несоответствия стандартам в предложенном коде.	Экзамен/зачет в форме собеседования: практическое задание по инспектированию программного кода Защита отчетов по практическим работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
Раздел модуля 2 Инструментальные средства разработки программного обеспечения		
ПК 2.2 Выполнять интеграцию модулей в программное обеспечение	Оценка «**отлично**» - в системе контроля версий выбрана верная версия проекта, проанализирована его архитектура, архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены (при необходимости); протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля и дополнительная обработка исключительных ситуаций в том числе с созданием классов-исключений (при необходимости); определены качественные показатели полученного проекта; результат интеграции сохранен в системе контроля версий. Оценка «**хорошо**» - в системе контроля версий выбрана верная версия проекта, его архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены (при необходимости); выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля и дополнительная обработка исключительных ситуаций (при необходимости); определены качественные показатели полученного	Экзамен/зачет в форме собеседования: практическое задание по обеспечению интеграции заданного модуля в предложенный программный проект Защита отчетов по практическим работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики

	проекта; результат интеграции сохранен в системе контроля версий. Оценка «**удовлетворительно**» - в системе контроля версий выбрана верная версия проекта, его архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, форматы сообщений обновлены (при необходимости); выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля (при необходимости); результат интеграции сохранен в системе контроля версий.	
ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств	Оценка «**отлично**» - в системе контроля версий выбрана верная версия проекта; протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; проанализирована и сохранена отладочная информация; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в полном объеме; результаты отладки сохранены в системе контроля версий. Оценка «**хорошо**» - в системе контроля версий выбрана верная версия проекта; протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в достаточном объеме; результаты отладки сохранены в системе контроля версий. Оценка «**удовлетворительно**» - в системе контроля версий выбрана верная версия проекта; выполнена отладка проекта с применением инструментальных средств среды; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в достаточном объеме; результаты отладки сохранены в системе контроля версий.	Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки программного модуля. Защита отчетов по практическим работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики

ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Оценка «**отлично**» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены все имеющиеся несоответствия стандартам в предложенном коде. Оценка «**хорошо**» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены существенные имеющиеся несоответствия стандартам в предложенном коде. Оценка «**удовлетворительно**» - продемонстрировано знание стандартов кодирования языка программирования, выявлены некоторые несоответствия стандартам в предложенном коде.	Экзамен/зачет в форме собеседования: практическое задание по инспектированию программного кода Защита отчетов по практическим работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
Раздел модуля 3 Математическое моделирование		
ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	Оценка «**отлично**» - обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием в соответствии с минимальным размером тестового покрытия, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, выявлены ошибки системных компонент (при наличии), заполнены протоколы тестирования. Оценка «**хорошо**»- обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, заполнены протоколы тестирования. Оценка «**удовлетворительно**»- определен размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты, выполнено тестирование интеграции и ручное тестирование, частично выполнено тестирование с применением инструментальных средств, частично заполнены протоколы тестирования.	Экзамен/зачет в форме собеседования: практическое задание по разработке тестовых сценариев и наборов для заданных видов тестирования и выполнение тестирования. Защита отчетов по практическим работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	Оценка «**отлично**» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены все имеющиеся несоответствия стандартам в предложенном коде. Оценка «**хорошо**» - продемонстрировано знание стандартов кодирования более чем	Экзамен/зачет в форме собеседования: практическое задание по инспектированию программного кода Защита отчетов по

	одного языка программирования, выявлены существенные имеющиеся несоответствия стандартам в предложенном коде. Оценка «**удовлетворительно**» - продемонстрировано знание стандартов кодирования языка программирования, выявлены некоторые несоответствия стандартам в предложенном коде.	практическим работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач	Экспертное наблюдение за выполнением работ
ОП 02 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.	использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач	
ОК 03 Планировать и реализовывать собственное профессиональное и личностное развитие.	демонстрация ответственности за принятые решения обоснованность самоанализа и коррекция результатов собственной работы	
ОК 04 Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик обоснованность анализа работы членов команды (подчиненных)	
ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей	
ОК 06 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.	соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик	
ОК 07 Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.	эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности	
ОК 08 Использовать средства физической культуры для	эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении	

сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.	профессиональной деятельности	
ОК 09 Использовать информационные технологии в профессиональной деятельности.	эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту	
ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языках.	эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке	
ОК 11 Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере	эффективность использования знаний по финансовой грамотности умение планировать предпринимательскую деятельность в профессиональной сфере	

МИНОБРНАУКИ РОССИИ

Филиал федерального государственного бюджетного образовательного учреждения высшего образования
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
в г. Артеме
(ФИЛИАЛ ФГБОУ ВО «ВВГУ» В Г. АРТЕМЕ)

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА

для проведения текущего контроля и промежуточной аттестации по профессиональному модулю

ПМ.01 Осуществление интеграции программных модулей

программы подготовки специалистов среднего звена

09.02.07 Информационные системы и программирование

Форма обучения: *очная*

Артем 2022

Контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации по профессиональному модулю *ПМ.01 Осуществление интеграции программных модулей* в соответствии с требованиями ФГОС СПО по специальности 09.02.07 **«Информационные системы и программирование»**, утвержденного приказом Минобрнауки РФ от 09 декабря 2016 г., № 1547, примерной образовательной программой, рабочей программой учебной дисциплины.

Разработчик(и): *Волошин Е.В., Ематина Н.И., преподаватель*

Утверждена на заседании цикловой методической комиссии по профилю математических и информационных дисциплин, протокол № 1 от 01.10.2022 г

Председатель ЦМК_____*А.С.Бажина*
подпись

1 Общие сведения

Контрольно-оценочные средства (далее – КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу профессионального модуля ПМ.01 «Осуществление интеграции программных модулей».

КОС разработаны на основании:

– основной образовательной программы СПО по специальности 09.02.07. «Информационные системы и программирование».

– рабочей программы профессионального модуля ПМ.01 «Осуществление интеграции программных модулей».

Формой промежуточной аттестации является квалификационный экзамен.

Код ОК, ПК	Код результата обучения	Наименование
ОК 1 ОК 2 ОК 3 ОК 4 ОК 5 ОК 6 ОК 7 ОК 8 ОК 9 ОК 10 ОК 11 ПК 2.1 ПК 2.2 ПК 2.3 ПК 2.4 ПК 2.5	З1	модели процесса разработки программного обеспечения
	З2	основные принципы процесса разработки программного обеспечения
	З3	основные подходы к интегрированию программных модулей
	З4	основы верификации и аттестации программного обеспечения
	У1	использовать выбранную систему контроля версий
	У2	использовать методы для получения кода с заданной функциональностью и степенью качества
	П1	процесса разработки программного обеспечения
	П2	процесса разработки программного обеспечения
	П3	в интегрировании программных модулей
	П4	в верификации и аттестации программного обеспечения

2 Распределение типов контрольных заданий по элементам знаний и умений, контролируемых в процессе изучения

Текущий контроль распределение типов заданий по элементам знаний и умений.

Код результата обучения	Содержание учебного материала (темы)	Вид оценочного средства	
		Текущий контроль	Промежуточная аттестация
МКД.01.01. Технология разработки программного обеспечения			
З1 З2 З3 З4 У1 У2 П1 П2 П3 П4	**Тема 1.1 Основные понятия и стандартизация требований к программному обеспечению**	Практическое занятие «Анализ предметной области» Практическое занятие «Разработка и оформление технического задания» Практическое занятие «Построение архитектуры программного средства» Практическое занятие «Изучение работы в системе контроля версий»	Собеседование
	Тема 1.2 Описание и анализ требований. Диаграммы IDEF	Практическое занятие «Построение диаграммы Вариантов использования и диаграммы. Последовательности» Практическое занятие «Построение диаграммы Кооперации и диаграммы Развертывания» Практическое занятие «Построение диаграммы Деятельности, диаграммы Состояний и диаграммы Классов» Практическое занятие «Построение диаграммы компонентов» Практическое занятие «Построение диаграмм потоков данных»	Собеседование
	Тема 1.3 Оценка качества программных средств	Практическое занятие «Разработка тестового сценария» Практическое занятие «Оценка необходимого количества тестов» Практическое занятие «Разработка тестовых пакетов»	Собеседование
	Тема 1.4 Оценка качества программных средств	Практическое занятие «Оценка программных средств с помощью метрик» Практическое занятие «Инспекция программного кода на предмет соответствия стандартам кодирования»	Собеседование
МКД.01.02. Инструментальные средства разработки программного обеспечения			
З1 З2	**Тема 2.1. Современные тех-**	Практическое занятие «Разработка структуры проекта»	Собеседование

З3 З4 У1 У2 П1 П2 П3 П4	**нологии и инструменты интеграции.**	Практическое занятие «Разработка модульной структуры проекта (диаграммы модулей)» Практическое занятие «Разработка перечня артефактов и протоколов проекта» Практическое занятие «Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)» Практическое занятие «Разработка и интеграция модулей проекта (командная работа)» Практическое занятие «Отладка отдельных модулей программного проекта» Практическое занятие «Организация обработки исключений»	
	Тема 2.2. Инструментарий тестирования и анализа качества программных средств	Практическое занятие «Применение отладочных классов в проекте» Практическое занятие «Отладка проекта» Практическое занятие «Инспекция кода модулей проекта» Практическое занятие «Тестирование интерфейса пользователя средствами инструментальной среды разработки» Практическое занятие «Разработка тестовых модулей проекта для тестирования отдельных модулей» Практическое занятие «Выполнение функционального тестирования» Практическое занятие «Тестирование интеграции» Практическое занятие «Документирование результатов тестирования»	Собеседование
МДК.01.03 Математическое моделирование			
З1 З2 З3 З4 У1 У2 П1 П2 П3 П4	**Тема 3.1. Основы моделирования. Детерминированные задачи**	Практическое занятие «Построение простейших математических моделей. Построение простейших статистических моделей» Практическое занятие «Решение простейших однокритериальных задач» Практическое занятие «Задача Коши для уравнения теплопроводности» Практическое занятие «Сведение произвольной задачи линейного программирования к основной задаче линейного программирования»	Собеседование Тестирование

		Практическое занятие «Решение задач линейного программирования симплекс–методом» Практическое занятие «Нахождение начального решения транспортной задачи. Решение транспортной задачи методом потенциалов» Практическое занятие «Применение метода стрельбы для решения линейной краевой задачи» Практическое занятие «Задача о распределении средств между предприятиями» Практическое занятие «Задача о замене оборудования» Практическое занятие «Нахождение кратчайших путей в графе.	
	Тема 3.2. Задачи в условиях неопределенности	Практическое занятие «Составление систем уравнений Колмогорова. Нахождение финальных вероятностей. Нахождение характеристик простейших систем массового обслуживания.» Практическое занятие «Решение задач массового обслуживания методами имитационного моделирования» Практическое занятие «Построение прогнозов» Практическое занятие «Решение матричной игры методом итераций» Практическое занятие «Моделирование прогноза» Практическое занятие «Выбор оптимального решения с помощью дерева решений»	Собеседование Тестирование

3 Описание процедуры оценивания

Результаты обучения по дисциплине, уровень сформированности компетенций оцениваются по четырёх бальной шкале оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Текущая аттестация по дисциплине проводится с целью систематической проверки достижений обучающихся. Объектами оценивания являются: степень усвоения теоретических знаний, уровень овладения практическими умениями и навыками по всем видам учебной работы, качество выполнения самостоятельной работы, учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине).

При проведении промежуточной аттестации оценивается достижение студентом запланированных по дисциплине результатов обучения, обеспечивающих результаты освоения образовательной программы в целом. Оценка на зачете выставляется с учетом оценок, полученных при прохождении текущей аттестации. При выведении итоговой оценки «вес» текущих оценок составит не ме-

нее 80% в итоговой, т.е. результат достижений учащегося по дисциплине зависит от эффективности его работы в течение учебного семестра.

Критерии оценивания устного ответа

(оценочные средства: кейс-задания, собеседование)

4 баллов - ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

4 балла - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

3 балла – ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

2 балла – ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

Критерии оценивания письменной работы

(оценочные средства: кейс-задания, бизнес-проект в форме презентации).

5 баллов - студент выразил своё мнение по сформулированной проблеме, аргументировал его, точно определив ее содержание и составляющие. Проблема раскрыта полностью, выводы обоснованы. Приведены данные отечественной и зарубежной литературы, статистические сведения, информация нормативно-правового характера. Студент владеет навыком самостоятельной работы по заданной теме; методами и приемами анализа теоретических и/или практических аспектов изучаемой области. Расчеты проведены грамотно, логические выводы сделаны верно. Фактических ошибок, связанных с пониманием проблемы, нет; графически работа оформлена правильно.

4 балла - работа характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки в расчетной части проекта. Проблема раскрыта. Не все выводы сделаны и/или обоснованы. Для аргументации приводятся данные исследования рынка. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет. Допущены одна-две ошибки в оформлении работы.

3 балла – студент проводит достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимает базовые основы и теоретическое обоснование выбранной темы. Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы Проведен анализ проблемы без привлечения дополнительной литературы. Допущено не более 3 ошибок в расчетной части проекта, оформлении работы.

2 балла - работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа автора проекта. Не раскрыта структура и теоретическая составляющая темы. Проблема не раскрыта. Выводы отсутствуют. Допущено более

трех ошибок в расчетной части проекта, оформление работы не соответствует требованиям выполнения бизнес-проектов.

Критерии оценивания тестового задания

При применении теста в качестве оценочного средства для проведения промежуточной аттестации используется шкала интервальных баллов, соответствующая итоговой оценке, а также критерии её выставления в привязке к четырех балльной системе:

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Количество правильных ответов	91 % и ≥	от 81% до 90,9 %	не менее 70%	менее 70%

Критерии выставления оценки студенту на зачете

(оценочные средства: устный опрос в форме собеседования)

Оценка по промежуточной аттестации	Характеристика качества сформированности компетенций
«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций на продвинутом уровне: обнаруживает всестороннее, систематическое и глубокое знание учебного материала, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на пороговом уровне: имеет знания только основного материала, но не усвоил его деталей, в ходе контрольных мероприятий допускаются значительные ошибки, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, при оперировании знаниями и умениями при их переносе на новые ситуации.
«не зачтено» / «неудовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже порогового: выявляется полное или практически полное отсутствие знаний значительной части программного материала, студент допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, умения и навыки не сформированы.

4 Примеры оценочных средств для проведения текущей аттестации

МКД.01.01 Технология разработки программного обеспечения

Практическое занятие №1. «Анализ предметной области»

Практическое занятие №2. «Разработка и оформление технического задания»

Разрабатывая техническое задание ГОСТ, придерживаемся принципов, которые помогут избежать абстракции в описании будущего товара, а также учесть интересы Заказчика, конечного потребителя и исполнителя.

Выделены следующие принципы:

- совместная работа всей проектной команды
- максимально подробное описание конечного продукта
- сдача-приёмка конечного продукта.

Несмотря на столь большую значимость технического задания требования к написанию изложены только в двух документах:

- ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению
- ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.

Практическое занятие №3. «Проектирование архитектуры программного средства»

При возникновении потребностей в заказе, приобретении, разработке, эксплуатации и сопровождении программ перед всеми сторонами, вовлеченными в жизненный цикл программного средства (ПС), возникает целый ряд вопросов, связанных с определением и детальным структурированием жизненного цикла (ЖЦ) ПС, с организационными и техническими правами и обязанностями сторон, с управлением ЖЦ и контролем за его реализацией. Одним из действенных инструментов для решения данных вопросов является использование унифицированных подходов, закрепленных в современных международных и российских стандартах.

Понятия «жизненный цикл системы» или «жизненный цикл программного средства» часто появляются в статьях и звучат в разговорах разработчиков, по крайней мере руководителей проектов и подразделений.

В стандартах, регламентирующих жизненный цикл программных средств, обобщаются опыт и результаты исследований множества специалистов и рекомендуются наиболее эффективные современные методы и процессы создания и развития комплексов программ. В результате таких обобщений оттачиваются технологические процессы и приемы разработки, а также методическая база для их автоматизации.

ЖЦ ПС в стандартах представляет собой набор этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих ведение работ от подготовки технического задания до завершения испытаний ряда версий и окончания эксплуатации ПС или информационной системы (ИС).

Стандарты включают правила описания исходной информации, способов и методов выполнения операций, устанавливают правила контроля технологических процессов, требования к оформлению их результатов, а также регламентируют содержание технологических и эксплуатационных документов на комплексы программ. Они определяют организационную структуру коллектива, обеспечивают распределение и планирование заданий, а также контроль за ходом создания ПС.

Кроме вопросов выбора типа общего устройства ЖЦ есть проблемы с решением частных вопросов о включении или невключении в ЖЦ отдельных работ, очень важных для качества ПС и системы: что документировать при создании системы и ПС, какие работы должны будут гарантировать качество продукта, с какой степенью организационной независимости должны выполняться проверочные процедуры разных типов, чем будет обеспечиваться соответствие разрабатываемого ПС требованиям ко всей системе и соответствие ПС потребностям в системе.

Для того чтобы привнести порядок и понимание, общие для любых сторон, участвующих в ЖЦ систем и ПС, давно разрабатывались стандарты различных уровней утверждения - национальные и международные.

В России основы построения и использования профилей стандартов ЖЦ ПС заложены принятием в качестве базового стандарта ГОСТ Р ИСО/МЭК 12207. Данный документ введен в дей-

ствие с 1 июля 2000 г., тесно взаимоувязан с рядом стандартов, принятых ранее, и с некоторыми стандартами, разрабатываемыми в данное время на основе прямого применения стандартов ИСО.

Актуальность стандарта ГОСТ Р ИСО/МЭК 12207 для современных условий настолько высока, что принятие в ISO его исходного, международного варианта вскоре вызвало самую положительную оценку российских экспертов. Был дан ряд рекомендаций по его использованию в реальных условиях. В данном стандарте программное обеспечение (ПО) или программный продукт определяется как набор компьютерных программ, процедур и, возможно, связанной с ними документации и данных.

Процесс определяется как совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные. Каждый процесс характеризуется определенными задачами и методами их решения, исходными данными, полученными от других процессов, и результатами.

В соответствии с ГОСТ Р ИСО/МЭК 12207 все процессы ЖЦ ПО разделены на три группы:

1) Основные процессы:
   - приобретение;
   - поставка;
   - разработка;
   - эксплуатация;
   - сопровождение.

2) Вспомогательные процессы:
   - документирование;
   - управление конфигурацией;
   - обеспечение качества;
   - верификация;
   - аттестация;
   - совместная оценка;
   - аудит;
   - разрешение проблем.

3) Организационные процессы:
   - управление;
   - усовершенствование;
   - создание инфраструктуры;
   - обучение.

Процесс разработки предусматривает действия и задачи, выполняемые разработчиком, и включает следующие действия:

А) Подготовительная работа начинается с выбора модели ЖЦ ПО, соответствующей масштабу, значимости и сложности проекта. Действия и задачи процесса должны соответствовать выбранной модели. Разработчик должен выбрать, адаптировать к условиям проекта и использовать согласованные с заказчиком стандарты, методы и средства разработки, а также составить план выполнения работ.

Б) Анализ требований к системе подразумевает определение ее функциональных возможностей, пользовательских требований, требований к надежности и безопасности, требований к внешним интерфейсам и т.д. Требования к системе оцениваются исходя из критериев реализуемости и возможности проверки при тестировании.

Анализ требований к ПО предполагает определение следующих характеристик для каждого компонента ПО:

− функциональных возможностей, включая характеристики производительности и среды функционирования компонента;

− внешних интерфейсов;

− спецификаций надежности и безопасности;

− эргономических требований;

− требований к используемым данным;

− требований к установке и приемке;

− требований к пользовательской документации;

− требований к эксплуатации и сопровождению.

Требования к ПО оцениваются исходя из критериев соответствия требованиям к системе, реализуемости и возможности проверки при тестировании.

В) Проектирование архитектуры системы на высоком уровне заключается в определении компонентов ее оборудования, ПО и операций, выполняемых эксплуатирующим систему персоналом. Архитектура системы должна соответствовать требованиям, предъявляемым к системе, а также принятым проектным стандартам и методам.

Проектирование архитектуры ПО включает задачи (для каждого компонента ПО):

− трансформацию требований к ПО в архитектуру, определяющую на высоком уровне структуру ПО и состав ее компонентов;

− разработку и документирование программных интерфейсов ПО и баз данных;

− разработку предварительной версии пользовательской документации;

− разработку и документирование предварительных требований к тестам и планам интеграции ПО.

Архитектура компонентов ПО должна соответствовать требованиям, предъявляемым к ним, а также принятым проектным стандартам и методам.

Г) Детальное проектирование ПО включает следующие задачи:

− описание компонентов и интерфейсов между ними на более низком уровне, достаточном для их последующего самостоятельного кодирования и тестирования;

− разработку и документирование детального проекта базы данных;

− обновление (при необходимости) пользовательской документации;

− разработку и документирование требований к тестам и плана тестирования компонентов ПО;

− обновление плана интеграции ПО.

Д) Кодирование и тестирование ПО охватывает задачи:

− разработку и документирование каждого компонента ПО и базы данных, а также совокупности тестовых процедур и данных для их тестирования;

− тестирование каждого компонента ПО и базы данных на соответствие, предъявляемых к ним требованиям. Результаты тестирования компонентов должны быть документированы;

− обновление (при необходимости) пользовательской документации;

− обновление плана интеграции ПО.

Е) Интеграция ПО предусматривает сборку разработанных компонентов ПО в соответствии с планом интеграции и тестирование агрегированных компонентов. Для каждого из агрегированных компонентов разрабатываются наборы тестов и тестовые процедуры, предназначенные для проверки каждого из квалификационных требований при последующем квалификационном тестировании.

Интеграция системы заключается в сборке всех ее компонентов, включая ПО и оборудование. После интеграции система, в свою очередь, подвергается квалификационному тестированию

на соответствие совокупности требований к ней. При этом также производится оформление и проверка полного комплекта документации на систему.

Ж) Квалификационное тестирование — это набор критериев и условий, которые необходимо выполнить, чтобы квалифицировать программный продукт как соответствующий своим спецификациям и готовый к использованию в условиях эксплуатации.

Квалификационное тестирование ПО проводится разработчиком в присутствии заказчика (по возможности) для демонстрации того, что ПО удовлетворяет своим спецификациям и готово к использованию в условиях эксплуатации. Квалификационное тестирование выполняется для каждого компонента ПО по всем разделам требований при широком варьировании тестов. При этом также проверяются полнота технической и пользовательской документации и ее адекватность самим компонентам ПО.

З) Установка ПО осуществляется разработчиком в соответствии с планом в той среде и на том оборудовании, которые предусмотрены договором. В процессе установки проверяется работоспособность ПО и баз данных. Если устанавливаемое программное обеспечение заменяет существующую систему, разработчик должен обеспечить их параллельное функционирование в соответствии с договором.

И) Приемка ПО предусматривает оценку результатов квалификационного тестирования ПО и системы и документирование результатов оценки, которые проводятся заказчиком с помощью разработчика. Разработчик выполняет окончательную передачу ПО заказчику в соответствии с договором, обеспечивая при этом необходимое обучение и поддержку.

Задание**:** разработать проект архитектуры программного средства в соответствии с ГОСТ Р ИСО/МЭК 12207

Алгоритм выполнения работы

С целью реализации начальных этапов разработки ПС в соответствии с техническим заданием:

− выполнить подготовительную работу;

− провести анализ требований к ПС;

− выполнить проектирование архитектуры ПС на высоком уровне.

Контрольные вопросы

1) В какой форме представлен ЖЦ ПО в ГОСТ Р ИСО/МЭК 12207?

2) Какая нормативная информация включена в современные стандарты, регламентирующие жизненный цикл программных средств?

3) Чем объясняется актуальность стандарта ГОСТ Р ИСО/МЭК 12207 в настоящее время?

4) Приведите определение программного обеспечения в соответствии с ГОСТ Р ИСО/МЭК 12207.

5) Как определяется процесс в ГОСТ Р ИСО/МЭК 12207?

6) Какие группы процессов ЖЦ ПО выделены в ГОСТ Р ИСО/МЭК 12207?

7) Какие действия и задачи, выполняемые разработчиком, предусмотрены ГОСТ Р ИСО/МЭК 12207 в процессе разработки ПС?

Практическое занятие №4. «Изучение работы в системе контроля версий»

Цель работы: получение первоначальных навыков использования систем контроля версий исходного кода программ, получение первоначальных навыков организации коллективной разработки программного обеспечения. Создание в системе контроля версий репозитория для нового проекта и выполнение всех основных действий с исходным кодом программы, связанных с контролем версий.

Управление версиями — это искусство работы с изменениями информации. Долгое время оно было жизненно важным инструментом программистов, которым необходимо произвести небольшие изменения в программе, или же сделать откат изменений, возвращаясь к предыдущей версии. Однако полезность систем управления версиями выходит далеко за пределы мира разработчиков программного обеспечения. Управление версиями требуется повсюду, где можно встретить людей, использующих компьютер для работы с постоянно изменяющейся информацией Software Configuration Management или Конфигурационное управление подразумевает под собой комплекс методов, направленных на то, чтобы систематизировать изменения, вносимые разработчиками в программный продукт в процессе его разработки и сопровождения, сохранить целостность системы после изменений, предотвратить нежелательные и непредсказуемые эффекты, а также сделать процесс внесения изменений более формальным. Изначально управление конфигурацией применялось не в программировании, но в связи с высокой динамичностью сферы разработки ПО, в ней она особенно полезна. К процедурам можно отнести создание резервных копий, контроль исходного кода, требований проекта, документации и т. д. Степень формальности выполнения данных процедур зависит от размеров проекта, и при правильной ее оценке данная концепция может быть очень полезна. Конфигурационное управление требует выполнения множества трудоемких рутинных операций. На практике, в большинстве случаев, для конфигурационного управления применяются специальные системы контроля версий исходного кода программ. В качестве примера такой системы рассмотрим самую распространенную на сегодняшний день Subversion. Subversion это бесплатная система управления версиями с открытым исходным кодом. Subversion позволяет управлять файлами и каталогами, а также сделанными в них изменениями во времени. Это позволяет восстановить более ранние версии данных, даѐт возможность изучить историю всех изменений. Благодаря этому многие считают систему управления версиями своего рода «машиной времени».

Практическое занятие №5. «Построение диаграммы Вариантов использования и диаграммы. Последовательности»

Практическое занятие №6. «Построение диаграммы Кооперации и диаграммы Развертывания»

Практическое занятие №7. «Построение диаграммы Деятельности, диаграммы Состояний и диаграммы Классов»

Практическое занятие №8. «Построение диаграммы компонентов»

Практическое занятие №9 «Построение диаграмм потоков данных»

МКД.01.02 Инструментальные средства разработки программного обеспечения

Практическое занятие

Разработка модульной структуры проекта (диаграммы модулей)

Цель занятия: изучение процесса разработки модульной структуры программного обеспечения, осуществляемого с помощью структурных карт Константайна.

Оборудование, технические и программные средства: персональный компьютер, среда программирования Visual Studio 2019.

Задание:

1. В соответствии с требованиями, предъявляемыми техническим заданием, и результатами внешнего проектирования разработаем модульную структуру подсистемы обслуживания клиента по его кредитной карте в банкомате.

Теоретические сведения:

Чтобы добиться декомпозиции на модули максимальной прочности и минимального сцепления, необходимо спроектировать модульную структуру в виде дерева, в том числе и со сросшимися ветвями. В узлах такого дерева размещаются программные модули, а направленные дуги (стрелки) показывают статическую подчинённость модулей, т.е. каждая дуга показывает, что в тексте модуля, из которого она исходит, имеется ссылка на модуль, в который она входит.

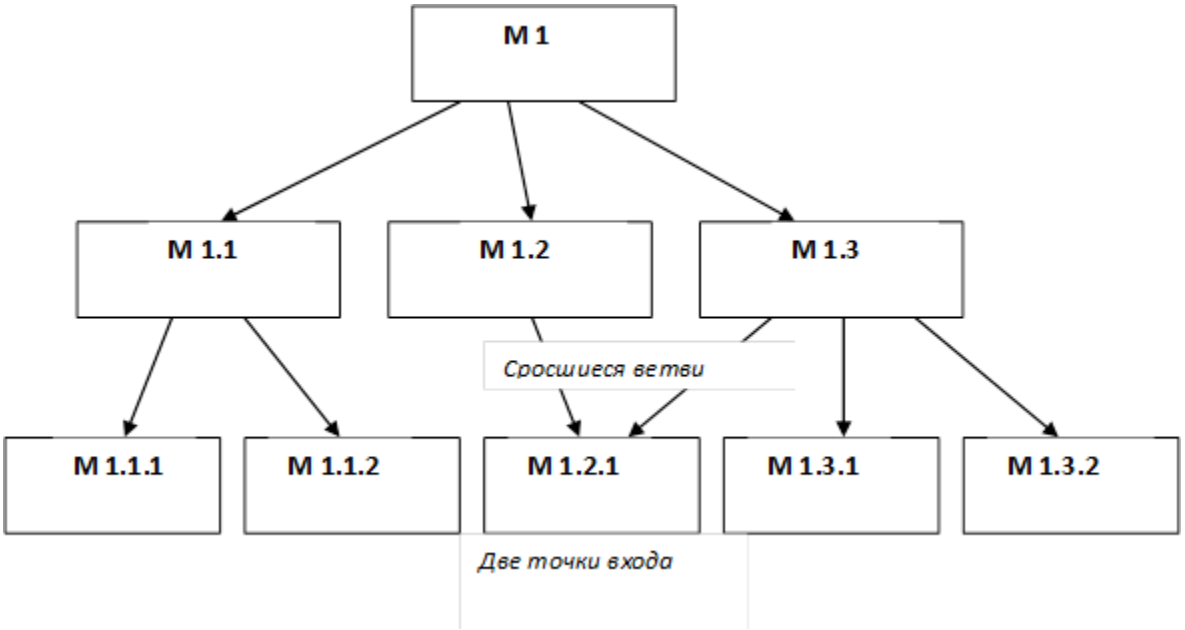

Рис.1. Пример иерархического дерева модулей

При этом модульная структура программы должна, помимо картинки, включать спецификацию программного модуля.

Спецификация программного модуля должна содержать:

- синтаксическую спецификацию его входов (имя модуля, типы передаваемых ему параметров, типы возвращаемых результатов, синтаксис обращения к любому ему входов)
- функциональную спецификацию (описание семантики функций, выполняемых модулем по каждому из его входов).

В процессе разработки модульная структура может по-разному использоваться для определения порядка программирования модулей — восходящая и нисходящая разработка.

В восходящей разработке модули программируются, начиная с нижних уровней, и сразу тестируются, исходя из функциональных спецификаций. Такой порядок представляется естественным, т.к. каждый новый модуль выражается через уже запрограммированные и проверенные модули. Однако современная технология не рекомендует этот прием, т.к. при этом трудно обеспечить концептуальную целостность ПС.

Концептуальная целостность предполагает общие принципы реализации, предположения, структуры данных, - а они могут быть ещё не ясны в начальных стадиях разработки.

Перепрограммирование же модулей нижних уровней связано с большими затратами, т.к. требует не только повторной разработки текстов, но и повторного тестирования.

Предпочтительной является нисходящая разработка. В этой технологии программирование начинается с модуля с самого верхнего уровня. При этом для тестирования модули нижних уровней заменяются простыми по конструкции имитаторами, которые либо моделируют работу нижних уровней (например, реализуют таблицы; вход-отклик), либо просто сообщают о своём вызове и завершаются признаком успеха. Такая реализация обеспечивает большую концептуальную целостность и меньший объём разрабатываемых тестов, каждый модуль здесь тестируется при т.н. «естественном» состоянии информационной среды, т.к. он вызывается реальным (оттестированным) модулем верхнего уровня.

Выполнение работы:

В составе программного обеспечения можно выделить следующие программные модули: Головной модуль (Main module), Модуль управления устройством считывания кредитной кары (Credit cart control module), Модуль аутентификации (Autentification module) и Модуль получения и обработки запроса на обслуживание (Reception and processing module). Кроме этого, в состав ПО необходимо включить модуль данных кредитной карты (Credit cart data).

Основной функцией Головного модуля является организация общего управления поведением подсистемы и выполняет вызов всех остальных программных модулей.

Модуль управления устройством считывания кредитной карты выполняет функции, связанные с обработкой кредитной карты: ввод, считывание хранящейся на ней информации, удаление.

Модуль аутентификации выдает сообщение клиенту на ввод ключевых данных, выполняет получение пароля и проверку его правильности.

Модуль получения и обработки запроса на обслуживание выполняет следующие функции: Получение запроса на обслуживание и проверка возможности его исполнения, Обработка запроса на обслуживание, включающая такие действия как:

- обработка внутренней банковской документации по клиенту;
- распечатка баланса клиента;
- выдача наличных денег и информирование компьютера банка об изъятых из банка деньгах;
- распечатка операции клиента.

На рис. 1 приведена структурная карта, демонстрирующая отношения между указанными модулями системы.

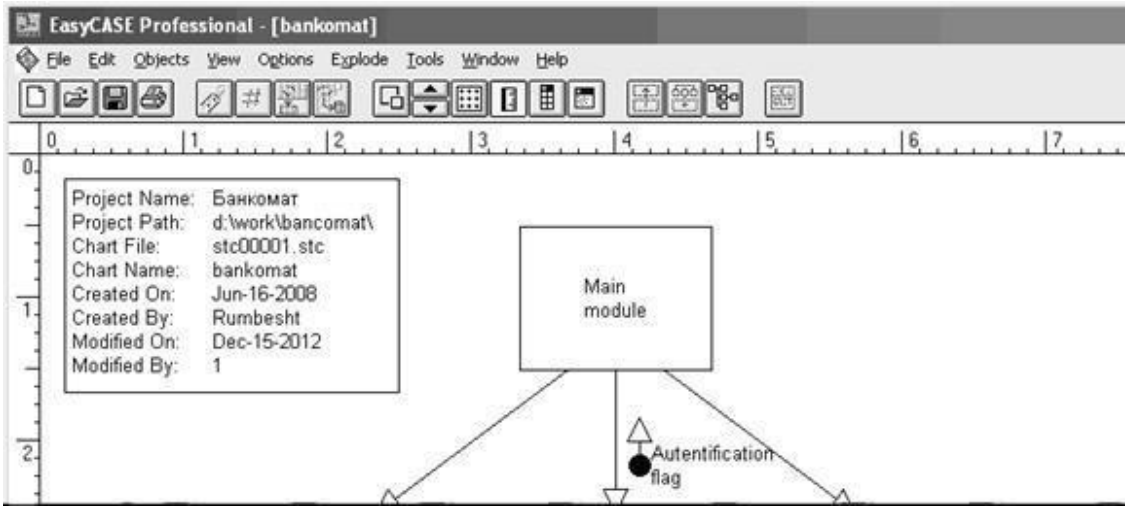

Рис. 2. Модульная структура программного обеспечения

Согласно этой диаграмме головной модуль обращается к модулям управления устройством считывания кредитной карты, аутентификации и получения, и обработки запроса на обслуживание. Вызов указанных модулей осуществляется согласно внутренней логике головного модуля, реализующей следующий сценарий: при инициации действий со стороны клиента головной модуль, вызывает модуль управления устройством считывания кредитной карты для ее ввода и считывания с нее информации. После завершения считывания управление возвращается головному модулю, который затем обращается к модулю аутентификации. Модуль аутентификации проверяет подлинность клиента и вместе с результатом этой проверки возвращает управление головному модулю. В зависимости от результатов аутентификации головной модуль либо вызывает модуль управления устройством считывания для удаления кредитной карты, либо обращается к модулю получения и

обработки запроса на обслуживание для предоставления требуемого сервиса. Если осуществляется вызов получения и обработки запроса на обслуживание, то после завершения его работы головной модуль обращается к модулю управления устройством считывания для удаления кредитной карты.

Обмен данными между программными модулями осуществляется через общую область памяти, в которую модуль управления устройством считывания помещает данные о пароле (Parol), атрибуты клиента (Client Attributes) и лимит денег на счету (Limit of money). Модуль аутентификации получает из этой общей области памяти сведения о пароле и возвращает в головной модуль управляющий параметр Autentification flag, содержащий результат аутентификации. Модуль получения и обработки запроса на обслуживание для своей работы получает из общей области памяти атрибуты клиента и лимит денег на счету.

Контрольные вопросы:

1. Цель разработки модульной структуры.
2. Понятие программного модуля, передачи управления, организации связи по управлению и по данным.
3. Виды связности модулей.
4. Виды целостности модулей.
5. Типовые модульные структуры.
6. Проектирование модульной структуры с помощью структурных карт.
7. Построение структурных карт с помощью программного продукта EasyCASE Professional Version 4.21.016.

Практическое занятие.

Тестирование интеграции

Цель работы: ознакомится с методами и видами тестирования ПО и провести тестирование разрабатываемого программного продукта.

Тестирование программного обеспечения.

Тестирование (software testing) деятельность, выполняемая для оценки и улучшения качества программного обеспечения. Эта деятельность, в общем случае, базируется на обнаружении дефектов и проблем в программных системах. Тестирование программных систем состоит из динамической верификации поведения программ на конечном (ограниченном) наборе тестов (set of test cases), выбранных соответствующим образом из обычно выполняемых действий прикладной области и обеспечивающих проверку соответствия ожидаемому поведению системы. Сегодня тестирование рассматривается как деятельность, которую необходимо проводить на протяжении всего процесса разработки и сопровождения и является важной частью конструирования программных продуктов. В области знаний Качество программного обеспечения (Software Quality) техники управления качеством четко разделены на статические (без выполнения кода) и динамические (с выполнением кода). Обе эти категории важны. Данная область знаний - Тестирование касается динамических техник. Тестирование тесно связано с областью знаний Конструирование (Software Construction). Более того, модульное (unit-) и интеграционное тестирование все чаще рассматривают как неотъемлемый элемент деятельности по конструированию. В литературе, посвященной программной инженерии, встречается множество терминов, описывающих нарушение функционирования программных систем недостатки (faults), дефекты (defects), сбои (failures), ошибки (errors) и др. Соответствующая терминология описана в IEEE Std Глоссарии терминов по программной инженерии. Важно четко разделять причину нарушения работы прикладных систем, обычно описываемую терминами недостаток или дефект, и наблюдаемый нежелательный эффект, вызываемый этими причинами сбой. Термин ошибка, в зависимости от контекста, может описывать и как причину сбоя, и сам сбой. Тестирование позволяет обнаружить дефекты, приводящие к сбоям. Необходимо понимать, что причина сбоя не всегда может быть однозначно определена. Не существует теоретических критериев, позволяющих гарантированно определить какой именно дефект, приводит к наблюдаемому сбою. Ключевые вопросы (Key Issues) тестирования ПО:

1 Критерии отбора тестов/критерии адекватности тестов, правила прекращения тестирования.

2 Эффективность тестирования/цели тестирования. Тестирование — это наблюдение за выполнением программы, запущенной в целях тестирования с заданными параметрами, по заданному сценарию или с другими заданными начальными условиями или целями тестирования. Эффективность теста может быть определена только в контексте заданных условий.

3 Тестирование для идентификации дефектов. Данный случай тестирования подразумевает успешность процедуры тестирования, если дефект найден. Это отличается от подхода в тестировании, когда тесты запускаются для демонстрации того, что программное обеспечение удовлетворяет предъявляемым требованиями и, соответственно, тест считается успешным, если не найдено дефектов.

4 Проблема оракула. Оракул, в данном контексте, любой агент (человек или программа), оценивающий поведение программы, формулируя вердикт - тест пройден или нет.

5 Теоретические и практические ограничения тестирования. Теория тестирования выступает против необоснованного уровня доверия к серии успешно пройденных тестов. Основная причина этого в том, что полное (всеобъемлющее) тестирование недостижимо для реального программного обеспечения.

Практическое занятие. Отладка проекта

Проанализировать прокрутку и записать условие задачи для программы, представленной в правой части. Выполнить прокрутки для программ из практического занятия № 6.	Термином прокрутка обозначают выполнение программы вручную так, как если бы она выполнялась на компьютере. Запись прокрутки для данного примера может выглядеть следующим образом:
	x = 0; 0 <= 6; 0 < 4; y = x = 0;
	вывод x = 0 и y = 0; x = x + 2 = 0 + 2 = 2 2 <= 6;
	2 < 4; y = x = 2; вывод x = 2 и y = 2;
	x = x + 2 = 2 + 2 = 4 4 <= 6; 4 = 4; y = 1;
	вывод x = 4 и y = 1; x = x + 2 = 4 + 2 = 6 6 = 6;
	6 > 4; y = x * x = 6 * 6 = 36;
	вывод x = 6 и y = 36; x = x + 2= 6 + 2 = 8 8 > 6
	
	Выполнение программы закончено.

Выполнить программу с использованием средств интерактивной отладки (пошаговое выполнение приложения). Внести изменения в программу (например, заменить == на = или убрать строчку x = x + 2; и т.п.), выполняя каждый раз отладку по шагам с обходом.	Пошаговое выполнение приложения может выполняться двумя способами: – без трассировки вызываемых методов (Отладка / Шаг с обходом или клавиша F10, или иконка на панели инструментов Отладка; – с трассировкой вызываемых методов (Отладка / Шаг с заходом или клавиша F11, или иконка на панели инструментов Отладка). При обоих способах производится останов перед выполнением текущей строки исходного кода (продолжается отладка повторным нажатием клавиш F10 или F11). Различия проявляются только тогда, когда в программе встречается вызов функции. Если выбрать команду Шаг с заходом, то отладчик войдет в функцию и начнет выполнять шаг за шагом все ее операторы. При выборе

	команды Шаг с обходом отладчик выполнит функцию как единое целое и перейдет к строке, следующей за вызовом функции. Эту команду удобно применять в тех случаях, когда в программе делается обращение к стандартной функции или созданной подпрограмме, которая уже была протестирована. Если поместить курсор мыши на имя переменной и удерживать его некоторое время, то появится всплывающее окно, в котором будет указано текущее значение переменной. После отладки, необходимо вывести приложение из отладочного режима командой Отладка / Остановить отладку илиShift+F5 или одноименной кнопкой на панели инструментов Отладка.
Выполнить программу с использованием средств планируемой отладки, установив точки останова на операторах вывода и пошагово проследив вывод на консоль. Удалить точки останова.	Создать исполняемый файл без отладочной информации Release. Объяснить назначение папок и файлов решения проекта. Выполнить файл с расширением *.exeв папке Debug.

Практическое занятие.

Разработка протоколов проекта

Цель занятия: получение первичных навыков планирования работ по разработке и внедрению автоматизированных информационных систем, разработка протоколов проекта.

Оборудование, технические и программные средства: персональный компьютер, среда программирования Visual Studio 2019, Microsoft Project.

Задание:

На основе технического задания на разработку и внедрение автоматизированной информационной системы сформировать календарный план выполнения работ по проекту.

Теоретические сведения:

Для проведения успешного проекта нужно оценить объем предстоящих работ, возможный риск, требуемые ресурсы, предстоящие задачи, определить контрольные точки, стоимость и план работ, которому желательно следовать. Процесс руководство программным проектом включает решение вышеперечисленных задач. Этот процесс начинается перед технической работой, продолжается по мере развития ПО от идеи к реальности и достигает наибольшей интенсивности к концу работ. Основной задачей при планировании является определение структуры распределения работ WBS (Work Breakdown Structure) с помощью средств планирования работ (например, MS Project). План выполнения работ составляется на основе декомпозиции проекта вплоть до постановки элементарных задач, которые могут быть выяснены по результатам предварительного анализа. При этом возможно применение содержательных моделей системного анализа. Например, использование модели декомпозиции типа «жизненный цикл» позволит разбивать отдельные задачи на подзадачи путем определения последовательности действий.

Процесс декомпозиции будет определяться принятой моделью жизненного цикла разработки программного обеспечения.

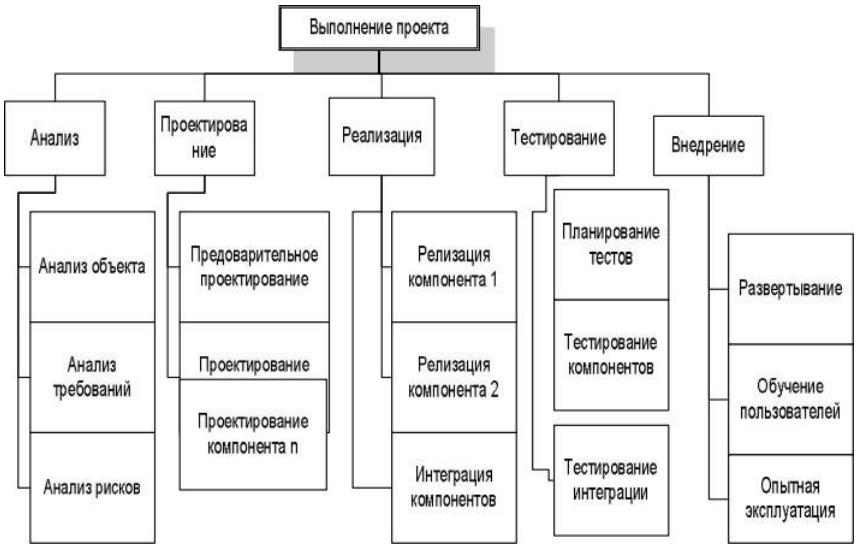

Рис. 3. Декомпозиция задач, которые необходимо решить в процессе выполнения проекта по разработке программного обеспечения

Для каждой элементарной задачи должны быть определены:

- ресурсы, необходимые для решения задачи (в том числе трудовые);
- объем работ, выраженный в принятой системе метрик;
- стоимость работ (может быть вычислена на основе объема работ и стоимости привлекаемых ресурсов);
- длительность работ (может быть вычислена на основе объема работ, количества привлекаемых рудовых ресурсов и принятых нормативов производительности).

Между отдельными элементарными задачами могут быть определенные зависимости, заключающиеся в том, что одни задачи могут выполняться параллельно, другие — в строгой последовательности (для выполнения одних задач могут требоваться результаты выполнения других).

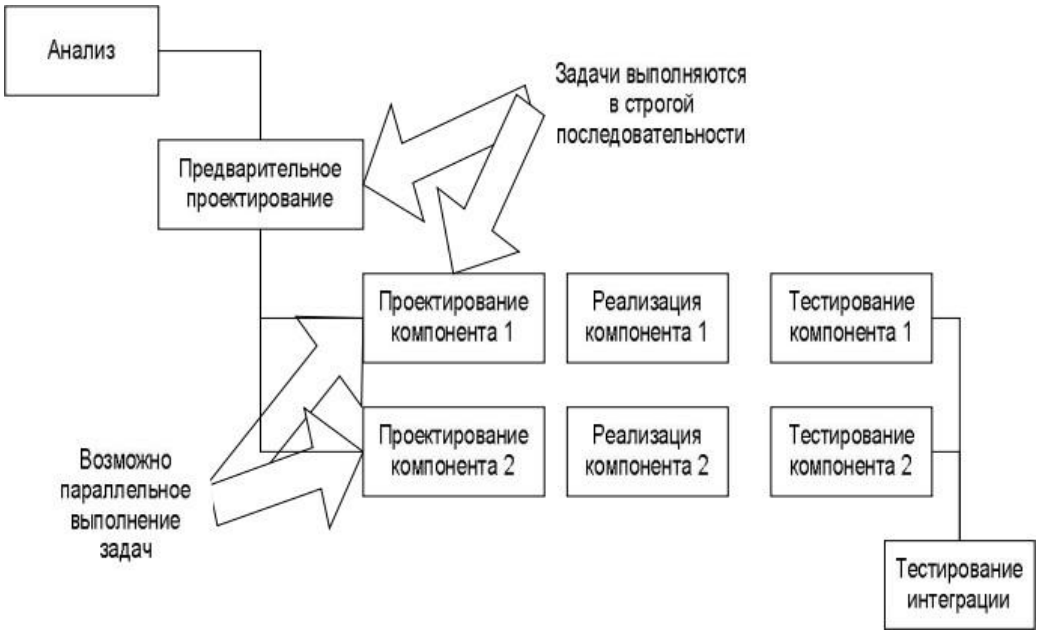

Рис. 4. Параллельное и последовательное выполнение задач

После определения зависимостей можно приступать к распределению элементарных задач по времени. При этом особое внимание следует остановить на задачах, выполняемых параллельно. Параллельность действий повышает требования к планированию.

Необходимо четко отследить наличие ресурсов, необходимых для выполнения каждой задачи. Если план предусматривает использование ресурса 1 для выполнения задач А и Б, то эти задачи не могут выполняться параллельно, даже если между ними нет концептуальной зависимости. Кроме того, руководитель проекта должен знать задачи, лежащие на критическом пути. Для того чтобы весь проект был выполнен в срок, необходимо выполнять в срок все критические задачи.

Календарный план помимо распределения задач и ресурсов по времени должен предусматривать процедуры контроля промежуточных результатов. Такие процедуры обычно называют вехами. Очень важно, чтобы вехи были расставлены через регулярные интервалы вдоль всего процесса разработки программного обеспечения. Кроме того, желательно чтобы вехи совпадали со сроком выполнения критических задач. Это даст руководителю возможность не только регулярно получать информацию о текущем положении дел, но и объективно оценивать риски срыва сроков выполнения проекта, принимать оперативные решения по снижению этих рисков. В первую очередь определите основные фазы выполнения проекта.

В основу может быть положена принятая модель жизненного цикла процесса разработки программного обеспечения. Например, при использовании каскадной модели основными фазами будут являться анализ, проектирование, реализация, тестирование, внедрение. Далее определите состав работ внутри каждой фазы, в соответствии с сутью разработки.

Таким образом будет определен состав работ по проекту. Каждая фаза должна заканчиваться вехой – специальной единицей работы, подразумевающей контроль выполнения работ по проекту и достижение некоторого промежуточного или окончательного результата. Далее определите длительность каждой работы, входящей в план работ.

Для определения длительности могут быть использованы различные регрессионные модели (например, COCOMO II), или же может применяться прямой метод оценки. Следующим этапом является определение связей между задачами. В большинстве средств планирования (например, в MS Project), существует четыре вида связей: Связь типа окончание – начало означает, что задача Б не может начаться раньше окончания задачи А (например, если в процессе выполнения задачи Б используются результаты, получаемые при решении задачи А).

Связь типа начало – начало означает, что задача Б не может начаться до тех пор, пока не началось выполнение задачи А. Например, тестирование программного компонента не может начинаться до того, как была начата его разработка, но, в то же время, для написания тестов не обязательно дожидаться окончания разработки этого компонента. Связь типа окончание – окончание означает, что работа Б не может окончиться до тех пор, пока не завершится выполнение работы А. Например, проектирование базы данных не может быть закончено до того, как будет завершено семантическое моделирование предметной области.

Связь окончание – начало означает, что задача Б не может закончиться до того, как началась задача А. Обычно такая связь используется в том случае, когда А является задачей с фиксированной датой начала, которую нельзя изменить. После того, как определен состав работ, нужно определить, кто эти задачи будет выполнять и какое оборудование должно использоваться.

Сформируйте список ресурсов, для каждого ресурса определите название и стоимость его использования. Далее назначьте ресурсы на выполнение конкретных задач. При первом назначении ресурса будут автоматически рассчитаны трудозатраты. В тех случаях, когда необходимо ускорить выполнение тех или иных задач, на них могут быть назначены дополнительные ресурсы. После распределения ресурсов необходимо выполнить выравнивание их нагрузки. В тех случаях, когда на

параллельно выполняемые задачи назначается один и тот же ресурс, нагрузка на него может превысить максимально допустимую. Для выравнивания нагрузки установите дополнительные связи между задачами таким образом, чтобы задачи, использующие один и тот же ресурс, выполнялись последовательно.

Выполнение работы:

1. Ознакомьтесь с техническим заданием
2. Выберите модель жизненного цикла процесса разработки и внедрения ПО, которая, по вашему мнению, в наибольшей степени соответствует рассматриваемой ситуации.
3. Выделите основные этапы работ.
4. Выделите основные задачи внутри отдельных этапов работ.
5. Определите зависимости между задачами.
6. Определите порядок выполнения отдельных задач.
7. Назначьте исполнителей на решаемые задачи.
8. Сбалансируйте нагрузку исполнителей.

Контрольные вопросы:

1. Опишите понятие техническое задание.
2. Что такое жизненный цикл ПО?
3. В чем разница между параллельным и последовательным выполнением задач?

Практическое занятие.

Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)

Цель занятия: изучить на практике понятия и компоненты систем контроля версий (СКВ), приемы работы с ними.

Оборудование, технические и программные средства: персональный компьютер, среда программирования Visual Studio 2019.

Задание:

1. Установите TortoiseSVN на компьютере.
2. Создайте новый проект.
3. Создайте локальный репозиторий для своего проекта.
4. Удалите созданный проект на своем компьютере и обновите проект из репозитория.
5. Внесите изменения в файлах с исходными кодами и сохраните изменения в репозитории. Обновите файлы с исходными кодами из репозитория.
6. Внесите изменения в файлах с исходными кодами таким образом, чтобы у двух участников проекта изменения были в одном и том же файле. Попытайтесь сохранить изменения в репозитории. Устраните обнаруженные конфликты версий. Повторно сохраните изменения в репозитории.
7. Создайте отдельную ветку проекта. Внесите изменения в файлы с исходными кодами. Сохраните изменения в репозитории.
8. Объедините созданную на предыдущем шаге ветку с основной веткой проекта.
9. Выведите на экран лог изменений файла, в котором было наибольшее количество изменений.
10. Отобразите на экране сравнение фала до и после внесения одного из изменений.
11. Создайте репозитарий в сети Интернет. Повторите шаги 4 – 6.

Теоретические сведения:

Управление версиями — это искусство работы с изменениями информации. Долгое время оно было жизненно важным инструментом программистов, которым необходимо произвести не-

большие изменения в программе, или же сделать ―откат‖ изменений, возвращаясь к предыдущей версии. Однако полезность систем управления версиями выходит далеко за пределы мира разработчиков программного обеспечения. Управление версиями требуется повсюду, где можно встретить людей, использующих компьютер для работы с постоянно изменяющейся информацией

Software Configuration Management или **Конфигурационное управление** подразумевает под собой комплекс методов, направленных на то, чтобы систематизировать изменения, вносимые разработчиками в программный продукт в процессе его разработки и сопровождения, сохранить целостность системы после изменений, предотвратить нежелательные и непредсказуемые эффекты, а также сделать процесс внесения изменений более формальным. Изначально управление конфигурацией применялось не в программировании, но в связи с высокой динамичностью сферы разработки ПО, в ней она особенно полезна. К процедурам можно отнести создание резервных копий, контроль исходного кода, требований проекта, документации и т. д. Степень формальности выполнения данных процедур зависит от размеров проекта, и при правильной ее оценке данная концепция может быть очень полезна. Конфигурационное управление требует выполнения множества трудоемких рутинных операций. На практике, в большинстве случаев, для конфигурационного управления применяются специальные системы контроля версий исходного кода программ. В качестве примера такой системы рассмотрим самую распространенную на сегодняшний день – Subversion.

Subversion — это бесплатная система управления версиями с открытым исходным кодом. Subversion позволяет управлять файлами и каталогами, а также сделанными в них изменениями во времени. Это позволяет восстановить более ранние версии данных, даёт возможность изучить историю всех изменений. Благодаря этому многие считают систему управления версиями своего рода «машиной времени».

МКД.01.03 Математическое моделирование

Тестирование

1. К какому классу моделей можно отнести спичечный коробок, если представить его моделью системного блока ПК при планировании своего рабочего места?
   (1) это идеальная, математическая модель
   (2) это вещественная, натурная модель
   (3) это вещественная, физическая модель
   (4) это не является моделью
2. Математическое моделирование это средство для
   (1) изучения свойств реальных объектов в рамках поставленной задачи
   (2) упрощения поставленной задачи
   (3) поиска физической модели
3. Какая из задач не имеет аналитической модели?
   (1) распознавание текста
   (2) поиск оптимального раскроя листа фанеры
   (3) демодуляция аналогового сигнала
   (4) расчет расхода топлива по заданной формуле
4. Какая математическая модель не относится к стохастическим?
   (1) идеальный газ
   (2) квантовый осциллятор
   (3) материальная точка
   (4) ни одна из предложенных

5. Инженеру во сне приснился новый шпиндель для двигателя, и он хочет его испытать, какую модель ему лучше предоставить токарям, чтобы ускорить процесс его изготовления?
   (1) идеальную, математическую
   (2) вещественную, математическую
   (3) идеальную, наглядную
   (4) вещественную, физическую
6. Какой модели быть не может?
   (1) вещественной, физической
   (2) идеальной, физической
   (3) вещественной, математической
   (4) идеальной, математической
7. Какая модель не является плодом человеческой мысли в общем случае?
   (1) математическая
   (2) наглядная
   (3) натурная
   (4) физическая
8. Материальная точка это не только математическая, но и
   (1) натурная модель
   (2) физическая модель
   (3) наглядная модель
   (4) знаковая модель
9. Математическая модель в общем случае представляется через
   (1) вектор входных переменных
   (2) вектор выходных переменных
   (3) вектор внешних воздействий
   (4) все предложенное
10. При анализе движения электронов в диодном промежутке было построено две математические модели: сперва написана программа, моделирующая взаимодействие частиц, затем выведено уравнение движения электронов из теоретических соображений. Какие математические модели были применены в данных случаях?
    (1) сперва аналитическая, затем имитационная
    (2) вначале имитационная, затем аналитическая
    (3) две аналитические
    (4) две имитационные
11. Посмотрев на набор различных математических моделей, математик сформировал четыре общих утверждения для всех математических моделей. Какое из утверждений для произвольной математической модели верно?
    (1) каждая модель может быть решена численно
    (2) каждой модели соответствует реальный объект
    (3) каждая модель имеет уравнение (систему уравнений) в явном виде
    (4) каждая модель не может при одном и том же входном параметре иметь несколько различных решений (корней)
    (5) все утверждения неверны
12. Во время поиска лучшего результата были построены две различные математические модели: эксперимент на ЭВМ, моделирующий систему атомов, и дифференциальная система уравнений, решенная численно, от двух полученных результатов взяли среднеквадратичный. Можно ли считать такой метод моделью?
    (1) да, это вещественная, математическая

(2) да, это идеальная, математическая
(3) да, это вещественная натурная
(4) нет

13. Может ли идеальный электрический контур быть моделью математического маятника?
(1) да, это случай изоморфизма
(2) да, при отсутствии консервативных сил (гомоморфизм)
(3) да, при отсутствии внешнего воздействия (гомоморфизм)
(4) Нет, так как оба случая - уже модели

14. По поведению математических моделей во времени их разделяют на
(1) детерминированные и стохастические
(2) статические и динамические
(3) непрерывные и дискретные
(4) аналитические и имитационные

15. Для того чтобы модель была гомоморфная необходимо и достаточно в рамках поставленной задачи
(1) полного соответствия между моделью и объектом
(2) соответствия наиболее значительных параметров модели и объекта
(3) какого-либо соответствия вообще между моделью и объектом
(4) любая модель гомоморфна любому объекту

16. Верно ли описание: детерминированная, непрерывная, аналитическая, модель?
(1) нет, т.к. модель не может быть детерминированной и непрерывной одновременно
(2) нет, т.к. любая аналитическая модель уже является непрерывной
(3) да

17. Как называется замещаемый моделью объект?
(1) копия
(2) оригинал
(3) шаблон
(4) макет

18. Какое максимальное количество моделей одного объекта можно составить?
(1) любое количество
(2) 1
(3) 3
(4) 7

19. Сколько классов моделей существует?
(1) 4
(2) 2
(3) 3

20. Какие модели относятся к классу вещественных моделей?
(1) физические
(2) идеальные
(3) наглядные
(4) натурные

21. Какие модели нельзя отнести к классу мысленных моделей?
(1) физические
(2) натурные
(3) математические
(4) наглядные

22. Какие модели входят в состав идеальных математических моделей?

(1) аналитические, функциональные, имитационные, комбинированные

(2) аналоговые, структурные, геометрические, графические, цифровые и кибернетические

(3) символы, алфавит, языки программирования, упорядоченная запись, топологическая запись, сетевое представление

23. Что такое математическая модель?

(1) точное представление реальных объектов, процессов или систем, выраженное в математических терминах и сохраняющее существенные черты оригинала

(2) точное представление реальных объектов, процессов или систем, выраженное в физических терминах и сохраняющее существенные черты оригинала

(3) приближенное представление реальных объектов, процессов или систем, выраженное в математических терминах и сохраняющее существенные черты оригинала

(4) приближенное представление реальных объектов, процессов или систем, выраженное в физических терминах и сохраняющее существенные черты оригинала

24. Чем является функционал "X" в представлении математической модели в виде системы функционалов Φ_i (X,Y,Z,t)=0?

(1) вектором входных переменных

(2) вектором выходных переменных

(3) вектором внешних воздействий

(4) координатой времени

25. В чем заключается построение математической модели?

(1) в определении связей между теми или иными процессами и явлениями, создании математического аппарата, позволяющего выразить количественно и качественно связь между теми или иными процессами и явлениями, между интересующими специалиста математическими величинами, и факторами, влияющими на конечный результат

(2) в определении связей между теми или иными процессами и явлениями, создании математического аппарата, позволяющего выразить количественно связь между теми или иными процессами и явлениями, между интересующими специалиста физическими величинами, и факторами, влияющими на конечный результат

(3) в определении связей между теми или иными процессами и явлениями, создании математического аппарата, позволяющего выразить количественно связь между теми или иными процессами и явлениями, между интересующими специалиста математическими величинами, и факторами, влияющими на конечный результат

(4) в определении связей между теми или иными процессами и явлениями, создании математического аппарата, позволяющего выразить количественно и качественно связь между теми или иными процессами и явлениями, между интересующими специалиста физическими величинами, и факторами, влияющими на конечный результат

26. Какие виды математических моделей получаются при разделении их по принципам построения?

(1) аналитические

(2) детерминированные

(3) стохастические

(4) имитационные

27. В зависимости от характера исследуемых реальных процессов и систем, на какие группы могут быть разделены математические модели?

(1) непрерывные

(2) детерминированные

(3) имитационные

(4) стохастические

28. Какие группы математических моделей не являются результатом распределения моделей по их поведению во времени?

(1) статические

(2) динамические

(3) изоморфные

(4) непрерывные

29. На какие группы можно разделить математические модели по виду входной информации?

(1) статические

(2) дискретные

(3) непрерывные

(4) динамические

30. На какие группы можно разделить математические модели по степени их соответствия реальным объектам, процессам или системам?

(1) стохастические

(2) изоморфные

(3) детерминированные

(4) гомоморфные

31. Как называется модель, если между ней и реальным объектом, процессом или системой существует полное поэлементное соответствие?

(1) стохастическая

(2) изоморфная

(3) детерминированная

(4) гомоморфная

32. Как называются модели, в которых предполагается отсутствие всяких случайных воздействий и их элементы (элементы модели) достаточно точно установлены?

(1) статические

(2) дискретные

(3) детерминированные

(4) динамические

33. В каком моделировании функционирование объектов, процессов или систем описывается набором алгоритмов?

(1) аппроксимационном

(2) имитационном

(3) аналитическом

34. Что означает сокращенное обозначение модели СДА?

(1) стохастическая, дискретная, аналитическая

(2) стохастическая, детерминированная, аналитическая

(3) дискретная, стохастическая, аналитическая