ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сведения о сертификате ЭП
Сертификат: 1ea208e57bd5c15d3d0fdd330db1ae60
Владелец: Соколов Максим Александрович
Действителен: с 06.08.2026 по 30.10.2027

УТВЕРЖДЕНО:

Ученым советом Института сервисных технологий
Протокол № 24 от «16» января 2025 г.

РАБОЧАЯ ПРОГРАММА
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ

основной профессиональной образовательной программы среднего профессионального образования – программы подготовки специалистов среднего звена

по специальности: ***09.02.07 Информационные системы и программирование***

Квалификация: ***Специалист по информационным системам***

год начала подготовки: 2025

Разработчики:

должность	ученая степень и звание, ФИО
преподаватель	***Границына М.С***

Рабочая программа согласована и одобрена руководителем ППССЗ:

должность	ученая степень и звание, ФИО
преподаватель	***Границына М.С***

Рабочая программа согласована и одобрена представителем работодателей:

должность	должность, ФИО
главный специалист отдела по защите информации	***Милосердов М.А***

СОДЕРЖАНИЕ

1 Общая характеристика рабочей программы профессионального модуля

2 Структура и содержание профессионального модуля

3 **Методические указания по проведению практических занятий/лабораторных работ/семинаров, занятий в форме практической подготовки (при наличии), и самостоятельной работе**

4 **Фонд оценочных средств профессионального модуля**

5 Фонд оценочных средств для аттестации по модулю

6 Условия реализации профессионального модуля

7 Информационное обеспечение реализации программы профессионального модуля

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Осуществление интеграции программных модулей

1.1. Цель и планируемые результаты освоения профессионального модуля

В результате изучения профессионального модуля обучающийся должен освоить основной вид деятельности Осуществление интеграции программных модулей и соответствующие ему общие компетенции и профессиональные компетенции:

1.1.1. Перечень общих компетенций[1]

Код	Наименование общих компетенций
ОК 1.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 2.	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде.
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языке

1.1.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций

[1] *В данном подразделе указываются только те компетенции и личностные результаты, которые формируются в рамках данного модуля и результаты которых будут оцениваться в рамках оценочных процедур по модулю.* Личностные результаты переносятся из Приложения 3 ПООП.

ВД 2	Осуществление интеграции программных модулей
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

1.1.3. В результате освоения профессионального модуля обучающийся должен[2]:

иметь практический опыт в:

- интеграции модулей в программное обеспечение;
- отладке программных модулей.

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

1.2. Количество часов, отводимое на освоение профессионального модуля

Всего часов **478**

в том числе в форме практической подготовки ***180***

Из них на освоение МДК **440**

в том числе самостоятельная работа **26**

практики, в том числе учебная **72**

производственная **108**

Промежуточная аттестация:

дифференцированный зачет по междисциплинарным курсам (4 семестр)

Аттестация по модулю (экзамен) - **12**

[2] *Берутся сведения, указанные по данному виду деятельности в п. 4.2.*

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2.1. Структура профессионального модуля

Коды профессиональных, общих компетенций	**Наименования разделов профессионального модуля***	**Всего часов** *(макс. учебная нагрузка и практики)*	**Объем времени, отведенный на освоение междисциплинарного курса (курсов)**						**Практика**	
			Обязательная аудиторная учебная нагрузка обучающегося				**Самостоятельная работа обучающегося**		**Учебная практика,** часов	**Производственная (по профилю специальности),** часов *(если предусмотрена рассредоточенная практика)*
			Всего, часов	**Лекции**	**в т.ч. лабораторные работы и практические занятия,** часов	**в т.ч., курсовая работа (проект),** часов	**Всего,** часов	**в т.ч., курсовая работа (проект),** часов		
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
ПК2.1, ПК2.4, ПК 2.5 ОК 01- ОК 09	**Раздел 1.** Технология разработки программного обеспечения	**76**	**70**	35	**35**		**6**		*	*
ПК 2.2, ПК 2.3, ПК 2.5 ОК 01- ОК 09	**Раздел 2.** Инструментальные средства разработки программного обеспечения	**132**	**114**	38	**38**	**38**	**18**	**10**	*	*
ПК 2.1, ПК 2.4, ПК 2.5 ОК 01- ОК 09	**Раздел 3.** Математическое моделирование	**78**	**76**	38	**38**		**2**		*	*
ПК 2.1-ПК 2.5 ОК 01- ОК 09	Учебная практика	**72**							**72**	
ПК 2.1-ПК 2.5 ОК 01- ОК 09	Производственная практика (по профилю специальности)	**108**								**108**

* Раздел профессионального модуля – часть программы профессионального модуля, которая характеризуется логической завершенностью и направлена на освоение одной или нескольких профессиональных компетенций. Раздел профессионального модуля может состоять из междисциплинарного курса или его части и соответствующих частей учебной и производственной практик. Наименование раздела профессионального модуля должно начинаться с отглагольного существительного и отражать совокупность осваиваемых компетенций, умений и знаний.

ПК 2.1-ПК 2.5 ОК 01- ОК 09	Аттестация по модулю (экзамен)	**12**								
	Всего:	**478**	**260**	**111**	**111**	**38**	**26**		**72**	**108**

2.2. Содержание обучения по профессиональному модулю (ПМ)

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
ПМ02 Осуществление интеграции программных модулей			
Раздел 1 Разработка программного обеспечения			
МДК 02.01. Технология разработки программного обеспечения			
Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению	**Содержание лекций** 1. Понятия требований, классификация, уровни требований. Методологии и стандарты, регламентирующие работу с требованиями. 2. Современные принципы и методы разработки программных приложений. 3. Методы организации работы в команде разработчиков. Системы контроля версий 4. Основные подходы к интегрированию программных модулей. 5. Стандарты кодирования.	13	ОК 01 ОК 02 ОК 06 ОК 07 ОК 08 ПК 2.1 ПК 2.2 ПК 2.3 ПК 2.4 ПК 2.5
	Практические занятия 1. Анализ предметной области	11	

	2. Разработка и оформление технического задания 3. Построение архитектуры программного средства 4. Изучение работы в системе контроля версий		
Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF	**Содержание лекций** 1. Описание требований: унифицированный язык моделирования - краткий словарь. Диаграммы UML. 2. Описание и оформление требований (спецификация). Анализ требований и стратегии выбора решения	12	ПК 2.1 ПК 2.2 ОК 01 ОК 02 ОК 09
	Практические занятия 1. Построение диаграммы Вариантов использования и диаграммы. Последовательности 2. Построение диаграммы Кооперации и диаграммы Развертывания 3. Построение диаграммы Деятельности, диаграммы Состояний и диаграммы Классов 4. Построение диаграммы компонентов 5. Построение диаграмм потоков данных	12	
Тема 2.1.3. Оценка качества программных средств	**Содержание лекций** 1. Цели и задачи и виды тестирования. Стандарты качества программной документации. Меры и метрики. 2. Тестовое покрытие. 3. Тестовый сценарий, тестовый пакет. 4. Анализ спецификаций. Верификация и аттестация программного обеспечения.	10	ОК 01 ОК 02 ОК 03 ОК 04
	Практические занятия 1. Разработка тестового сценария 2. Оценка необходимого количества тестов 3. Разработка тестовых пакетов 4. Оценка программных средств с помощью метрик 5. Инспекция программного кода на предмет соответствия стандартам кодирования	12	

Раздел 2. Средства разработки программного обеспечения			
МДК 02.02. ***Инструментальные средства разработки программного обеспечения***			
Тема 2.2.1 Современные технологии и инструменты интеграции.	**Содержание лекций** 1. Понятие репозитория проекта, структура проекта. 2. Виды, цели и уровни интеграции программных модулей. Автоматизация бизнес-процессов. 3. Выбор источников и приемников данных, сопоставление объектов данных. 4. Транспортные протоколы. Стандарты форматирования сообщений. 5. Организация работы команды в системе контроля версий.	20	ОК 01 ОК 02 ОК 03 ОК 04 ПК 2.1 ПК 2.2 ПК 2.3
	Практические занятия 1. Разработка структуры проекта 2. Разработка модульной структуры проекта (диаграммы модулей) 3. Разработка перечня артефактов и протоколов проекта 4. Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий) 5. Разработка и интеграция модулей проекта (командная работа) 6. Отладка отдельных модулей программного проекта 7. Организация обработки исключений	20	
Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств	**Содержание лекций** 1. Отладка программных продуктов. Инструменты отладки. Отладочные классы. 2. Ручное и автоматизированное тестирование. Методы и средства организации тестирования. 3. Инструментарии анализа качества программных продуктов в среде разработке. 4. Обработка исключительных ситуаций. Методы и способы идентификации сбоев и ошибок.	18	

	5. Выявление ошибок системных компонентов.		
	Практические занятия 1. Применение отладочных классов в проекте 2. Отладка проекта 3. Инспекция кода модулей проекта 4. Тестирование интерфейса пользователя средствами инструментальной среды разработки 5. Разработка тестовых модулей проекта для тестирования отдельных модулей 6. Выполнение функционального тестирования 7. Тестирование интеграции 8. Документирование результатов тестирования	18	
Раздел 3. Моделирование в программных системах			
МДК 02.03. ***Математическое моделирование***			
Тема 2.3.1. Основы моделирования. Детерминированные задачи	**Содержание лекций**	20	ПК 2.1 ПК 2.2 ПК 2.3 ОК 01 ОК 02 ОК 03 ОК 04
	1. Понятие решения. Множество решений, оптимальное решение. Показатель эффективности решения 2. Математические модели, принципы их построения, виды моделей. 3. Задачи: классификация, методы решения, граничные условия. 4. Общий вид и основная задача линейного программирования. Симплекс - метод. 5. Транспортная задача. Методы нахождения начального решения транспортной задачи. Метод потенциалов. 6. Общий вид задач нелинейного программирования. Графический метод решения задач нелинейного программирования. Метод множителей Лагранжа. 7. Основные понятия динамического программирования: шаговое управление, управление операцией в целом, оптимальное управление, выигрыш на данном шаге, выигрыш за всю операцию, аддитивный критерий, мультипликативный критерий.		

	8. Простейшие задачи, решаемые методом динамического программирования. 9. Методы хранения графов в памяти ЭВМ. Задача о нахождении кратчайших путей в графе и методы ее решения. 10 Задача о максимальном потоке и алгоритм Форда-Фалкерсона.		
	Практические занятия 1. Построение простейших математических моделей. Построение простейших статистических моделей 2. Решение простейших однокритериальных задач 3. Задача Коши для уравнения теплопроводности 4. Сведение произвольной задачи линейного программирования к основной задаче линейного программирования 5. Решение задач линейного программирования симплекс- методом 6. Нахождение начального решения транспортной задачи. Решение транспортной задачи методом потенциалов 7. Применение метода стрельбы для решения линейной краевой задачи 8. Задача о распределении средств между предприятиями 9. Задача о замене оборудования 10. Нахождение кратчайших путей в графе. Решение задачи о максимальном потоке	20	
Тема 2.3.2 Задачи в условиях неопределенности	**Содержание лекций**	18	ОК 01 ОК 02 ОК 06 ОК 07 ОК 08 ПК 2.1 ПК 2.2 ПК 2.3 ПК 2.4 ПК 2.5
	1. Системы массового обслуживания: понятия, примеры, модели. 2. Основные понятия теории марковских процессов: случайный процесс, марковский процесс, граф состояний, поток событий, вероятность состояния, уравнения Колмогорова, финальные вероятности состояний. 3. Схема гибели и размножения. 4. Метод имитационного моделирования. Единичный жребий и формы его организации. Примеры задач 5. Понятие прогноза. Количественные методы прогнозирования: скользящие средние, экспоненциальное сглаживание, проектирование тренда. Качественные		

	методы прогноза 6. Предмет и задачи теории игр. Основные понятия теории игр: игра, игроки, партия, выигрыш, проигрыш, ход, личные и случайные ходы, стратегические игры, стратегия, оптимальная стратегия. 7. Антагонистические матричные игры: чистые и смешанные стратегии. 8. Методы решения конечных игр: сведение игры mxn к задаче линейного программирования, численный метод - метод итераций. 9. Область применимости теории принятия решений. Принятие решений в условиях определенности, в условиях риска, в условиях неопределенности. 10. Критерии принятия решений в условиях неопределенности. Дерево решений.		
	Практические занятия 1. Составление систем уравнений Колмогорова. Нахождение финальных вероятностей. Нахождение характеристик простейших систем массового обслуживания. 2. Решение задач массового обслуживания методами имитационного моделирования 3. Построение прогнозов 4. Решение матричной игры методом итераций 5. Моделирование прогноза 6. Выбор оптимального решения с помощью дерева решений	18	
Самостоятельная работа при изучении ПМ.02. Самостоятельная работа выполняется в форме систематической проработки конспектов занятий, учебной и специальной технической литературы (по вопросам к параграфам, главам учебных пособий, составленным преподавателем), подготовки к практическим занятиям с использованием методических рекомендаций преподавателя..		26	
Тематика внеаудиторной самостоятельной работы: Составить краткую таблицу для отображения всех моделей жизненного цикла программного обеспечения и указанием их достоинств и недостатков Разработать техническое задание на разработку ИС согласно варианту			
Самостоятельная работа по курсовому проекту			

Обязательная аудиторная учебная нагрузка по курсовому проекту при изучении МДК 02.02.	38	
Примерная тематика курсовых проектов 1. Разработка ИС «Книжный магазин» 2. Разработка ИС «Студенческая библиотека» 3. Разработка ИС «Отдел кадров предприятия» 4. Разработка ИС «Абитуриент» 5. Разработка ИС «Телефонный справочник» 6. Разработка ИС для студентов колледжа 7. Разработка ИС больницы 8. Разработка ИС гостиницы 9. Разработка ИС фармацевтической компании 10. Разработка ИС салона по производству мебели 11. Разработка ИС оптового склада 12. Разработка ИС компьютерного салона 13. Разработка ИС регистратуры поликлиники 14. Разработка ИС автосалона 15. Разработка ИС агентства недвижимости		
Учебная практика **Виды работ** Ознакомление студентов с программным обеспечением учебного заведения: внедрением автоматизированных систем управления, развитием и распространением автоматизированных систем управления, перспективами развития автоматизации в области электронных вычислительных машин. Изучение программных продуктов учебного заведения: «первичные» программы для обработки информации в организации, внедрение новых программных продуктов специализирующихся на конкретное учебное заведение, способы работы с конкретным программным обеспечением, имеющимся в организации. Разработка программного обеспечения, которое будет направлено на автоматизацию специализированных информационных потоков в организации Тестирование разработанного программного продукта Оформление технической и программной документации	**72**	
Производственная практика (по профилю специальности)	**108**	

Виды работ Ознакомление студентов с программным обеспечением информационного центра предприятия: внедрением автоматизированных систем управления, развитием и распространением автоматизированных систем управления, перспективами развития автоматизации в области электронных вычислительных машин. Изучение программных продуктов предприятия: «первичные» программы для обработки информации на предприятии, внедрение новых программных продуктов специализирующихся на конкретное производство, способы работы с конкретным программным обеспечением, имеющимся на предприятии. Разработать программное обеспечение, которое будет направлено на автоматизацию специализированных информационных потоков на предприятии Провести тестирование разработанного программного продукта Для разработанного программного обеспечения оформить техническую и программную документацию		
Промежуточная аттестация	**12**	
Всего	**478**	

3. Методические указания по проведению практических занятий/лабораторных работ/семинаров, занятий в форме практической подготовки (при наличии), и самостоятельной работе

Практические занятия заключаются в выполнении студентами, под руководством преподавателя, комплекса учебных заданий направленных на усвоение научно-теоретических основ учебной дисциплины, приобретение практических навыков овладения методами практической работы с применением современных средств компьютерной графики, мультимедиа, коммуникационных технологий.

Практические занятия способствуют более глубокому пониманию теоретического материала учебного курса, а также развитию, формированию и становлению различных уровней составляющих профессиональной компетентности студентов. Наряду с формированием умений и навыков в процессе практических занятий обобщаются, систематизируются, углубляются и конкретизируются теоретические знания, вырабатывается способность и готовность использовать эти навыки на практике, развиваются интеллектуальные умения.

Практические занятия проводятся в форме практических работ.

3.1. Тематика и содержание практических занятий/лабораторных работ/ семинаров

Раздел 1 Разработка программного обеспечения

МДК.02.01. Технология разработки программного обеспечения

Тема 2.1.1. Основные понятия и стандартизация требований к программному обеспечению
Практические занятия

1. Анализ предметной области
2. Разработка и оформление технического задания
3. Построение архитектуры программного средства
4. Изучение работы в системе контроля версий

Тема 2.1. 2. Описание и анализ требований. Диаграммы IDEF

Практические занятия

1. Построение диаграммы Вариантов использования и диаграммы. Последовательности
2. Построение диаграммы Кооперации и диаграммы Развертывания
3. Построение диаграммы Деятельности, диаграммы Состояний и диаграммы Классов
4. Построение диаграммы компонентов
5. Построение диаграмм потоков данных

Тема 2.1.3. Оценка качества программных средств

Практические занятия

1. Разработка тестового сценария
2. Оценка необходимого количества тестов

3. Разработка тестовых пакетов
4. Оценка программных средств с помощью метрик
5. Инспекция программного кода на предмет соответствия стандартам кодирования

Раздел 1. Результаты обучения (умения):

Анализировать проектную и техническую документацию.
Использовать специализированные графические средства построения и анализа архитектуры программных продуктов.
Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов.
Определять источники и приемники данных.
Проводить сравнительный анализ. Выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace).
Оценивать размер минимального набора тестов.
Разрабатывать тестовые пакеты и тестовые сценарии.
Выявлять ошибки в системных компонентах на основе спецификаций.
Использовать выбранную систему контроля версий.
Выполнять тестирование интеграции.
Организовывать постобработку данных.
Использовать приемы работы в системах контроля версий.
Разрабатывать тестовые пакеты и тестовые сценарии.
Выполнять ручное и автоматизированное тестирование программного модуля.
Использовать методы для получения кода с заданной функциональностью и степенью качества.
Приемы работы в системах контроля версий.

Раздел 2. Средства разработки программного обеспечения

МДК 02.02. Инструментальные средства разработки программного обеспечения

Тема 2.2.1 Современные технологии и инструменты интеграции

Практические занятия

1. Разработка структуры проекта
2. Разработка модульной структуры проекта (диаграммы модулей)
3. Разработка перечня артефактов и протоколов проекта
4. Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)
5. Разработка и интеграция модулей проекта (командная работа)
6. Отладка отдельных модулей программного проекта
7. Организация обработки исключений

Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств

Практические занятия

1. Применение отладочных классов в проекте

2. Отладка проекта
3. Инспекция кода модулей проекта
4. Тестирование интерфейса пользователя средствами инструментальной среды разработки
5. Разработка тестовых модулей проекта для тестирования отдельных модулей
6. Выполнение функционального тестирования
7. Тестирование интеграции
8. Документирование результатов тестирования

Раздел 2. Результаты обучения (умения):

Использовать выбранную систему контроля версий.
Использовать методы для получения кода с заданной функциональностью и степенью качества.
Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов.
Использовать различные транспортные протоколы и стандарты форматирования сообщений.
Выполнять тестирование интеграции.
Организовывать постобработку данных.
Создавать классы- исключения на основе базовых классов.
Выполнять ручное и автоматизированное тестирование программного модуля.
Выявлять ошибки в системных компонентах на основе спецификаций.
Использовать приемы работы в системах контроля версий.
Анализировать проектную и техническую документацию.
Использовать инструментальные средства отладки программных продуктов.
Определять источники и приемники данных.
Выполнять тестирование интеграции.
Использовать приемы работы в системах контроля версий.
Выполнять отладку, используя методы и инструменты условной компиляции.
Приемы работы в системах контроля версий.

Раздел 3. Моделирование в программных системах

МДК 02.03. Математическое моделирование

Тема 2.3.1. Основы моделирования. Детерминированные задачи

Практические занятия
1. Построение простейших математических моделей. Построение простейших статистических моделей
2. Решение простейших однокритериальных задач
3. Задача Коши для уравнения теплопроводности
4. Сведение произвольной задачи линейного программирования к основной задаче линейного программирования
5. Решение задач линейного программирования симплекс- методом
6. Нахождение начального решения транспортной задачи. Решение транспортной задачи методом потенциалов

7. Применение метода стрельбы для решения линейной краевой задачи
8. Задача о распределении средств между предприятиями
9. Задача о замене оборудования
10. Нахождение кратчайших путей в графе. Решение задачи о максимальном потоке

Тема 2.3.2 Задачи в условиях неопределенности

Практические занятия
1. Составление систем уравнений Колмогорова. Нахождение финальных вероятностей. Нахождение характеристик простейших систем массового обслуживания.
2. Решение задач массового обслуживания методами имитационного моделирования
3. Построение прогнозов
4. Решение матричной игры методом итераций
5. Моделирование прогноза
6. Выбор оптимального решения с помощью дерева решений

Раздел 3. Результаты обучения (умения):

Анализировать проектную и техническую документацию.
Использовать специализированные графические средства построения и анализа архитектуры программных продуктов.
Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов.
Определять источники и приемники данных.
Проводить сравнительный анализ. Выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace).
Оценивать размер минимального набора тестов.
Разрабатывать тестовые пакеты и тестовые сценарии.
Выявлять ошибки в системных компонентах на основе спецификаций.
Использовать выбранную систему контроля версий.
Выполнять тестирование интеграции.
Организовывать постобработку данных.
Использовать приемы работы в системах контроля версий.
Разрабатывать тестовые пакеты и тестовые сценарии.
Выполнять ручное и автоматизированное тестирование программного модуля.
Использовать методы для получения кода с заданной функциональностью и степенью качества.
Приемы работы в системах контроля версий.

3.2. Тематика и содержание самостоятельной работы

Самостоятельная работа является неотъемлемой частью образовательного процесса, связанного с формированием компетенций обучающихся.

Целью самостоятельной (внеаудиторной) работы студентов является обучение навыкам работы с научно-теоретической, периодической, научно-технической литературой и технической документацией, необходимыми для углубленного изучения дисциплины, а также развитие у них устойчивых способностей к самостоятельному изучению и изложению полученной информации.

4.Фонд оценочных средств профессионального модуля

4.1. Формы аттестации по профессиональному модулю

Элемент модуля	**Форма контроля и оценивания**	
	Промежуточная аттестация	**Текущий контроль**
МДК.02.01 Технология разработки программного обеспечения	комплексный дифференцированный зачет в 4 семестре	Оценка выполнения практических работ, оценка выполнения самостоятельных работ, устный опрос
МДК.02.02 Инструментальные средства разработки программного обеспечения		
МДК.02.03 Математическое моделирование	Другая форма контроля - 4 семестр	
УП.02.01Учебная практика	дифференцированный зачет - 4 семестр	Оценка выполнения практических работ. Выполнение отчета.
ПП.02.01Производственная практика (по профилю специальности)	дифференцированный зачет - 4 семестр	Оценка выполнения практических работ. Выполнение отчета.
Экзамен по модулю в 4 семестре		

4.2. Результаты освоения профессионального модуля

Профессиональные, общие компетенции, личностные результаты

В результате контроля и оценки по профессиональному модулю осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Код	**Наименование результата обучения**
ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.	**Практический опыт:** Разрабатывать и оформлять требования к программным модулям по предложенной документации. Разрабатывать тестовые наборы (пакеты) для программного модуля. Разрабатывать тестовые сценарии программного средства. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования. **Умения:** Анализировать проектную и техническую документацию. Использовать специализированные графические средства построения и анализа **Знания:** Модели процесса разработки программного обеспечения.

	Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Виды и варианты интеграционных решений. Современные технологии и инструменты интеграции. Основные протоколы доступа к данным. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы отладочных классов. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Графические средства проектирования архитектуры программных продуктов. Методы организации работы в команде разработчиков.
ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.	**Практический опыт:** Интегрировать модули в программное обеспечение. Отлаживать программные модули. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования. **Умения:** Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. Использовать различные транспортные протоколы и стандарты форматирования сообщений. Выполнять тестирование интеграции. Организовывать постобработку данных. Создавать классы- исключения на основе базовых классов. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций. Использовать приемы работы в системах контроля версий. **Знания:** Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации программного обеспечения. Современные технологии и инструменты интеграции. Основные протоколы доступа к данным. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Основные методы отладки. Методы и схемы обработки исключительных ситуаций. Основные методы и виды тестирования программных продуктов.

	Стандарты качества программной документации. Основы организации инспектирования и верификации. Приемы работы с инструментальными средствами тестирования и отладки. Методы организации работы в команде разработчиков.
ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.	**Практический опыт:** Отлаживать программные модули. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования. **Умения:** Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Использовать инструментальные средства отладки программных продуктов. Определять источники и приемники данных. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Выполнять отладку, используя методы и инструменты условной компиляции. Выявлять ошибки в системных компонентах на основе спецификаций. **Знания:** Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Основные методы отладки. Методы и схемы обработки исключительных ситуаций. Приемы работы с инструментальными средствами тестирования и отладки. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков.
ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.	**Практический опыт:** Разрабатывать тестовые наборы (пакеты) для программного модуля. Разрабатывать тестовые сценарии программного средства. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования. **Умения:** Использовать выбранную систему контроля версий. Анализировать проектную и техническую документацию. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах

	контроля версий. Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций. **Знания:** Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы и схемы обработки исключительных ситуаций. Основные методы и виды тестирования программных продуктов. Приемы работы с инструментальными средствами тестирования и отладки. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков.
ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	**Практический опыт:** Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования. **Умения:** Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Организовывать постобработку данных. Приемы работы в системах контроля версий. Выявлять ошибки в системных компонентах на основе спецификаций. **Знания:** Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков.
ОК 01. Выбирать способы решения задач профес-	**Умения:** распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части;

сиональной деятельности, применительно к различным контекстам	определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника) **Знания:** актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.	**Умения:** определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска **Знания:** номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях.	**Умения**: определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования **Знания:** содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования
ОК 04. Эффективно взаимодействовать и работать в коллективе и команде. .	**Умения:** организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности **Знания:** психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности
ОК 05. Осуществлять	**Умения:** грамотно излагать свои мысли и оформлять документы

устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.	по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе **Знания:** особенности социального и культурного контекста; правила оформления документов и построения устных сообщений.
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	**Умения**: описывать значимость своей специальности **Знания:** сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях.	**Умения:** соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности **Знания:** правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной	**Умения:** использовать физкультурно-оздоровительную деятельность для укрепления здоровья, достижения жизненных и профессиональных целей; применять рациональные приемы двигательных функций в профессиональной деятельности; пользоваться средствами профилактики перенапряжения характерными для данной специальности

деятельности и поддержания необходимого уровня физической подготовленности.	**Знания:** роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.	**Умения:** понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы **Знания:** правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности

4.3. Контрольно-измерительные материалы включают:

4.3.1. Типовые задания для оценки знаний и умений текущего контроля

Контроль и оценка результатов освоения темы осуществляется преподавателем в процессе выполнения обучающимися индивидуальных заданий **в виде тестов.**

Примерные тесты:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.(включает ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9)

Тесты закрытого типа на установление последовательности

1. Задание:
   Какой из следующих этапов является первым в процессе разработки требований к программным модулям?
   A) Анализ проектной документации
   B) Сбор требований от пользователей
   C) Проектирование архитектуры системы
   D) Тестирование модуля

2. Задание:
   Какой метод используется для выявления взаимодействий между компонентами системы?
   A) SWOT-анализ
   B) Моделирование потоков данных

C) Метод мозгового штурма
D) Анализ рисков

Задание закрытого типа на установление соответствия
1.

Элементы:
1. Входные данные
2. Выходные данные
3. Протокол взаимодействия
4. Интерфейс API
5. Система управления версиями

Описания:
A) Определяет, как различные модули будут обмениваться данными и командами.
B) Описывает, какие данные должны быть предоставлены модулю для его работы.
C) Указывает, какие данные модуль должен вернуть после обработки запросов.
D) Обеспечивает совместимость и управление изменениями в коде и документации.
E) Определяет набор методов и структур данных, доступных для взаимодействия с модулем.

Ответы
1. Входные данные - ___
2. Выходные данные - ___
3. Протокол взаимодействия - ___
4. Интерфейс API - ___
5. Система управления версиями - ___

2.

Элементы:
1. Функциональные требования
2. Нефункциональные требования
3. Интерфейсные требования
4. Требования к производительности
5. Требования к безопасности

Описания:
A) Описывают, как система будет реагировать на определенные действия пользователей или внешние события.
B) Определяют, как быстро система должна выполнять свои функции и обрабатывать запросы.
C) Указывают на аспекты, касающиеся защиты данных и доступа к системе.
D) Определяют общие характеристики системы, такие как надежность и удобство использования.
E) Определяют взаимодействие между различными модулями и компонентами системы.

Ответы
1. Функциональные требования - ___
2. Нефункциональные требования - ___
3. Интерфейсные требования - ___
4. Требования к производительности - ___
5. Требования к безопасности - ___

Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора

1. Какой из следующих аспектов является наиболее важным при разработке требований к программным модулям, чтобы обеспечить эффективное взаимодействие между компонентами системы?
   A) Определение языков программирования, используемых в проекте.
   B) Установление четких спецификаций интерфейсов между модулями.
   C) Выбор инструментов для тестирования программного обеспечения.
   D) Описание пользовательского интерфейса для конечных пользователей.
   Выберите один правильный ответ и обоснуйте свой выбор.

2. При разработке требований к программным модулям на основе анализа проектной и технической документации, какой из следующих аспектов следует учитывать в первую очередь для обеспечения корректного взаимодействия компонентов системы?
   A) Описание функциональных возможностей каждого модуля.
   B) Определение форматов данных, передаваемых между модулями.
   C) Установление графика разработки и тестирования модулей.
   D) Выбор методов управления версиями программного обеспечения.

   Выберите один правильный ответ и обоснуйте свой выбор.

   Задание открытого типа с развернутым ответом

1. Опишите основные этапы анализа проектной и технической документации для разработки требований к программным модулям. Как каждый из этих этапов влияет на качество итоговых требований?

2. Приведите пример формулировки функционального и нефункционального требования для программного модуля. Объясните, почему важно учитывать оба типа требований.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение .(включает ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9)

Задание закрытого типа на установление последовательности

1. Задание: Установите правильную последовательность этапов интеграции модулей в программное обеспечение. Номер этапа укажите в соответствующем поле.
   1. Тестирование интегрированных модулей
   2. Подготовка модулей к интеграции
   3. Разработка и тестирование отдельных модулей
   4. Интеграция модулей в единую систему
   5. Документирование процесса интеграции

2. Этапы интеграции модулей в программное обеспечение

   1. Разработка и тестирование отдельных модулей
   2. Подготовка модулей к интеграции
   3. Интеграция модулей в единую систему

4. Тестирование интегрированных модулей
5. Документирование процесса интеграции

Задание закрытого типа на установление соответствия

1. Этапы интеграции и их описания
Этапы:
1. Разработка модулей
2. Тестирование модулей
3. Интеграция модулей
4. Тестирование интегрированной системы
5. Документирование интеграции

Описание:
A. Проверка функциональности и выявление ошибок в каждом отдельном модуле перед их объединением.
B. Процесс объединения различных модулей в одну систему для обеспечения их взаимодействия.
C. Создание и написание кода для каждого модуля, который будет использоваться в системе.
D. Оценка работы всей системы после интеграции, чтобы убедиться, что все модули функционируют корректно вместе.
E. Запись всех этапов интеграции, включая проблемы и решения, для будущего использования и анализа.

2. Установите соответствие между этапами интеграции модулей и их описаниями
Этапы интеграции:
1. Разработка модулей
2. Тестирование модулей
3. Интеграция модулей
4. Тестирование интегрированной системы
5. Документирование интеграции
Описание этапов:
A. Процесс объединения различных модулей в одну систему для обеспечения их взаимодействия.
B. Проверка функциональности и выявление ошибок в каждом отдельном модуле перед их объединением.
C. Создание и написание кода для каждого модуля, который будет использоваться в системе.
D. Оценка работы всей системы после интеграции, чтобы убедиться, что все модули функционируют корректно вместе.
E. Запись всех этапов интеграции, включая проблемы и решения, для будущего использования и анализа.

Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора

1. Вопрос: Какой из следующих этапов является наиболее критичным для успешной интеграции модулей в программное обеспечение?
A. Разработка модулей

B. Тестирование интегрированной системы
C. Интеграция модулей
D. Документирование интеграции
Выберите один правильный ответ и обоснуйте свой выбор.

2. Какой из следующих этапов является наиболее важным при выполнении интеграции модулей в программное обеспечение?
A. Определение требований к системе
B. Разработка интерфейсов между модулями
C. Проведение юнит-тестирования отдельных модулей
D. Обучение пользователей работе с системой
Выберите один правильный ответ и обоснуйте свой выбор.

Задание комбинированного типа с выбором нескольких вариантов ответа из предложенных и развернутым обоснованием выбора

1. Какие из следующих этапов являются важными при выполнении интеграции модулей в программное обеспечение? Выберите все подходящие варианты и обоснуйте свой выбор.
A. Определение требований к системе
B. Разработка интерфейсов между модулями
C. Проведение юнит-тестирования отдельных модулей
D. Интеграционное тестирование системы
E. Обучение пользователей работе с системой
F. Документирование архитектуры системы

Задание открытого типа с развернутым ответом

1. Опишите основные этапы интеграции модулей в программное обеспечение. Укажите, какие задачи решаются на каждом этапе и почему они важны для успешной интеграции.
2. Объясните, что такое интерфейсы между модулями и какую роль они играют в процессе интеграции. Приведите примеры возможных проблем, которые могут возникнуть из-за неправильно спроектированных интерфейсов.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.(включает ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9)

Задание закрытого типа на установление последовательности

1. Установите правильную последовательность действий для подготовки к отладке программного модуля.
a) Определите цель отладки
b) Убедитесь, что у вас есть доступ к исходному коду
c) Выберите специализированное программное средство
d) Настройте окружение

2. Каковы шаги для запуска программы под отладчиком? Установите правильную последовательность.
a) Установите точки останова

b) Запустите программу под отладчиком
c) Анализируйте состояние программы
d) Исправьте ошибки

Задание закрытого типа на установление соответствия

1. Установите соответствие между терминами (колонка А) и их определениями или действиями (колонка В). Напишите номера соответствий.

1. Точка останова
2. Отладчик
3. Логирование
4. Инструмент профилирования
5. Проверка переменных

a. Утилита для анализа производительности программы
b. Инструмент для пошагового выполнения программы
c. Метод отслеживания и записи действий программы
d. Место в коде, где выполнение программы приостанавливается
e. Анализ значений переменных во время выполнения

2. Установите соответствие между терминами (колонка А) и их определениями (колонка В). Напишите номера соответствий.

1. Отладка
2. Точка останова
3. Стек вызовов
4. Логирование
5. Инструмент профилирования

a. Программа, позволяющая анализировать производительность кода
b. Процесс поиска и устранения ошибок в программе
c. Место в коде, где выполнение программы приостанавливается
d. Структура данных, содержащая информацию о вызванных функциях
e. Запись действий программы для последующего анализа

Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора

1. Какой из следующих инструментов является наиболее подходящим для выполнения отладки программного модуля, если необходимо выявить и устранить ошибки в коде?
A) Инструмент профилирования
B) Дебаггер
C) Система контроля версий

D) Тестовый фреймворк
Выберите один правильный ответ и обоснуйте свой выбор.

2. Какое специализированное программное средство наиболее эффективно для выполнения отладки программного модуля, позволяя разработчику анализировать выполнение кода и выявлять ошибки?
A) Интегрированная среда разработки (IDE)
B) Система управления проектами
C) Средство для статического анализа кода
D) Тестировщик пользовательского интерфейса

Задание комбинированного типа с выбором нескольких вариантов ответа из предложенных и развернутым обоснованием выбора

1. Какие из следующих инструментов и методов являются наиболее эффективными для выполнения отладки программного модуля с использованием специализированных программных средств? Выберите все подходящие варианты.
A) Дебаггер в интегрированной среде разработки (IDE)
B) Логирование (лог-файлы)
C) Статический анализатор кода
D) Юнит-тесты
E) Ручное тестирование пользовательского интерфейса

2. Какие действия следует предпринять после исправления ошибки в программном модуле? (Выберите ВСЕ правильные варианты)
A. Немедленно развернуть обновленный модуль в production-среде.
B. Провести повторное тестирование модуля, включая регрессионное тестирование.
C. Задокументировать исправленную ошибку и процесс ее устранения.
D. Удалить все юнит-тесты, связанные с исправленной ошибкой.
E. Обновить систему контроля версий с изменениями.
F. Сообщить об исправлении ошибки команде разработчиков.

Задание открытого типа с развернутым ответом

1. Опишите процесс отладки программного модуля, используя специализированные программные средства. Укажите ключевые этапы, методы и инструменты, которые вы бы использовали, а также объясните их значимость в процессе отладки.

2. Вы разрабатываете сложный программный модуль, включающий в себя взаимодействие с базой данных, сетевые запросы и многопоточность. Опишите, какие трудности могут возникнуть при отладке такого модуля, и какие специализированные инструменты и методы вы бы применили для выявления и устранения ошибок в каждой из этих областей. Например, как бы вы отлаживали ошибки, связанные с параллельным доступом к данным?

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.(включает ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9)

Задание закрытого типа на установление последовательности

1. Установите правильную последовательность этапов разработки тестовых наборов и тестовых сценариев. Выберите правильный порядок из предложенных вариантов.
   1. Анализ требований к программному обеспечению.
   2. Определение типов тестирования (функциональное, нефункциональное и т.д.).
   3. Разработка тестовых сценариев.
   4. Создание тестовых наборов.
   5. Выполнение тестирования.
   6. Анализ результатов тестирования и документирование.

   Варианты последовательности:
   A. 1 → 2 → 3 → 4 → 5 → 6
   B. 2 → 1 → 4 → 3 → 5 → 6
   C. 3 → 2 → 1 → 4 → 5 → 6
   D. 1 → 3 → 2 → 4 → 5 → 6

Задание закрытого типа на установление соответствия

1. Установите соответствие между этапами разработки тестовых наборов и тестовых сценариев (колонка А) и их описаниями (колонка Б). Укажите номер соответствия.

| Колонка А: Этапы разработки
1. Анализ требований

2. Определение целей тестирования

3. Разработка тестовых сценариев

4. Создание тестовых наборов

5. Выполнение тестирования

6. Анализ результатов тестирования

| Колонка Б: Описания
A. Процесс проверки соответствия фактического поведения системы ожидаемым результатам.
B. Сбор информации о функциональности и ограничениях программного обеспечения для дальнейшего тестирования.
C. Формулирование задач, которые необходимо решить в процессе тестирования, и определение критериев успешности.
D. Составление подробных инструкций по тестированию отдельных функций или компонентов ПО.
E. Объединение тестовых сценариев в группы для упрощения процесса тестирования.
F. Подведение итогов тестирования, выявление дефектов и оценка качества программного обеспечения.

Установите соответствие между этапами и их описаниями!

Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора

1. Какой из следующих этапов является наиболее критичным при разработке тестовых наборов и тестовых сценариев для программного обеспечения?

   A) Составление отчетов о тестировании
   B) Анализ требований к программному обеспечению
   C) Выполнение тестирования
   D) Обучение команды тестировщиков

2. Какой из следующих факторов наиболее важен при написании тестовых случаев?

A) Доступность ресурсов для выполнения тестирования

B) Четкость и понятность формулировок

C) Количество тестируемых функций

D) Уровень сложности программного обеспечения

Задание комбинированного типа с выбором нескольких вариантов ответа из предложенных и развернутым обоснованием выбора

1. Какие из следующих утверждений верны в контексте разработки тестовых наборов и тестовых сценариев?
   1. Тестовые сценарии должны быть независимыми друг от друга.
   2. Тестовые наборы должны охватывать все возможные сценарии использования программного обеспечения.
   3. Разработка тестовых сценариев должна основываться на требованиях и спецификациях продукта.
   4. Тестовые случаи не должны включать ожидаемые результаты.
   5. Тестовые наборы могут включать как автоматизированные, так и ручные тесты.

2. Какие инструменты могут быть использованы для автоматизации тестирования программного обеспечения? (Выберите ВСЕ правильные варианты)
   A. Текстовый редактор, например, Notepad++.
   B. Selenium.
   C. JUnit/TestNG (для Java).
   D. Postman.
   E. Git.
   F. Jira.

Задание открытого типа с развернутым ответом

1. Объясните разницу между тестированием “белого ящика” (white-box testing) и тестированием “черного ящика” (black-box testing). Приведите примеры техник, которые используются в каждом типе тестирования. Какие преимущества и недостатки у каждого из этих подходов? В каких ситуациях следует применять каждый из этих подходов?

2. Опишите процесс разработки тестового набора (test suite) для тестирования веб-приложения электронной коммерции (e-commerce). Какие типы тестовых сценариев необходимо включить в этот набор? Какие критерии следует использовать для организации и приоритизации тестовых сценариев?

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.(включает ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9)

Задание закрытого типа на установление последовательности

1. Установите правильный порядок действий:

   1. Подготовка к инспектированию

   2. Сбор и анализ документации

   3. Проведение инспекции кода

   4. Запись замечаний и рекомендаций

   5. Обсуждение результатов инспекции с командой

   6. Корректировка кода на основе замечаний

   7. Повторная проверка исправленного кода

2. Установите правильную последовательность шагов процесса инспектирования компонентов программного обеспечения на предмет соответствия стандартам кодирования.

   Шаги:

   1. Определение стандартов кодирования.

   2. Подготовка к инспектированию.

   3. Проведение инспекции.

   4. Документирование результатов.

   5. Исправление выявленных недостатков.

   6. Повторная проверка исправлений.

   Варианты ответа:
   A) 1 → 2 → 3 → 4 → 5 → 6
   B) 2 → 1 → 3 → 4 → 5 → 6
   C) 1 → 3 → 2 → 4 → 5 → 6
   D) 1 → 2 → 4 → 3 → 5 → 6

Задание закрытого типа на установление соответствия

1. Какова основная цель инспектирования кода?

   A) Увеличение объема кода
   B) Выявление ошибок и несоответствий стандартам кодирования
   C) Ускорение процесса разработки
   D) Улучшение пользовательского интерфейса

2. Какой из следующих шагов не является частью процесса инспектирования?

   A) Подготовка к инспекции
   B) Проведение тестирования
   C) Обсуждение результатов инспекции
   D) Запись замечаний

Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора

1. Какое из следующих утверждений наиболее точно описывает важность инспектирования кода на предмет соответствия стандартам кодирования?

   A) Инспектирование кода позволяет увеличить объем документации, что всегда полезно для команды.
   B) Инспектирование кода помогает выявить потенциальные ошибки и улучшить качество программного обеспечения, что снижает затраты на исправление ошибок в будущем.
   C) Инспектирование кода не имеет значительного влияния на качество, так как большинство ошибок выявляется на этапе тестирования.
   D) Инспектирование кода является формальной процедурой, которая не требует участия команды разработки.

2. Что является наиболее важной целью инспектирования компонент программного обеспечения на предмет соответствия стандартам кодирования?

A. Выявление и исправление всех синтаксических ошибок в коде.
B. Улучшение читаемости, поддерживаемости и общего качества кода.
C. Оптимизация производительности программного обеспечения.
D. Обеспечение соответствия кода требованиям безопасности.

Задание комбинированного типа с выбором нескольких вариантов ответа из предложенных и развернутым обоснованием выбора

1. Какие из следующих утверждений о процессе инспектирования кода на предмет соответствия стандартам кодирования являются верными? (Выберите все подходящие варианты)

   A) Инспектирование кода помогает выявить не только синтаксические ошибки, но и логические несоответствия.
   B) Процесс инспектирования требует участия только тестировщиков, так как они наиболее квалифицированы для выявления ошибок.

C) Инспектирование кода может улучшить командное взаимодействие и обмен знаниями между разработчиками.
D) Соответствие стандартам кодирования может повысить читаемость и поддерживаемость кода.
E) Инспектирование кода является формальной процедурой, которая не требует документирования результатов.

2. Какие из перечисленных ниже действий могут помочь обеспечить соблюдение стандартов кодирования в проекте? (Выберите ВСЕ правильные варианты)

A. Внедрение автоматических проверок кода в процесс сборки (CI/CD).
B. Регулярное проведение code review.
C. Обучение разработчиков стандартам кодирования.
D. Предоставление обратной связи разработчикам о нарушениях стандартов. E. Применение штрафов к разработчикам за несоблюдение стандартов.
F. Создание культуры ответственности за качество кода.

Задание открытого типа с развернутым ответом

1. Какие преимущества дает регулярное инспектирование кода в команде разработчиков? Приведите примеры.
2. Опишите процесс инспектирования кода. Какие этапы он включает и каковы их основные задачи?

4.4. Требования к курсовому проекту как части аттестации

К курсовому проекту предъявляются следующие общие требования:
- логическая последовательность изложения материала;
- четкость построения;
- убедительность аргументации;
- единство формы и содержания;
- содержательность анализа и элементы исследования поставленных вопросов;
- краткость и точность формулировок, исключающих возможность субъективного и неоднозначного толкования;
- конкретность изложения результатов работы;
- доказательность выводов и обоснованность рекомендаций;
- наличие ссылок на источники используемого материала.

Поскольку курсовой проект выполняется в форме пояснительной записки, то следует выделить ряд специальных требований, предъявляемых к ней.

Содержание курсового проекта, количество и состав графической части определяется заданием на его разработку.

Введение. Это вступительная часть курсового проекта, в которой рассматриваются основные тенденции состояния и развития проблемы, обосновывается теоретическая и практическая **актуальность** проблемы, формируются **цель и задачи проекта**, **предмет и объект** курсового проекта, дается характеристика исходного состояния объекта исследования. Объем введения не должен превышать 2-3% от общего объема курсового проекта.

Цель курсового проекта представляет собой формулировку результата исследовательской деятельности и путей его достижения с помощью определенных средств. Учитывается, что у проекта может быть только одна цель.

Задачи конкретизируют цель; в соответствии с основной целью целесообразно выделить три-четыре задачи.

Задачи исследования - это теоретические и практические результаты, которые должны быть получены в курсовом проекте. Это обычно делается в форме перечисления (изучить..., установить..., выяснить..., вывести формулу и т.п.). Постановку задач следует делать как можно более тщательно, т.к. их решение составляет содержание разделов курсового проекта.

1 Проектирование ИС - часть проекта в которой необходимо дать характеристику предметной области для которой решаются задачи автоматизации соответствующих функций, проанализировать имеющиеся для решения подобных задач разработки, выбрать методы и средства проектирования. Провести разработку требований к оборудованию и системному программному обеспечению, провести концептуальное проектирование базы данных (перечень сущностей, перечень атрибутов; инфологическое проектирование: модель «сущность-связь», описание связей между сущностями; нормализация отношений), логическое моделирование предметной области (логическая модель и её описание, характеристика входной и выходной информации, создание системы таблиц, обеспечение целостности данных, запросы к базе данных), физическое моделирование предметной области (функциональная схема проекта, структурная схема проекта, описание программных модулей, схема взаимосвязи программных модулей и информационных файлов), В этом разделе описываются компоненты системы, определяется состав элементов системы, связываются конструктивные компоненты системы, проектируются несколько вариантов будущей системы, выбирается наилучший вариант для реализации с обоснованием. В тексте обязательно должны быть **ссылки на используемые источники**.

В заключение данного раздела должно быть сформулировано техническое задание на решение поставленной в теме курсового проекта задачи и определены используемые для этого технологии, методы и материалы.

В техническом задании должен быть отражен перечень средств, требующихся для реализации поставленной задачи

2 Реализация ИС – часть проекта в которой необходимо выполнить преобразование проектных решений в программную систему. В этом разделе описывают разработку каждой из подсистем, тестирование и документирование программного продукта.

Заключение – основные выводы и рекомендации, вытекающие из результатов проведенной работы.

Кратко излагаются основные результаты разработки, отмечаются оригинальные решения, полученные студентом. Приводятся основные показатели и характеристики разработанного курсового проекта, анализируется соответствие выполненной разработки заданию на курсовое проектирование.

Список использованных источников должен содержать перечень использовавшихся в ходе работы первоисточников, расположенных в алфавитном порядке и отражающих тему курсового проекта.

В **приложения** рекомендуется включать материалы, дополняющие содержание проекта, промежуточные расчеты, вспомогательные таблицы, спецификации,

нормативные документы (Устав, инструкции и т.д.), иллюстрации вспомогательного характера.

Следует помнить, что наиболее важные положения в проекте целесообразно иллюстрировать и пояснять примерами из существующих ныне разработок в аналогичных или иных, но пограничных областях деятельности.

Каждый раздел записки должен начинаться с постановки соответствующей частной задачи и завершаться выводами, отражающими полученные результаты и переход к последующим разделам.

В текст не следует помещать материалы, особенно информативного характера без какого-либо анализа. Текст должен полностью отвечать наименованиям разделов, подразделов, пунктов и подпунктов. В тексте записки не допускается: применять сокращение слов, кроме установленных правилами русской орфографии и пунктуации.

4.5. Оценка освоения теоретического курса профессионального модуля – МДК:

4.5.1 Типовые задания для оценки освоения МДК 02.01

Примерный перечень вопросов:

1. Что такое программное обеспечение (ПО)? Какие виды ПО вы знаете?
2. Опишите жизненный цикл программного обеспечения (ЖЦПО).
3. Какие основные этапы входят в процесс разработки ПО?
4. Что такое требования к программному обеспечению? Какие виды требований существуют?
5. Какие роли участвуют в процессе разработки ПО (например, заказчик, разработчик, тестировщик)?
6. Что такое методология разработки ПО?
7. Опишите waterfall (каскадную) модель разработки. Какие ее преимущества и недостатки?
8. Что такое Agile? Какие принципы лежат в основе Agile-подходов?
9. Опишите Scrum как методологию разработки. Какие роли, артефакты и события входят в Scrum?
10. Что такое Kanban? Как он отличается от Scrum?
11. Какие еще гибкие методологии разработки вы знаете (например, XP, Lean)?
12. Что такое архитектура программного обеспечения?
13. Какие основные архитектурные стили вы знаете (например, монолитная, микросервисная, клиент-серверная)?
14. Что такое UML? Какие диаграммы используются при проектировании ПО?
15. Опишите процесс создания технического задания (ТЗ) на разработку ПО.
16. Что такое паттерны проектирования? Приведите примеры (например, Singleton, Factory, Observer).
17. Опишите этап анализа требований. Какие методы сбора требований вы знаете?
18. Что такое проектирование ПО? Какие задачи решаются на этом этапе?
19. Опишите процесс кодирования (реализации) ПО. Какие принципы программирования важно соблюдать?
20. Что такое тестирование ПО? Какие виды тестирования вы знаете (например, модульное, интеграционное, системное)?
21. Опишите этап внедрения и сопровождения ПО.
22. Какие инструменты используются для управления проектами в разработке ПО (например, Jira, Trello)?

23. Что такое системы контроля версий? Приведите примеры (например, Git, SVN).
24. Какие инструменты используются для автоматизации сборки и развертывания ПО (например, Jenkins, Docker)?
25. Что такое интегрированные среды разработки (IDE)? Приведите примеры (например, Visual Studio, IntelliJ IDEA).
26. Что такое тестирование ПО? Какие цели тестирования?
27. Опишите виды тестирования (например, функциональное, нагрузочное, регрессионное).
28. Что такое автоматизированное тестирование? Какие инструменты вы знаете (например, Selenium, JUnit)?
29. Что такое отладка ПО? Какие методы отладки вы знаете?
30. Что такое баг-трекинг? Какие системы используются для отслеживания ошибок (например, Bugzilla, Redmine)?
31. Зачем нужно документирование в разработке ПО? Какие виды документации вы знаете?
32. Что такое техническая документация? Какие разделы она включает?
33. Какие стандарты используются в разработке ПО (например, ISO/IEC 12207, IEEE 830)?
34. Что такое пользовательская документация? Какие требования к ней предъявляются?
35. Что такое качество программного обеспечения? Какие характеристики качества вы знаете?
36. Опишите процесс обеспечения качества ПО (QA).
37. Что такое метрики качества ПО? Приведите примеры (например, количество багов, покрытие тестами).
38. Какие методы используются для улучшения качества ПО?
39. Что такое DevOps? Какие принципы лежат в основе DevOps?
40. Опишите концепцию Continuous Integration (CI) и Continuous Delivery (CD).
41. Что такое микросервисная архитектура? Какие ее преимущества и недостатки?
42. Какие современные технологии и языки программирования используются в разработке ПО?
43. Какие правовые аспекты необходимо учитывать при разработке ПО (например, лицензии, авторское право)?
44. Что такое лицензирование ПО? Какие виды лицензий вы знаете (например, open-source, proprietary)?
45. Какие этические принципы должны соблюдаться при разработке ПО?

4.5.2. Типовые задания для оценки освоения МДК 02.02

Примерный перечень вопросов:

1. Что такое инструментальные средства разработки ПО? Какие категории инструментов вы знаете?
2. Какие задачи решаются с помощью инструментальных средств разработки?
3. Что такое интегрированная среда разработки (IDE)? Приведите примеры популярных IDE.
4. Какие функции обычно предоставляют современные IDE (например, подсветка синтаксиса, автодополнение, отладка)?
5. Что такое система контроля версий (VCS)? Зачем она нужна?

6. Опишите основные принципы работы систем контроля версий.
7. Что такое Git? Какие основные команды Git вы знаете (например, clone, commit, push, pull)?
8. Что такое ветвление (branching) в Git? Как происходит слияние веток (merge)?
9. Какие платформы для хостинга репозиториев вы знаете (например, GitHub, GitLab, Bitbucket)?
10. Что такое конфликты слияния (merge conflicts)? Как их разрешать?
11. Что такое сборка (build) программного обеспечения?
12. Какие инструменты используются для автоматизации сборки (например, Maven, Gradle, Make)?
13. Что такое Continuous Integration (CI)? Какие инструменты CI вы знаете (например, Jenkins, Travis CI)?
14. Опишите процесс настройки автоматической сборки и тестирования в CI/CD.
15. Что такое контейнеризация? Какие инструменты вы знаете (например, Docker, Kubernetes)?
16. Какие инструменты используются для модульного тестирования (например, JUnit, pytest)?
17. Что такое фреймворки для автоматизированного тестирования? Приведите примеры (например, Selenium, Cypress).
18. Какие инструменты используются для нагрузочного тестирования (например, JMeter, LoadRunner)?
19. Что такое отладка? Какие инструменты отладки вы знаете (например, отладчики в IDE, GDB)?
20. Как организовать автоматическое тестирование в проекте?
21. Какие инструменты используются для управления проектами (например, Jira, Trello, Asana)?
22. Что такое баг-трекинг? Какие системы используются для отслеживания ошибок (например, Bugzilla, Redmine)?
23. Как организовать работу команды с использованием инструментов управления задачами?
24. Что такое доски задач (Kanban, Scrum)? Как они используются в разработке ПО?
25. Какие инструменты используются для документирования кода (например, Doxygen, Javadoc)?
26. Что такое генерация кода? Какие инструменты вы знаете (например, Yeoman, JHipster)?
27. Как организовать автоматическое создание документации в проекте?
28. Что такое Continuous Deployment (CD)? Какие инструменты используются для CD (например, Jenkins, GitLab CI)?
29. Как организовать автоматическое развертывание приложения?
30. Какие инструменты используются для мониторинга приложений (например, Prometheus, Grafana)?
31. Что такое логирование (logging)? Какие библиотеки и инструменты вы знаете (например, Log4j, ELK Stack)?
32. Что такое DevOps? Какие инструменты используются в DevOps (например, Ansible, Terraform)?
33. Какие инструменты используются для работы с облачными платформами (например, AWS, Azure, Google Cloud)?

34. Что такое низкокодовые платформы (low-code)? Приведите примеры (например, OutSystems, Mendix).
35. Какие современные инструменты для разработки мобильных приложений вы знаете (например, Flutter, React Native)?
36. Какие инструменты используются для статического анализа кода (например, SonarQube, ESLint)?
37. Что такое анализ уязвимостей в коде? Какие инструменты вы знаете (например, OWASP ZAP, Burp Suite)?
38. Как организовать проверку безопасности в процессе разработки?
39. Как выбрать подходящие инструменты для конкретного проекта?
40. Какие критерии важно учитывать при выборе инструментов разработки?
41. Как организовать обучение команды новым инструментам?
42. Какие проблемы могут возникнуть при использовании инструментов разработки? Как их решать?

4.5.2. Типовые задания для оценки освоения МДК 02.03

Примерный перечень вопросов:

1. Что такое математическое моделирование? Какие цели оно преследует?
2. Опишите основные этапы процесса математического моделирования.
3. Какие типы моделей вы знаете (например, детерминированные, стохастические, дискретные, непрерывные)?
4. Что такое адекватность модели? Как проверить адекватность модели?
5. Какие основные принципы построения математических моделей?
6. Чем отличаются детерминированные и стохастические модели?
7. Что такое дискретные и непрерывные модели? Приведите примеры.
8. Какие модели называются статическими, а какие — динамическими?
9. Что такое линейные и нелинейные модели? В чем их различие?
10. Какие модели относятся к численным, а какие — к аналитическим?
11. Опишите метод наименьших квадратов. Где он применяется?
12. Что такое регрессионный анализ? Какие виды регрессий вы знаете?
13. Как строится модель на основе дифференциальных уравнений? Приведите пример.
14. Что такое метод Монте-Карло? В каких задачах он применяется?
15. Какие методы используются для оптимизации моделей?
16. Приведите пример математической модели в физике.
17. Как используются математические модели в экономике?
18. Опишите пример математической модели в биологии (например, модель популяции).
19. Какие математические модели применяются в технических системах?
20. Как математическое моделирование используется в экологии?
21. Что такое численные методы? Какие задачи они решают?
22. Опишите метод Эйлера для решения дифференциальных уравнений.
23. Что такое метод конечных разностей? Где он применяется?
24. Какие методы используются для решения систем линейных уравнений (например, метод Гаусса, метод Якоби)?
25. Что такое интерполяция и аппроксимация? Какие методы вы знаете?
26. Что такое оптимизация? Какие задачи оптимизации вы знаете?
27. Опишите метод градиентного спуска. Где он применяется?

28. Что такое линейное программирование? Приведите пример задачи.
29. Какие методы используются для анализа устойчивости моделей?
30. Что такое чувствительность модели? Как ее оценить?
31. Какие программные средства используются для математического моделирования (например, MATLAB, MathCAD, Python)?
32. Как использовать Python для построения и анализа математических моделей?
33. Что такое библиотеки для научных вычислений (например, NumPy, SciPy)?
34. Какие инструменты используются для визуализации моделей (например, Matplotlib, Plotly)?
35. Как организовать численные расчеты в Excel?
36. Приведите пример задачи оптимизации в логистике.
37. Как математическое моделирование используется в прогнозировании?
38. Опишите пример моделирования в медицине (например, распространение заболеваний).
39. Какие модели используются в финансовой аналитике?
40. Как математическое моделирование применяется в робототехнике?
41. Какие ошибки могут возникать при построении математических моделей?
42. Что такое погрешность модели? Как ее минимизировать?
43. Какие ограничения имеют математические модели?
44. Как учитывать неопределенность в математических моделях?
45. Что такое переобучение модели? Как его избежать?
46. Что такое машинное обучение? Как оно связано с математическим моделированием?
47. Какие модели используются в искусственном интеллекте?
48. Что такое нейронные сети? Как они применяются в моделировании?
49. Как математическое моделирование используется в Big Data?
50. Какие современные технологии и методы расширяют возможности математического моделирования?

4.6. Оценка по учебной и (или) производственной (по профилю специальности) практике

4.6.1 Виды работ учебной практики и проверяемые результаты обучения по профессиональному модулю:

Таблица 6

Виды работ	Проверяемые результаты (ПК, ОК, профессиональный опыт, умения)
Ознакомление студентов с программным обеспечением учебного заведения: внедрением автоматизированных систем управления, развитием и распространением автоматизированных систем управления, перспективами развития автоматизации в области электронных вычислительных машин. Изучение программных продуктов учебного заведения: «первичные»	ОК 01- ОК09, ПК 2.1- ПК 2.5

программы для обработки информации в организации, внедрение новых программных продуктов специализирующихся на конкретное учебное заведение, способы работы с конкретным программным обеспечением, имеющимся в организации. Разработка программного обеспечения, которое будет направлено на автоматизацию специализированных информационных потоков в организации Тестирование разработанного программного продукта Оформление технической и программной документации	

4.6.2. Виды работ производственной (по профилю специальности) практики и проверяемые результаты обучения по профессиональному модулю:

Виды работ	Проверяемые результаты (ПК, ОК, профессиональный опыт, умения)
Ознакомление студентов с программным обеспечением информационного центра предприятия: внедрением автоматизированных систем управления, развитием и распространением автоматизированных систем управления, перспективами развития автоматизации в области электронных вычислительных машин. Изучение программных продуктов предприятия: «первичные» программы для обработки информации на предприятии, внедрение новых программных продуктов специализирующихся на конкретное производство, способы работы с конкретным программным обеспечением, имеющимся на предприятии. Разработать программное обеспечение, которое будет направлено на автоматизацию специализированных информационных потоков на предприятии Провести тестирование разработанного программного продукта Для разработанного программного обеспечения оформить техническую и программную документацию	ОК 01- ОК09, ПК 2.1- ПК 2.5

5.Фонд оценочных средств для аттестации по модулю[3] ***(экзамен квалификационный)***

ПАСПОРТ

Назначение:
ФОС предназначен для контроля и оценки результатов освоения профессионального модуля

ПАКЕТ ЭКЗАМЕНАТОРА

УСЛОВИЯ

Количество вариантов каждого задания / пакетов заданий для экзаменующегося: …

Время выполнения каждого задания: …

Оборудование: Оборудование учебного кабинета и рабочих мест кабинета: рабочие места (25-30) по количеству обучающихся; рабочее место преподавателя.

Технические средства обучения: компьютер; аудиовизуальные средства обучения; программное обеспечение общего и профессионального обучения.

КРИТЕРИИ ОЦЕНКИ

1) Ход выполнения задания

Таблица 6

Коды проверяемых компетенций	Показатели оценки результата	Оценка (да / нет)

6. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

6.1. Для реализации программы профессионального модуля должны быть предусмотрены следующие специальные помещения:
Лаборатории Программного обеспечения и сопровождения компьютерных систем, оснащенные:

Задания к аттестации по модулю формируются 3 способами:

1. Задания, ориентированные на проверку освоения вида деятельности (всего модуля) в целом.

2. Задания, проверяющие освоение группы компетенций, соответствующих определенному разделу модуля.

3. Задания, проверяющие освоение отдельной компетенции внутри профессионального модуля.

- Автоматизированные рабочие места на 12-15 обучающихся (процессор не ниже Core i3, оперативная память объемом не менее 4 Гб;) или аналоги;
- Автоматизированное рабочее место преподавателя (процессор не ниже Core i3, оперативная память объемом не менее 4 Гб;) или аналоги;
- Проектор и экран;
- Маркерная доска;
- Программное обеспечение общего и профессионального назначения

Учебная практика проводится на базе учебного заведения в рамках работы полигона разработки бизнес-приложений и проектирования информационных систем и (или) профильных предприятий различных форм собственности по договорам. Завершается учебная практика дифференцированным зачетом.

Производственная практика (по профилю специальности) проводится на базе учебного заведения в рамках работы полигона разработки бизнес-приложений и проектирования информационных систем и (или) профильных предприятий различных форм собственности по договорам.

7. Информационное обеспечение реализации программы профессионального модуля

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, рекомендованные ФУМО, для использования в образовательном процессе. При формировании библиотечного фонда образовательной организации выбирается не менее одного издания из перечисленных ниже печатных изданий и (или) электронных изданий в качестве основного, при этом список, может быть дополнен новыми изданиями.

Основная литература

1. Макушкин, В. А. Интеграция приложений на основе WebSphere MQ : курс лекций / В. А. Макушкин, Д. С. Володичев. - Москва : ИНТУИТ, 2016. - 240 с. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2139090
2. Алексеев, Ю. Е. Разработка функций и модулей пользователя на языке С в среде VS C++ : учебное пособие / Ю. Е. Алексеев, А. В. Куров. - Москва : МГТУ им. Баумана, 2020. - 114 с. - ISBN 978-5-7038-5340-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/2013681
3. Ленд, М. Python: Непрерывная интеграция и доставка : практическое руководство / М. Ленд ; пер. с англ. А. Е. Мамонова, Д. А. Беликова. - Москва : ДМК Пресс, 2020. - 168 с. - ISBN 978-5-97060-797-8. - Текст : электронный. - URL: https://znanium.com/catalog/product/1210637
4. Бабушкин, В. М. Разработка защищенных программных средств информатизации производственных процессов предприятия : учебное пособие / В. М. Бабушкин, М. В. Тумбинская. - Москва ; Вологда : Инфра-Инженерия, 2024. - 260 с. - ISBN 978-5-9729-1618-4. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2169718