МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕУЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

Рабочая программа профессионального модуля ПМ.05 «ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ»

для специальности

09.02.07 «Информационные системы и программирование»

Екатеринбург

2025

Чистяков Валерий Николаевич
я подтверждаю этот документ своей удостоверяющей подписью
место подписания
2026.04.08 17:04:54+05'00'

Рабочая программа ПМ.05 **«Проектирование и разработка информационных систем»** разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности **09.02.07 «Информационные системы и программирование».**

Организация-разработчик: ГАПОУ СО «Екатеринбургский монтажный колледж»

Разработчик Грибова А.В. - преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», ВКК;, Раковская Ю.В. преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», 1КК; Софьина Н.А. - преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», ВКК;,

Рассмотрена методическим объединением Информационных технологий
Руководитель МО _____Софьина Н.А.
07.05.25

УТВЕРЖДАЮ
Заместитель директора по учебной работе
_____ Л.С.Хоринова
«13» мая 2025

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ ПМ05. «ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ»

1.1. ОБЛАСТЬ ПРИМЕНЕНИЯ РАБОЧЕЙ ПРОГРАММЫ

Рабочая программа профессионального модуля является частью основной профессиональной образовательной программы среднего профессионального образования подготовки специалистов среднего звена по специальности 09.02.07 «Информационные системы и программирование».

1.2. ЦЕЛЬ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

В результате изучения профессионального модуля обучающиеся должны освоить основной вид деятельности «Проектирование и разработка информационных систем» и соответствующие ему профессиональные и общие компетенции.

Выпускник, освоивший образовательную программу, должен обладать следующими компетенциями:

1.2.1. Перечень общих компетенций

Код	**Наименование общих компетенций**
ОК 1.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 2.	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
ОК 3.	Планировать и реализовывать собственное профессиональное и личностное развитие
ОК 4.	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
ОК 5.	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
ОК 9.	Пользоваться профессиональной документацией на государственном и иностранном языках. (в ред. Приказа Минпросвещения России от 17.12.2020 N 747)

1.2.2. Перечень профессиональных компетенций

Код	**Наименование видов деятельности и профессиональных компетенций**
ВД 5.	Проектирование и разработка информационных систем
ПК 5.1.	Собирать исходные данные для разработки проектной документации на информационную систему.
ПК 5.2.	Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика

ПК 5.3.	Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием.
ПК 5.4.	Производить разработку модулей информационной системы в соответствии с техническим заданием
ПК 5.5.	Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы

ПК 5.1. Собирать исходные данные для разработки проектной документации

Требования по практическому опыту	Умения	Знания
- Иметь практический опыт анализа предметной области. - Использовать инструментальные средства обработки информации. - Обеспечивать сбор данных для анализа использования и функционирования информационной системы. - Определять состав оборудования и программных средств разработки ИС. - Выполнять работы предпроектной стадии. - Собирать документацию заказчика в соответствии с трудовым заданием. - Проводить анкетирование и интервьюирование представителей заказчика по заданию. - Документировать собранные данные по регламентам организации.	- Осуществлять постановку задачи по обработке информации. - Выполнять анализ предметной области. - Использовать алгоритмы обработки информации для разных приложений. - Работать с инструментальными средствами обработки информации. - Выбирать модель построения информационной системы. - Анализировать системы с помощью методик моделирования: — прецедентов использования (диаграммы, описания актёров и пр.), — структурного моделирования (диаграммы классов и объектов), — динамического моделирования (схемы последовательности, состояний, деятельности), — инструментов и методов моделирования (диаграммы сущностей и связей, нормализация, словарь данных). - Проводить анкетирование, интервьюирование, собирать исходную документацию. - Проектировать системы на основе диаграмм классов, последовательности, состояний, деятельности, описаний объектов, схем баз данных, интерфейсов человек–машина. - Синтезировать сложную или неоднородную информацию. - Определять функциональные и нефункциональные требования спецификации.	- Основные виды и процедуры обработки информации, модели и методы решения задач обработки информации. - Основные платформы для создания, исполнения и управления информационной системой. - Основные модели построения ИС, их структура, особенности, области применения. - Основные процессы управления проектом разработки. - Методы и средства проектирования, разработки и тестирования ИС. - Возможности типовой информационной системы. - Предметная область автоматизации. - Инструменты и методы выявления требований.

ПК 5.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика

Требования по практическому опыту	Умения	Знания
- Иметь практический опыт разработки проектной документации на ИС.	- Осуществлять математическую и информационную постановку задач по обработке информации. - Использовать алгоритмы обработки информации для различных приложений.	- Основные платформы для создания, исполнения и управления информационной системой. - Национальную и международную систему стандартизации и сертификации, систему обеспечения качества продукции, методы контроля качества. - Сервисно-ориентированные архитектуры. - Важность рассмотрения всех вариантов и поиска наилучшего решения на основе анализа интересов клиента. - Методы и средства проектирования информационных систем. - Основные понятия системного анализа. - Важность тщательного документирования разработанных решений.

ПК 5.3. Разрабатывать подсистемы безопасности информационной системы

Требования по практическому опыту	Умения	Знания
- Управлять процессом разработки приложений с использованием инструментальных средств. - Модифицировать отдельные модули информационной системы. - Программировать согласно требованиям технического задания.	- Создавать и управлять проектом разработки приложения, формулировать его задачи. - Использовать языки структурного, объектно-ориентированного программирования и сценариев для создания независимых программ. - Разрабатывать графический интерфейс приложения. - Кодировать на языках программирования.	- Национальные и международные стандарты и системы сертификации, системы обеспечения качества. - Методы контроля качества объектно-ориентированного программирования. - Объектно-ориентированное программирование. - Спецификации языков программирования, принципы создания GUI, файлового ввода-вывода, сетевых серверов и клиентов. - Использование системного анализа и методологий проектирования (например, UML, MVC, фреймворки, шаблоны проектирования). - Оптимизация архитектуры системы с учетом модульности и повторного использования. - Важность применения объектно-ориентированных технологий и методологий разработки. - Обработка всех сценариев и исключений. - Соблюдение стандартов кодирования, оформление документации, дизайн интерфейсов, управление каталогами и файлами. - Точное и постоянное управление версиями. - Использование существующего кода для анализа и модификации. - Варианты выбора средств разработки.

ПК 5.4. Производить разработку модулей информационной системы в соответствии с техническим заданием

Требования по практическому опыту	Умения	Знания

- Разрабатывать документацию по эксплуатации информационной системы. - Проводить оценку качества и экономической эффективности информационной системы в рамках своей компетенции. - Модифицировать отдельные модули информационной системы. - Разрабатывать код прототипа информационной системы и баз данных прототипа согласно трудовому заданию.	- Использовать языки структурного, объектно-ориентированного программирования и языка сценариев для создания независимых программ. - Решать прикладные вопросы программирования и языка сценариев. - Проектировать и разрабатывать систему по заданным требованиям и спецификациям. - Разрабатывать графический интерфейс приложения. - Создавать проект разработки приложения и формулировать его задачи. - Кодировать на языках программирования. - Использовать последнюю версию ПО среды разработки и инструменты для изменения существующих кодов и написания нового кода клиент-серверного приложения - Определять и интегрировать соответствующие библиотеки и фреймворки в программное решение.	- Национальную и международную системы стандартизации и сертификации, систему обеспечения качества продукции, методы контроля качества. - Основы объектно-ориентированного программирования. - Спецификации языков программирования, принципы создания графического пользовательского интерфейса (GUI). - Важность рассмотрения различных вариантов и выбора наилучшего решения на основе анализа и интересов клиента. - Принципы файлового ввода-вывода, создания сетевых серверов и клиентов. - Платформы для создания, исполнения и управления информационной системой. - Современные объектно-ориентированные языки программирования. - Программные средства и платформы инфраструктуры ИТ-организаций. - Важность системного анализа и методологий проектирования (Unified Modeling Language, MVC, фреймворки, шаблоны проектирования). - Оценка и принятие решений о целесообразности применения новых технологий. - Оптимизация архитектуры системы с учетом модульности и повторного использования. - Использование методологий разработки системы (например, объектно-ориентированные технологии). - Обработка нормальных и ненормальных сценариев, исключений. - Соблюдение стандартов: правила кодирования, руководство по стилю, оформление документации, дизайн интерфейса, управление каталогами и файлами. - Точный и непрерывный контроль версий. - Использование существующего кода для анализа и модификации. - Важность выбора наиболее подходящих средств разработки из предложенных вариантов.

ПК 5.5. Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы

Требования по практическому опыту	Умения	Знания
- Применять методики тестирования	- Использовать методы тестирования по техническому	- Языки, утилиты, среды программирования и средства пакетного выполнения тестовых процедур.

разрабатываемых приложений. - Проводить тестирование в соответствии с трудовым заданием. - Выполнять интеграционное тестирование на основе тест-планов. - Документировать результаты тестов. - Устранять обнаруженные несоответствия. - Фиксировать результаты тестирования в системе учёта.	заданию. - Тестировать результаты собственной работы. - Тестировать информационную систему с использованием тест-планов. - Составлять планы тестирования (модульное, объемное, интеграционное, приемочное и др.). - Разрабатывать тест-кейсы и проверять результаты. - Исправлять ошибки. - Составлять отчёты о тестировании.	- Методы контроля качества ПО согласно стандартам. - Основы верификации и аттестации программного обеспечения. - Особенности программных средств разработки и тестирования ИС. - Устройство и функционирование современных информационных систем. - Стандарты качества ПО и процессов обеспечения качества. - Роль тестирования в жизненном цикле ПО. - Основные понятия и принципы тестирования. - Основы автоматизации тестирования. - Техники тест-дизайна. - Базовые принципы взаимодействия с REST API и тестирования API. - Инструменты и методы модульного тестирования. - Современные методики тестирования ИС. - Методы и правила создания и документирования тестовых наборов данных. - Требования к структуре и форматам хранения тестовых данных. - Средства проверки работоспособности и отладки ПО. - Методы автоматической проверки ПО. - Метрики ПО и методы оценки его характеристик.

ПК 5.6. Разрабатывать техническую документацию на эксплуатацию информационной системы

Требования по практическому опыту	Умения	Знания
- Разрабатывать проектную документацию на информационную систему. - Формировать отчетную документацию по итогам работ. - Использовать стандарты оформления программной документации.	- Разрабатывать проектную документацию на эксплуатацию информационной системы. - Использовать стандарты при оформлении документации. - Проявлять профессионализм в подготовке документации. - Разрабатывать документацию пользователей. - Работать с технической документацией.	- Основные модели построения информационных систем и их структура. - Критерии оценки качества и надежности функционирования информационной системы. - Реинжиниринг бизнес-процессов. - Важность тщательного документирования разработанных решений.

ПК 5.7. Производить оценку информационной системы для выявления возможности ее модернизации

Требования по практическому опыту	Умения	Знания
- Проводить оценку качества и экономической эффективности	- Использовать методы и критерии оценки предметной области.	- Системы обеспечения

информационной системы в своей компетенции. - Использовать критерии оценки качества и надежности функционирования информационной системы.	- Определять стратегию развития бизнес-процессов организации. - Решать прикладные задачи интеллектуальных систем с использованием статических и реального времени экспертных систем.	качества продукции. - Методы контроля качества согласно стандартам.

1.3. Количество часов, отводимое на освоение профессионального модуля
Всего часов **648 часа**
Из них на освоение МДК: **378 часов**, на практики, в том числе учебную **144 часа**

и производственную: **180 часов**
самостоятельная работа **90 часов**

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2.1. Структура профессионального модуля

			Занятия во взаимодействии с преподавателем, час					
Коды профессиональных общих компетенций	Наименования разделов профессионального модуля**	Суммарный объем нагрузки, час.	Обучение по МДК			Практики		Самостоятельная работа
			Всего	Лабораторных и практических занятий	Курсовых работ (проектов) *	Учебная	Производственная (если предусмотрена рассредоточенная практика)	
1	2	3	4	5	6	7	8	9
ПК 5.1, ПК 5.2, ПК 5.6, ПК 5.7	МДК. 05.01 Проектирование и дизайн информационных систем	**150**	**110**	58	-	**72**	-	**40**
ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4	МДК. 05.02 Разработка кода информационных систем.	**166**	**140**	82	-	**72**	-	**26**
ПК 5.2, ПК 5.5, ПК 5.6,	МДК. 05.03 Тестирование информационных систем	**152**	**128**	46	30	-	-	**24**
ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4	Производственная практика (по профилю специальности)	**180**					**180**	-
	Всего:	**648**	**378**	**186**	**30**	**144**	**180**	**90**

2.2. Тематический план и содержание профессионального модуля (ПМ)

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовая работа (проект) *(если предусмотрены)*	Объем часов	Коды компетенций, формированию которых способствует элемент программы
1	**2**	**3**	**4**
МДК. 05.01 Проектирование и дизайн информационных систем		**110**	
Наименование разделов и тем	**Содержание**	**46**	
Тема 5.1.1. Введение в проектирование ИС	Информационная система (ИС). Компоненты информационной системы.	**2**	ОК 01, ОК 02 ОК 03, ПК 5.1, ПК 5.2, ПК 5.6, ПК 5.7
	Технологии проектирования ИС. Введение в проектирование цифрового продукта. Принципы и методы проектирования ИС, этапы проектирования ИС. Особенности проектирования цифрового продукта.	**2**	
	Жизненный цикл программного обеспечения ИС. Каскадная модель проектирования ИС. Спиральная модель проектирования ИС. Итерационная модель проектирования ИС.	**2**	
	Свойства информационной системы. Основы создания и принципы функционирования информационной системы. Основные модели построения информационных систем, их структура, особенности и области применения. Основные направления и тенденции развития конструирования программного обеспечения. Понятие и обобщённая схема конструирования программного обеспечения, его место в жизненном цикле программного обеспечения(ПО).	**2**	
Тема 5.1.2. Основы системного анализа	Исходные данные для проектирования. Предметная область автоматизации. Проведение предпроектного обследования предприятия. Анализ интересов клиента. Анализ предметной области. Выбор вариантов решений. Построение организационно-функциональной структуры компании для которой разрабатывается цифровой продукт. Основные понятия организационного бизнес-моделирования. Миссия компании, дерево целей и стратегии их достижения. Полная бизнес-модель компании	**2**	

	Методы и средства проектирования информационных систем. Поиск идеи продукта и его места на рынке. Исследование идеи. Создание концепции программного продукта. Презентация идей. Моделирование функциональных требований в виде пользовательских историй и пользовательских сценариев. Пользовательские сценарии: бизнес-сценарий использования и системный Сценарий использования.	**2**	
	Сбор данных для анализа использования и функционирования информационной системы. Организация и методы сбора информации. Основные понятия системного и структурного анализа. Методы обследования управленческих процедур. Инструменты и методы выявления требований к информационной системе.	**2**	
Тема 5.1.3. CASE-средства проектирования	Case-средства для моделирования функциональных требований к информационной системе.	**2**	
	Принципы построения модели IDEF0: контекстная диаграмма, субъект моделирования, цель и точка зрения. Диаграммы IDEF0: диаграммы декомпозиции, диаграммы дерева узлов.	**2**	
	Моделирование предметной области. Использование унифицированного языка моделирования (UnifiedModellingLanguage).	**2**	
	Язык UML. Диаграммы последовательности. Диаграммы классов. Диаграммы объектов.	**2**	
	Язык UML. Диаграммы прецедентов. Диаграммы взаимодействия. Диаграммы кооперации.	**2**	
Тема 5.1.4. Проектирование интерфейсов	Основные принципы построения интерфейса информационной системы. Исследование интерфейсов. Тенденции и тренды в дизайне. Концепты в дизайне продукта. Визуальный язык. Применение основных принципов дизайна при проектировании интерфейса информационной системы.	**2**	
	Методы проектирования пользовательского интерфейса. Понятия эскиза и дизайн-макета информационной системы. Прототипирование - создание прототипа пользовательского интерфейса. Жизненный цикл прототипа.	**2**	
	Инструменты макетирования и прототипирования пользовательских интерфейсов. Способы проверки решения. Тестирование прототипов. Развитие продукта. Инструменты макетирования и прототипирования программных продуктов в зависимости от поставленной задачи.	**2**	

Тема 5.1.5. Система обеспечения качества информационных систем	Основные понятия качества информационной системы. Национальный стандарт обеспечения качества автоматизированных информационных систем.	**2**	
	Международная система стандартизации и сертификации качества продукции. Стандарты группы ISO.	**2**	
	Методы контроля качества в информационных системах. Особенности контроля в различных видах систем	**2**	
Тема 5.1.6 Разработка документации информационных систем	Перечень и комплектность документов на информационные системы согласно ЕСПД и ЕСКД. Задачи документирования. Стандарты при оформлении программной документации. Предпроектная стадия разработки. Документирование процесса проектирования ИС.	**2**	
	Разработка технического задания (далее — ТЗ) на проектирование, технических требований к проектируемой информационной системе (далее — ТТ). Техническое задание на разработку: основные разделы. Ведения переговоров с клиентом, сбор и подтверждение требований клиента.	**2**	
	Проектная документация. Документирование программной системы. Техническая документация. Отчетная документация. Документирования разработанных решений.	**2**	
	Пользовательская документация: руководство пользователя. Проектная документация на эксплуатацию информационной системы. Разработка документации пользователей; Работа с технической документацией.	**2**	
	Самодокументирующиеся программы. Назначение, виды и оформление сертификатов.	**2**	
	Тематика практических занятий и лабораторных работ	**58**	
	Анализ компонентов информационной системы. Исследование структуры и взаимодействия компонентов ИС на примере реальных кейсов.	**2**	ОК 01, ОК 02 ОК 03, ПК 5.1, ПК 5.2, ПК 5.6, ПК 5.7
	Модели жизненного цикла ИС. Сравнительный анализ каскадной, спиральной и итерационной моделей проектирования.	**2**	
	Предпроектное обследование предприятия. Проведение анализа предметной области и составление отчета.	**2**	
	Построение организационно-функциональной структуры. Разработка структуры компании для цифровой трансформации.	**2**	
	Сбор и анализ требований к ИС. Методы выявления и документирования требований заказчика.	**2**	
	Работа с IDEF0-диаграммами. Построение контекстной диаграммы и диаграмм декомпозиции.	**2**	
	Работа с IDEF0-диаграммами. Построение диаграмм декомпозиции.	**2**	

	Язык UML. Построение диаграммы последовательности.	**2**	
	Язык UML. Построение диаграммы объектов.	**2**	
	Язык UM. Построение диаграммы классов.	**2**	
	Язык UML. Построение диаграмм вариантов использования.	**2**	
	Язык UML. Построение диаграммы деятельности.	**2**	
	Язык UML. Построение диаграммы кооперации**.**	**2**	
	Проектирование пользовательских интерфейсов. Работа с современными инструментами создания макетов интерфейсов.	**2**	
	Разработка прототипов интерфейсов с учетом принципов юзабилити/	**2**	
	Стандарты качества ИС. Изучение требований национальных и международных стандартов.	**2**	
	Метрики качества. Оценка показателей эффективности ИС.	**2**	
	Анализ рисков и характеристик качества программного обеспечения	**2**	
	Разработка требований безопасности информационной системы.	**2**	
	Разработка технического задания на проектирование.	**2**	
	Разработка технических условий эксплуатации ИС. Определения состава оборудования и программных средств разработки информационной системы	**2**	
	Разработка проектной документации на информационную систему	**2**	
	Разработка пользовательской документации на информационную систему	**2**	
	Проектирование спецификации информационной системы по индивидуальному заданию	**2**	
	Разработка документации по безопасности информационной системы	**2**	
	Разработка документации по управлению качеством информационной системы	**2**	
	Разработка руководства по инсталляции программного средства индивидуальному заданию	**2**	
	Разработка руководства пользователя программного средства по индивидуальному заданию	**2**	
	Оформление сертификата на информационную систему.	**2**	
Экзамен		**6**	

Внеаудиторная (самостоятельная) учебная работа при изучении МДК. 05.01 1.Анализ предметной области 2.Применение метода персонажей 3.Проектирование экранных форм приложения (создание прототипа приложения) 4.Создание технического задания на разработку информационной системы	**40**	ОК 01, ОК 02 ОК 03, ПК 5.1, ПК 5.2, ПК 5.6, ПК 5.7

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовая работа (проект)	Объем часов	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
МДК. 05.02 Разработка кода информационных систем.		**134**	
Тема 5.2.1. Архитектура и проектирование серверных приложений	**Содержание**	**54**	
	Среды и платформы разработки, стандарты кодирования. Среда разработки как комплекс инструментов: редактор кода, сборка, отладка, деплой. Кроссплатформенность и поддержка технологий. Среды разработки и платформы для серверной части: назначение и возможности. **С**тандарты наименование переменных и функций,	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Клиент-серверная архитектура. Описание моделей распределения вычислений и данных между клиентом, сервером и базой данных. Одноуровневая: всё на одном устройстве. Двухуровневая: клиент и сервер с базой данных. Трёхуровневая: клиент, бизнес-логика, база данных. Многоуровневая: несколько логических и физических уровней для масштабируемости и гибкости.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Роли компонентов приложения. Распределение ответственности: интерфейс, бизнес-логика, данные; коммуникация между компонентами, обеспечение надежности и производительности.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Сервисно-ориентированные архитектуры. Организация приложения как набора независимых сервисов, взаимодействующих через четко определённые интерфейсы; особенности масштабируемости, обновляемости и безопасности.	**2**	ОК 01, ОК 02 ОК 03, ПК 5.1. ПК 5.2., ПК 5.6
	Объектно-ориентированные и структурные подходы. Принципы проектирования на уровне абстракций, инкапсуляция, наследование, полиморфизм; разница подходов и их влияние на архитектуру. Объектно-ориентированные и структурные подходы в программировании серверной части.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Паттерны проектирования. Типовые решения для архитектурных и программных задач: Singleton, Factory, Observer, MVC и др.	**2**	ПК5.1, ПК 5.2,
	Методологии и структура проектов. Подходы к управлению проектом и кодом: Waterfall, Agile, Continuous Integration; построение дерева исходников и модулей.	**2**	ПК 5.3, ПК 5.4
	Стандарты оформления документации. форматирование документации, управление артефактами проектирования	**2**	ПК5.1, ПК 5.2,

Тема 5.2.2. Разработка и модификация информационных систем	**Системы контроля версий** .Системы контроля версий: принципы, организация репозиториев и управление доступом. Ведение истории изменений кода, ветвление, слияния, разграничение прав на репозиторий, работа с конфликтами.	**2**	ОК 01, ОК 02 ОК 03, ПК 5.1. ПК 5.2., ПК 5.6
	Управление конфигурациями. Управление конфигурациями и настройка окружения серверного приложения. Изоляция параметров от кода, настройка через переменные среды, конфигурационные файлы, обеспечение повторяемости развёртывания.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Взаимодействие с базой данных . Взаимодействие серверного приложения с базой данных: типы, модели, основные операции. Принципы хранения и извлечения данных, основные модели (реляционная, NoSQL), операции CRUD.	**2**	ОК 01, ОК 02 ОК 03, ПК 5.1. ПК 5.2., ПК 5.6
	ORM: концепции и применение. ORM: концепции, преимущества и применение. Автоматизация маппинга объектов в базе данных, снижение дублирования кода, упрощение поддержки.	**6**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Работа с REST API. Работа с REST API: проектирование, реализация и взаимодействие. Стандарты построения API, методы HTTP, форматы передачи данных, построение гибких и расширяемых интерфейсов. Использование средств для тестирования API – запросов .	**4**	ОК 01, ОК 02 ОК 03, ПК 5.1. ПК 5.2., ПК 5.6
Тема 5.2.3. Безопасность, развёртывание и оптимизация	**Аутентификация.** Механизмы подтверждения личности пользователя: Basic, OAuth, JWT, многофакторная аутентификация. Протоколы и практические методы реализации аутентификации	**4**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Авторизация . Авторизация: модели ролей и разграничение доступа. Основы безопасности серверной части: защита от уязвимостей и типичных атак. Определение прав и ограничений на уровне ролей и ресурсов, контроль доступа.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Безопасность в микросервисах. Безопасность и шифрование в микросервисных системах. Защита от SQL Injection, XSS, CSRF, обеспечение целостности и конфиденциальности.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Асинхронное программирование. Асинхронное программирование и параллелизм в серверных приложениях. Механизмы неблокирующей обработки запросов, управление потоками и задачами для высокой производительности.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Контейнеризация приложений. Контейнеризация приложений (Docker): концепции и основные принципы. Изоляция приложений, переносимость, создание образов, запуск контейнеров.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Оркестрация контейнеров. Управление кластером, масштабирование, балансировка нагрузки, обновления без простоев. Оркестрация контейнеров (Kubernetes), развертывание серверных приложений и автоматизация развертывания.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Сессии и состояние пользователей. Сессии и управление состоянием пользователей на сервере. Хранение данных о пользователях между запросами, куки, сессии, устойчивость к сбоям.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4

	Логирование и мониторинг. Логирование и мониторинг серверных приложений. Сбор и анализ логов, настройка оповещений, мониторинг показателей здоровья системы.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Балансировка нагрузки и масштабируемость. Балансировка нагрузки и масштабируемость. Распределение запросов по ресурсам, горизонтальное/вертикальное масштабирование, отказоустойчивость. Интеграция сервисов: транспортные протоколы и форматы сообщений.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Работа с веб-сокетами . Работа с веб-сокетами и реализация real-time взаимодействия.	**2**	ПК5.1, ПК 5.2, ПК 5.3, ПК 5.4
	Тематика практических занятий и лабораторных работ	**82**	
	Лабораторная работа «**Настройка среды разработки и платформы».** Настройка IDE, конфигурацию среды, запускать минимальные серверные приложения.	**2**	ОК 01, ОК 02 ОК 03, ПК 5.1. ПК 5.2., ПК 5.6
	Лабораторная работа «**Системы контроля версий» .** Работа с репозиториями. Моделирование взаимодействия компонентов в приложении с клиент-серверной архитектурой, отправка результатов работы в репозиторий.	**2**	
	Лабораторная работа «**Объектно-ориентированные и структурные подходы».** Реализация основных классов и модулей с четкой структурой для серверных функций.	**2**	
	Лабораторная работа «**Паттерны проектирования».** Использование паттернов (например, MVC, Repository) при создании приложений.	**2**	
	Лабораторная работа «**Взаимодействие с базой данных» .** Проектирование бизнес-логики приложения. Создание CRUD -запросов: вывод данных. Получение данных из базы данных. Реализация запросов к БД	**2**	
	Лабораторная работа «**Взаимодействие с базой данных» .** Проектирование бизнес-логики приложения. Создание CRUD -запросов: запросы на добавление в базу данных. Создание интерфейсов для реализации взаимодействия с базой данных.	**2**	
	Лабораторная работа «**Взаимодействие с базой данных» .** Проектирование бизнес-логики приложения. Создание CRUD -запросов: запросы на изменение данных в базе данных, удаление информации. Создание интерфейсов для реализации взаимодействия с базой данных.	**2**	
	Лабораторная работа «**Взаимодействие с базой данных» .** Проектирование бизнес-логики приложения. Создание CRUD -запросов: запросы на вывод данных из нескольких отношений. Создание интерфейсов для реализации взаимодействия с базой данных.	**2**	
	Лабораторная работа «**Взаимодействие с базой данных» .** Проектирование бизнес-логики приложения, создание запросов. Создание интерфейсов для реализации взаимодействия с базой данных.	**2**	
	Лабораторная работа «**ORM: концепции и применение»** . Моделирование сущностей и выполнение операций на объектном уровне.	**2**	
	Лабораторная работа «**Сервисно-ориентированные архитектуры (SOA) и микросервисы».** Разработка упрощенного сервисного API с отдельными компонентами.	**2**	

	Лабораторная работа «**Взаимодействие между сервисами».** Разработка API. Взаимодействия между сервисами, учитывая независимость и масштабируемость.	**2**	
	Лабораторная работа «**Работа с REST API». П**роектирование URL, методы, протоколы, создание и тестирование API конечные точки, обработка HTTP-запросов и ответов	**2**	
	Лабораторная работа «**Валидация данных в REST API».** Реализация механизмов проверки корректности входящих данных на уровне моделей или DTO с использованием аннотаций или схем валидации. Создание тестовых сценариев для проверки обязательных полей, форматов, диапазонов значений. Обработка ошибок валидации с возвратом информативных сообщений клиенту и корректным HTTP-статусом. Внедрение валидации на уровне контроллеров, сервисов или промежуточных слоев для обеспечения безопасности и качества данных API. Использование популярных библиотек и инструментов для валидации в выбранном фреймворке.	**2**	
	Лабораторная работа «**Соблюдение стандартов кодирования и документации» .** Оформление API в OpenAPI/Swagger.	**2**	
	Лабораторная работа **«Безопасность и управление доступом в** API**: аутентификация».** Реализация аутентификации с помощью JWT, OAuth.	**2**	
	Лабораторная работа **«Безопасность и управление доступом в API: авторизация».** Создание ролей и контроль доступа к методам API. Настройка системы доступа к функциям приложения с разграничением ролей	**2**	
	Лабораторная работа **«Безопасность микросервисов API» .** Шифрование данных, безопасный обмен между сервисами.	**2**	
	Лабораторная работа **«Асинхронное программирование».** Реализация асинхронных вызовов, обработка нескольких запросов параллельно	**2**	
	Лабораторная работа **«Развёртывание и эксплуатация: контейнеризация».** Dockerfile, Сборка и запуск контейнеров.	**2**	
	Лабораторная работа «**Создание базового веб-приложения с фреймворком**». Создание и запуск минимального веб-приложения с одной страницей.	**2**	
	Лабораторная работа «**Добавление динамических представлений (views)**». Разработка нескольких страниц с динамическим содержимым и обработкой пользовательских запросов.	**2**	
	Лабораторная работа «**Настройка URL-маршрутизации веб-приложения**». Конфигурирование маршрутов для многостраничного приложения с пере-дачей параметров.	**4**	
	Лабораторная работа «**Использование шаблонов для UI**». Использование шаблонов для построения интерфейса веб-приложения. Создание пользовательского интерфейса через систему шаблонов фрейм-ворка.	**2**	
	Лабораторная работа «**Определение и работа с моделями данных в веб-приложении**». Создание моделей, заполнение и вывод данных через административный интерфейс и пользовательский.	**2**	

	Лабораторная работа «**Реализация форм регистрации и авторизации пользователей в веб-приложении**». Настройка системы регистрации и авторизации пользователей и безопасного входа в систему.	**2**	
	Лабораторная работа «**Управление сессиями и состоянием».** Создание и поддержание пользовательских сессий через куки и сессионные данные.	**2**	
	Лабораторная работа «**Работа с формами и валидация данных**» Разработка форм для ввода данных, проверка корректности и обработка ошибок.	**2**	
	Лабораторная работа «**Обработка загрузки файлов в приложении**» Приём, сохранение и обработка загруженных пользователем файлов на сервер.	**2**	
	Лабораторная работа «**Реализация функционала REST API внутри веб-приложения**». Создание и тестирование RESTful точек доступа с CRUD операциями.	**2**	
	Лабораторная работа «**Интеграция внешних API в веб-приложение**» Использование сторонних API для расширения функционала.	**2**	
	Лабораторная работа «**Оптимизация производительности и кэширование**» Настройка кэширования шаблонов и данных для ускорения ответа сервера.	**2**	
	Лабораторная работа «**Обеспечение безопасности веб-приложения**» Внедрение мер защиты от распространённых уязвимостей (XSS, CSRF, SQL Injection).	**2**	
	Лабораторная работа «**Логирование и мониторинг».** Сбор логов запросов и ошибок, настройка alert-ов на аномалии.	**2**	
	Лабораторная работа «**Развёртывание и эксплуатация: контейнеризация**» Создание Dockerfile, сборка и запуск контейнера с Django-приложением	**2**	
	Лабораторная работа «**Развёртывание и эксплуатация: оркестрация (Kubernetes)**». Разворачивание приложения в кластере, масштабирование, обновление без простоев.	**2**	
	Лабораторная работа «**Интеграция сервисов: протоколы и форматы**». Выбрать и реализовать в проекте протокол обмена данными между сервисами (HTTP, gRPC, AMQP). Настроить сериализацию и десериализацию сообщений в форматах JSON и Protocol Buffers (protobuf). Протестировать обмен сообщениями, обеспечить корректную обработку ошибок и соответствие контракту данных.	**4**	
	Лабораторная работа «**Веб-сокеты и real-time взаимодействия**». Создать веб-приложение с двунаправленной связью между клиентом и сервером с использованием веб-сокетов. Реализовать функционал чата или системы нотификаций с обновлением данных в реальном времени. Обеспечить устойчивость соединения и обработку ошибок подключения. Проверить производительность и масштабируемость решения.	**4**	
Экзамен		**4**	

Внеаудиторная (самостоятельная) учебная работа при изучении МДК. 05.02 1. Создание приложения 2. Подключение к базе данных 1.	**26**	ОК 01, ОК 02 ОК 03, ПК 5.1. ПК 5.2., ПК 5.6

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовая работа (проект) *(если предусмотрены)*	Объем часов	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
МДК. 05.03. Тестирование информационных систем		**98**	
Тема 5.3.1. Основы тестирования программного обеспечения		**16**	
Тема 5.3.1. «Основы тестирования программного обеспечения»	**Содержание**	**12**	ОК 01, ОК 02 ОК 03, ПК 5.1. ПК 5.2., ПК 5.6
	Метрики и атрибуты оценки качества программного обеспечения. Анализ спецификаций . Понятие процесса тестирования программного обеспечения. Этапы процесса тестирования ПО. Основные определения тестирования. Цели и задачи процесса тестирования.	**2**	
	Полный цикл тестирования. (фазы тестирования, процесс тестирования на разных этапах разработки ПО). Валидация. Верификация.	**2**	
	Методы и виды тестирования. Классификация видов тестирования. Объекты тестирования.	**2**	
	Базовые инструменты тестировщика. Организация тестирования в команде разработчиков. Современные методики тестирования разрабатываемых ИС.	**2**	
	Стратегии тестирования. Стратегии тестирования: черного ящика, белого ящика. Тестирования с применением функциональных диаграмм. Тестирования методом предположения об ошибке. Предположение ошибок.	**2**	
	Понятие программной ошибки. Анализ ошибки. Модель работы с дефектами. Жизненный цикл бага. Ошибки программного обеспечения: источники ошибок ПО, классификация ошибок, основные пути борьбы с ошибками. Критерии покрытия кода программы тестами.	**2**	
	Тематика практических занятий и лабораторных работ	**4**	
	Лабораторная работа «Определение объектов тестирования. Формирование набора тестовых случаев на примере конкретного программного продукта. Создание чек-листа»	**2**	
	Лабораторная работа «Тестирование настольного приложения. Тестовая документация. Виды документации. Тест-план, чек-лист, тест-кейс Определение объектов тестирования, разработка тестового сценария проекта (тест-плана)»	**2**	

Тема 5.3.2. «Методы и принципы тестирования программного обеспечения»		**28**	
Тема 5.3.2. «Методы и принципы тестирования программного обеспечения»	**Содержание**	**14**	ОК 01, ОК 02 ОК 03, ПК 5.1. ПК 5.2., ПК 5.6
	Инструменты тестирования. Тест дизайн Позитивные и негативные тест-кейсы.	**2**	
	Техники тест-дизайна:. Эквивалентное Разделение (Equivalence Partitioning – EP); Анализ Граничных Значений (Boundary Value Analysis – BVA); Таблица принятия решений (Decision Table); Попарное тестирование (Pairwise Testing - PT); Причина / Следствие (Cause/Effect - CE); Предугадывание ошибки (Error Guessing - EG); Комбинирование параметров тест-кейса при доменном тестировании	**2**	
	Применение техник тестирования для разработки тест-кейсов с учетом особенностей физической и логической структуры. Исследовательское тестирование (Exploratory testing). Поиск причин возникновения дефектов	**4**	
	Процесс модульного тестирования, его задачи и цели. Понятие модуля и его границ. Тестовое окружение при модульном тестировании. Test Driven Development (разработка через тестирование).	**2**	
	Фреймворки для unit-тестирования. Оформление unit-теста	**2**	
	Интеграционное тестирование. Способы интеграционного тестирования. Системное тестирование.	**2**	
	Тематика практических занятий и лабораторных работ	**14**	
	Лабораторная работа «Тестирование библиотеки-классов. Определение объектов тестирования, формирование тест-кейсов»	**2**	
	Лабораторная работа «Создание библиотеки-классов. Разработка unit-тестов»	**4**	
	Лабораторная работа «Тестирование библиотеки-классов. Обработка исключительных ситуаций в unit-тестах. Документирование процесса тестирования»	**4**	
	Тестирование интеграции. Разработка интеграционных тестов в IDE. Создание и документирование unit-тестов интеграционного тестирования	**4**	
Тема 5.3.3. Тема "Автоматизация тестирования программного обеспечения"		**54**	
Тема 5.3.3. Тема "Автоматизация тестирования программного обеспечения"	**Содержание**	**22**	ОК 01, ОК 02 ОК 03, ПК 5.1. ПК 5.2., ПК 5.6
	Автоматизация тестирования. Сокращение тестовых наборов. Автоматизация приемочного и регрессионного тестирования. Основные направления автоматизации тестирования. Автоматизация построения тестов, символьное исполнение программ	**2**	
	Инструменты автоматизации тестирования. Определение направлений автоматизации. Сравнительный анализ инструментов автоматизации	**2**	
	Автоматизация тестирования при использовании различных интерфейсов (API, CLI, GUI и другие). Нефункциональное тестирование: тестирование	**6**	

	производительности, безопасности, удобства использования. Документирование процесса тестирования		
	Автоматизация тестирования. Построение минимального дугового покрытия управляющего графа программы и на его основе минимального набора тестов для критерия С1. Контроль качества набора тестов. Системы контроля полноты набора тестов для определенных критериев. Взаимодействие автотестов с тестируемой системой.	**4**	
	Подходы к автоматизации Функциональная декомпозиция. Data-driven. Keyword-driven. Behavior-driven.	**2**	
	Проект Selenium. Поддерживаемые платформы. Компоненты Selenium Webdriver. Взаимодействие с браузером и UI	**6**	
	Тематика практических занятий и лабораторных работ	**32**	
	Лабораторная работа «Автоматизация процесса тестирования. Подходы к автоматизации: Функциональная декомпозиция. Разработка unit-тестов. Документирование процесса тестирования»	**4**	
	Лабораторная работа «Автоматизация процесса тестирования. Подходы к автоматизации: Data driven testing»	**4**	
	Лабораторная работа «Автоматизация процесса тестирования. Подходы к автоматизации: Keyword driven testing»	**4**	
	Лабораторная работа «Автоматизация процесса тестирования. Подходы к автоматизации: Behavior driven testing»	**4**	
	Лабораторная работа «Автоматизация процесса тестирования. Нагрузочное тестирование. Документирование процесса тестирования»	**2**	
	Лабораторная работа «Автоматизация процесса тестирования. Стрессовое тестирование. Документирование процесса тестирования»	**2**	
	Лабораторная работа «Регрессионное тестирование. Использование инструментария анализа качества программного средства.»	**4**	
	Лабораторная работа «Разработка баг-репорта»	**4**	
	Лабораторная работа «Работа с Selenium. Установка системы. Создание автотеста.»	**4**	
Экзамен		**4**	
Внеаудиторная (самостоятельная) учебная работа при изучении МДК. 05.03 1. Создание тестовых наборов 2. Разработка модульных тестов для проверки существования пользователя в БД 3. Разработка модульных тестов для проверки корректности заполнения данных экранных форм 4. Разработка модульных тестов для проверки подключения к БД 5. Разработка интеграционных тестов по проверке добавления данных в БД 6. Разработка интеграционных тестов по проверке редактирования данных в БД		**24**	ОК 01 ОК 02 ОК 03 ПК 5.1. ПК 5.2. ПК 5.6

Курсовое проектирование 1. Общее описание системы; Цели, функции и задачи системы; Описание предметной области системы; 2. Описание модели информационной системы с помощью диаграммы прецедентов 3. Описание внутренней структуры информационной системы: описание классов (диаграмма классов), объектов, отношений между ними, внутренней структуры классов 4. Описание взаимодействия объектов (описание алгоритмов взаимодействия между объектами). Диаграммы последовательности и кооперации. Описанием функций, доступных разным группам пользователей, описание принципов разграничения доступа к данным пользователями; 5. Создание схемы базы данных (ERD-диаграммы) ; 6. Проектные решения по реализации интерфейса приложения (создание прототипа приложения); 7. Разработка руководства по стилю. 8. Создание физической модели базы данных системы в СУБД. Подключение к СУБД. Выполнение SQL-запросов для работы с СУБД; 9. Реализация модуля по разграничению доступа к информационной системе различных групп пользователей системы 10. Разработка модулей информационной системы (разработка форм по добавлению данных в БД) 11. Разработка модулей информационной системы (разработка форм по редактированию, удалению данных в БД) 12. Описание программы (краткое описание структуры программы, руководство по использованию программы, листинг программы с комментариями); 13. Формирование плана тестирования (TestPlan) – документа, описывающего весь объем работ по тестированию, начиная с описания объекта, стратегии. тестирование программы (исходные данные для тестовых прогонов программы, результаты тестирования); **14.** Описание технологии тестирования приложения. Разработка позитивных и негативных тестовых случаев (testcase) **15.** Разработка документации информационной системы Примерные темы курсового проектирования: 1. Создание и тестирование АРМ для расчета квартплаты жителей многоквартирного дома 2. Создание и тестирование АРМ работника стационара больницы (учет пациентов стационара больницы) 3. Создание и тестирование АРМ для поиска вакансий 4. Создание и тестирование АРМ для обработки данных о медицинских анализах 5. Создание и тестирование АРМ стоматолога 6. Создание и тестирование АРМ по учету заказов на ремонт (ПК и оргтехники) и учету комплектующих ПК в компьютерном магазине 7. Создание и тестирование АРМ по учету льготных категорий граждан 8. Создание и тестирование АРМ для обработки информации об участии студентов в олимпиадном движении, конкурсах профессионального мастерства 9. Создание и тестирование АРМ «Транспорт» (движение общественного транспорта) 10. Создание и тестирование АРМ работника отдела кадров компании	**30**	

11. «Создание и тестирование АРМ по учету производственных задач, выполняемых сотрудниками предприятия» 12. Создание и тестирование АРМ учета успеваемости студентов колледжа 13. Создание и тестирование АРМ регистрации постояльцев в гостинице 14. Создание и тестирование АРМ учета нагрузки преподавателей колледжа 15. Создание и тестирование АРМ риэлторской компании 16. Создание и тестирование АРМ диспетчера аэропорта 17. Создание и тестирование АРМ работника аптеки 18. Создание и тестирование АРМ библиотекаря 19. Создание и тестирование АРМ застройщика 20. Создание и тестирование АРМ фирмы, занимающейся обучением иностранным языкам 21. Создание и тестирование АРМ ресторана 22. Создание и тестирование АРМ по обслуживанию клиентов оператора сотовой связи 23. «Создание и тестирование АРМ туристического оператора» **24.** Создание и тестирование АРМ фитнес-центра		

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Материально-техническое обеспечение

Для реализации программы учебной дисциплины предусмотрена Мастерская «Программные решения для бизнеса», оснащенная в соответствии с п. 6.2.2. примерной программы по профессии/специальности, которая предполагает наличие следующего оборудования:

- Автоматизированные рабочие места обучающихся– 15 шт, оснащенные мониторами и персональными компьютерами с техническими характеристиками (Processor - IntelCore i7.Ethernet - 10/100/1000 mbps.RAM - 8GB или больше.Видеокарта с памятью 1GB или больше.SSD 256 Gb или больше), клавиатура, мышь

- Автоматизированное рабочее место преподавателя

- Проектор и мультимедийная доска;

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, рекомендуемых для использования в образовательном процессе

Перечень используемых учебных изданий, Интернет-ресурсов, дополнительной литературы

3.2.1. Печатные издания

1. Кугаевских, А. В. Проектирование информационных систем. Системная и бизнес-аналитика : учебное пособие / А. В. Кугаевских. - Новосибирск : Изд-во НГТУ, 2018. - 256 с. - ISBN 978-5-7782-3608-0.
2. Методология и технология проектирования информационных систем : учебное пособие / Ю. М. Казаков, А. А. Тищенко, А. А. Кузьменко [и др.]. - Москва : ФЛИНТА, 2018. - 136 с. - ISBN 978-5-9765-4013-2.
3. Орлов, С. А. Технологии разработки программного обеспечения : учебник для вузов / С. А. Орлов, Б. Я. Цилькер. - 4-е изд. - Санкт-Петербург : Питер, 2021. - 608 с. - (Стандарт третьего поколения). - ISBN 978-5-4461-9773-6.
4. Брежнев, Р. В. Методы и средства проектирования информационных систем и технологий : учебное пособие / Р. В. Брежнев. - Красноярск : Сиб. федер. ун-т, 2021. - 216 с. - ISBN 978-5-7638-4416-0.
5. Лисяк, В. В. Разработка информационных систем : учебное пособие / В. В. Лисяк ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2019. - 96 с.
6. Орлов, С. А. Технологии разработки программного обеспечения : учебник для вузов / С. А. Орлов, Б. Я. Цилькер. - 4-е изд. - Санкт-Петербург : Питер, 2021. - 608 с.
7. Гагарина Л.Г., Кокорева Е.В., Сидорова-Виснадул Б.Д. Технология разработки программного обеспечения.- М: Издательство: Инфра-М, 2022
8. Галина Иванова. Технология программирования. Проектирование. Комплексное тестирование. Программная документация. Издательство: Кнорус, 2022 г.

Дополнительная литература:

1. Шитов, В. Н. Разработка информационного контента (по отраслям) : учебное пособие / В.Н. Шитов. — Москва : ИНФРА-М, 2022. — 178 с.

2. Кузниченко, М. А. Основы баз данных : учебно-методическое пособие / М. А. Кузниченко. - 2-е изд., стер. - Москва : Флинта, 2022. - 102 с. - ISBN 978-5-9765-5139-8.

3. Морозова, Ю. В. Тестирование программного обеспечения : учебное пособие / Ю. В. Морозова. - Томск : Эль-Контент, 2019.

4. Стружкин, Н. П. Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2022. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9.

5. Галиаскаров, Э. Г. Анализ и проектирование систем с использованием UML : учебное пособие для вузов / Э. Г. Галиаскаров, А. С. Воробьев. — Москва : Издательство Юрайт, 2022. — 125 с. — (Высшее образование). — ISBN 978-5-534-14903-6.

6. Лаврищева, Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. — Москва : Издательство Юрайт, 2022. — 280 с. — (Высшее образование). — ISBN 978-5-534-01056-5.

3.2.2. Электронные издания (электронные ресурсы)

1. Тестирование программного обеспечения. Базовый курс [Электронный ресурс] URL: https://svyatoslav.biz/software_testing_book/ (дата обращения: 21.02.2024г.).

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ (ПО РАЗДЕЛАМ)

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
Раздел модуля 1.Технологии проектирования и дизайн информационных систем		
ПК 5.1 Собирать исходные данные для разработки проектной документации на информационную систему.	Оценка «**отлично**» - сформулирована задача по обработке информации; выполнен анализ предметной области; выполнены сбор и обработка исходной информации с помощью инструментальных средств. Оценка «**хорошо**» - сформулирована задача по обработке информации; выполнен анализ предметной области; собрана исходная информация; выполнена обработка исходной информации с помощью инструментальных средств. Оценка «**удовлетворительно**» - сформулирована задача по обработке информации; выполнен анализ предметной области; собрана исходная информация; частично выполнена обработка исходной информации с помощью инструментальных средств.	Экзамен: практическое задание по постановке задачи по обработке информации в заданной сфере деятельности, анализу предметной области, сбору и обработке исходной информации Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик
ПК 5.2 Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.	Оценка «**отлично**» - требования клиента проанализированы, предложен и обоснован математический алгоритм решения задачи по обработке информации; указаны стандарты на оформление алгоритмов; предложенный алгоритм оформлен в соответствии с требованиями стандартов. Оценка «**хорошо**» - требования клиента проанализированы, предложен математический алгоритм решения задачи по обработке информации; предложенный алгоритм оформлен в соответствии с требованиями стандартов. Оценка «**удовлетворительно**» - требования клиента проанализированы, предложен математический алгоритм решения задачи по обработке информации; предложенный алгоритм оформлен в соответствии с требованиями стандартов с некоторыми отклонениями.	Экзамен: практическое задание по анализу интересов клиента (изложенным в задании); разработке и оформлению алгоритма решения задачи по обработке информации Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик

ПК 5.6 Разрабатывать техническую документацию на эксплуатацию информационной системы.	Оценка «**отлично**» - разработанные документы по содержанию и оформлению полностью соответствуют стандартам; содержание отдельных разделов хорошо структурировано, логически увязано, проиллюстрировано диаграммами и схемами; терминология полностью соответствует принятой в соответствующей области профессиональной терминологии. Оценка «**хорошо**» - разработанные документы по содержанию и оформлению соответствуют стандартам; содержание отдельных разделов логически увязано, проиллюстрировано диаграммами и схемами; терминология соответствует принятой в соответствующей области профессиональной терминологии. Оценка «**удовлетворительно**» - разработанные документы по содержанию и оформлению соответствуют стандартам с незначительными отклонениями; содержание отдельных разделов проиллюстрировано диаграммами и схемами; терминология соответствует общепринятой	Экзамен: практическое задание по разработке технической документации на эксплуатацию информационной системы (или отдельных документов). Защита отчетов по практическим и лабораторным работам. Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик
ПК 5.7 Производить оценку информационной системы для выявления возможности ее модернизации.	Оценка «**отлично**» - определены и обоснованы критерии для оценки качества информационной системы; выполнена оценка качества информационной системы в соответствии с выбранными критериями; определены конкретные направления модернизации. Оценка «**хорошо**» - определены и обоснованы критерии для оценки качества информационной системы; выполнена оценка качества информационной системы в соответствии с выбранными критериями; определены общие направления модернизации. Оценка «**удовлетворительно**» - определены основные критерии для оценки качества информационной системы; выполнена оценка качества информационной системы в соответствии с выбранными критериями; определены некоторые направления модернизации.	Экзамен: практическое задание по оценке качества предложенной информационной системы Защита отчетов по практическим и лабораторным работам. Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик

Раздел модуля 2. Инструментарий и технологии разработки кода информационных систем		
ПК 5.1 Собирать исходные данные для разработки проектной документации на информационную систему.	Оценка «**отлично**» - сформулирована задача по обработке информации; выполнен анализ предметной области; выполнены сбор и обработка исходной информации с помощью инструментальных средств. Оценка «**хорошо**» - сформулирована задача по обработке информации; выполнен анализ предметной области; собрана исходная информация; выполнена обработка исходной информации с помощью инструментальных средств. Оценка «**удовлетворительно**» - сформулирована задача по обработке информации; выполнен анализ предметной области; собрана исходная информация; частично выполнена обработка исходной информации с помощью инструментальных средств.	Экзамен : практическое задание по постановке задачи по обработке информации в заданной сфере деятельности, анализу предметной области, сбору и обработке исходной информации Защита отчетов по практическим и лабораторным работам. Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик
ПК 5.2 Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.	Оценка «**отлично**» - требования клиента про анализированы, предложен и обоснован математический алгоритм решения задачи по обработке информации; указаны стандарты на оформление алгоритмов; предложенный алгоритм оформлен в соответствии с требованиями стандартов. Оценка «**хорошо**» - требования клиента про-анализированы, предложен математический алгоритм решения задачи по обработке информации; предложенный алгоритм оформлен в соответствии с требованиями стандартов. Оценка «**удовлетворительно**» - требования клиента проанализированы, предложен математический алгоритм решения задачи по обработке информации; предложенный алгоритм оформлен в соответствии с требованиями стандартов с некоторыми отклонениями.	Экзамен в форме собеседования: практическое задание по анализу интересов клиента (изложенным в задании);разработке и оформлению алгоритма решения задачи по обработке информации Защита отчетов по практическим и лабораторным работам. Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик

ПК 5.3 Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием.	Оценка «**отлично**» - разработан проект подсистемы безопасности информационной системы, в спецификации отражены задачи проекта в полном объеме. Оценка «**хорошо**» - разработан проект подсистемы безопасности информационной системы, в спецификации отражены основные задачи проекта. Оценка «**удовлетворительно**» - разработан проект подсистемы безопасности информационной системы, в спецификации отражены задачи проекта с некоторыми недочетами.	Экзамен: практическое задание по разработке проекта (подсистемы) по обеспечению безопасности информационной системы. Защита отчетов по практическим и лабораторным работам. Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик
ПК 5.4 Производить разработку модулей информационной системы в соответствии с техническим заданием.	Оценка «**отлично**» - разработаны варианты возможных решений, выбран и обоснован оптимальный на основе анализа интересов клиента; разработаны модули информационной системы; при разработке использованы языки структурного, объектно-ориентированного программирования и языка сценариев; разработана документация на модули (по перечню в задании); выполнена оценка качества разработанных модулей по выбранным и обоснованным метрикам. Оценка «**хорошо**» - разработан и обоснован вариант возможного решения, на основе анализа интересов клиента; разработаны модули информационной системы; при разработке использованы языки структурного, объектно-ориентированного программирования и языка сценариев; разработана документация на модули (по перечню в задании); выполнена оценка качества разработанных модулей по набору метрик. Оценка «**удовлетворительно**» - разработан вариант возможного решения; разработаны модули информационной системы; при разработке использованы языки структурного, объектно-ориентированного программирования и языка сценариев; разработана документация на модули (по перечню в задании); выполнена оценка качества разработанных модулей по набору метрик.	Экзамен: практическое задание по разработке модулей информационной системы, документации на разработанные модуле и оценке их качества. Защита отчетов по практическим и лабораторным работам. Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик
Раздел модуля 3.Методы и средства тестирования информационных систем		

ПК 5.1 Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.	Оценка «**отлично**» - требования клиента про анализированы, предложен и обоснован математический алгоритм решения задачи по обработке информации; указаны стандарты на оформление алгоритмов; предложенный алгоритм оформлен в соответствии с требованиями стандартов. Оценка «**хорошо**» - требования клиента про-анализированы, предложен математический алгоритм решения задачи по обработке информации; предложенный алгоритм оформлен в соответствии с требованиями стандартов. Оценка «**удовлетворительно**» - требования клиента проанализированы, предложен математический алгоритм решения задачи по обработке информации; предложенный алгоритм оформлен в соответствии с требованиями стандартов с некоторыми отклонениями.	Экзамен: практическое задание по анализу интересов клиента (изложенным в задании); разработке и оформлению алгоритма решения задачи по обработке информации Защита отчетов по практическим и лабораторным работам. Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик
ПК 5.5 Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы.	Оценка «**отлично**» - выбраны и обоснованы методики тестирования информационной системы; информационная система протестирована в соответствии с выбранными методами в полном объеме; в результате тестирования выявлены и зафиксированы ошибки кодирования; результаты тестирования оформлены в соответствии с рекомендованными нормативными документами. Оценка «**хорошо**» - выбраны и обоснованы методики тестирования информационной системы; информационная система протестирована в соответствии с выбранными методами в достаточном объеме; в результате тестирования выявлены ошибки кодирования; результаты тестирования оформлены в соответствии с рекомендованными нормативными документами. Оценка «**удовлетворительно**» - выбраны методики тестирования информационной системы; информационная система протестирована в соответствии с в достаточном объеме; в результате тестирования выявлены ошибки кодирования; результаты тестирования зафиксированы.	Экзамен: практическое задание по тестированию информационной системы. Защита отчетов по практическим и лабораторным работам. Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик

ПК 5.6 Разрабатывать техническую документацию на эксплуатацию информационной системы.	Оценка «**отлично**» - разработанные документы по содержанию и оформлению полностью соответствуют стандартам; содержание отдельных разделов хорошо структурировано, логически увязано, проиллюстрировано диаграммами и схемами; терминология полностью соответствует принятой в соответствующей области профессиональной терминологии. Оценка «**хорошо**» - разработанные документы по содержанию и оформлению соответствуют стандартам; содержание отдельных разделов логически увязано, проиллюстрировано диаграммами и схемами; терминология соответствует принятой в соответствующей области профессиональной терминологии. Оценка «**удовлетворительно**» - разработанные документы по содержанию и оформлению соответствуют стандартам с незначительными отклонениями; содержание отдельных разделов проиллюстрировано диаграммами и схемами; терминология соответствует общепринятой.	Экзамен: практическое задание по разработке технической документации на эксплуатацию информационной системы (или отдельных документов). Защита отчетов по практическим и лабораторным работам. Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практик

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

ОК 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, ресурс сбережению, 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 осознания и укрепления здоровья в процессе профессиональной деятельности и подражания необходимого уровня физической подготовленности.	- эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.	
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.	- эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке.	

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
МДК 05.01 ПРОЕКТИРОВАНИЕ И ДИЗАЙН ИНФОРМАЦИОННЫХ СИСТЕМ

для специальности

09.02.07 «Информационные системы и программирование»

Екатеринбург
2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данный фонд оценочных средств разработан для дисциплины МДК 05.01 «Проектирование и дизайн информационных систем», направленной на формирование профессиональных компетенций в области проектирования и разработки информационных систем.

Основная цель пособия — обеспечение качественного освоения студентами теоретических знаний и практических навыков в сфере проектирования информационных систем, включая методы анализа, моделирования и документирования.

Структура пособия

Материал делится на две основные части:

1. **Теоретическая часть** — посвящена изучению основ проектирования ИС, методологий разработки, принципов моделирования и документирования.
2. **Практическая часть** — направлена на закрепление полученных знаний через выполнение практических работ и индивидуальных заданий. Каждая практическая работа сопровождается примером, образцом решения и рекомендациями по проверке результата.

Учебные цели и задачи

Основные учебные цели включают:

- Формирование системного подхода к проектированию ИС.
- Освоение современных методологий разработки.
- Развитие навыков моделирования бизнес-процессов.
- Изучение принципов создания качественной проектной документации.

Задачи курса состоят в следующем:

- Освоение методов анализа предметной области.
- Изучение инструментов проектирования ИС.
- Формирование навыков работы с CASE-средствами.
- Развитие умений в области документирования проектов.

Контроль знаний

Контроль знаний осуществляется поэтапно:

- Текущие проверки знаний в форме вопросов и опросов.
- Практические работы с последующим обсуждением.
- Итоговая аттестация в виде экзамена, содержащего практические задания.

Оценочная деятельность

Оценка проводится по следующим критериям:

- Качество выполнения практических работ.
- Уровень владения материалами курса.
- Активность и инициативность на занятиях.
- Результативность индивидуальной проектной работы.

Краткое изложение планов хода лекционных занятий:

1. **Введение в проектирование ИС**
   Образовательный результат: Понимает основные понятия и определения ИС, жизненный цикл информационных систем, модели проектирования, принципы разработки и функционирования информационных систем.
2. **Основы системного анализа**
   Образовательный результат: Владеет методами сбора и обработки требований, умеет проводить предпроектное обследование предприятия, анализировать предметную область и интересы клиента, определять организационно-функциональную структуру.
3. **CASE-средства проектирования**
   Образовательный результат: Владеет инструментами моделирования функциональных требований, умеет работать с CASE-средствами, создавать IDEF0-диаграммы и UML-модели.
4. **Проектирование интерфейсов**
   Образовательный результат: Понимает основные принципы построения пользовательских интерфейсов, умеет создавать прототипы и макеты, владеет методами проектирования интерфейсов.
5. **Система обеспечения качества ИС**
   Образовательный результат: Знает стандарты качества ИС, умеет применять методы контроля качества, понимает принципы сертификации и стандартизации информационных систем.
6. **Разработка документации ИС**
   Образовательный результат: Владеет навыками разработки технической документации, умеет составлять ТЗ и проектную документацию, знает стандарты оформления программной документации.

Краткое изложение планов хода практических занятий и примерные задания:

Анализ компонентов информационной системы. Исследование структуры и взаимодействия компонентов ИС на примере реальных кейсов.

- Ход занятия: изучение основных компонентов информационной системы, анализ их взаимодействия и взаимосвязей, исследование механизмов обмена данными между подсистемами на примере конкретной ИС, построение схемы компонентов и их взаимосвязей, анализ документации и технических спецификаций.
- Задание: провести анализ компонентов информационной системы на примере CRM-системы, составить схему компонентов с указанием их функций и взаимодействия, описать механизмы обмена данными между компонентами, подготовить отчет со структурой системы и описанием основных подсистем.

Модели жизненного цикла ИС. Сравнительный анализ каскадной, спиральной и итерационной моделей проектирования.

- Ход занятия: изучение особенностей построения и применения каскадной, спиральной и итерационной моделей, анализ их преимуществ и недостатков, рассмотрение критериев выбора модели для разных проектов.
- Задание: построить схемы трёх моделей жизненного цикла, сравнить их характеристики на конкретном примере, обосновать выбор модели для заданного проекта.

Предпроектное обследование предприятия. Проведение анализа предметной области и составление отчета.

- Ход занятия: освоение методов сбора и анализа информации о предприятии, изучение способов выявления требований, формирование структуры отчетной документации.
- Задание: провести обследование условной организации, составить отчет с анализом предметной области и рекомендациями.

Построение организационно-функциональной структуры. Разработка структуры компании для цифровой трансформации.

- Ход занятия: изучение принципов построения оргструктуры, анализ бизнес-процессов, разработка стратегии цифровой трансформации, создание схем организационной структуры.
- Задание: разработать структуру организации с учетом цифровой трансформации, описать функции подразделений.

Сбор и анализ требований к ИС. Методы выявления и документирования требований заказчика.

- Ход занятия: освоение техник сбора требований, изучение методов их классификации и документирования, анализ противоречий.
- Задание: провести сбор требований к ИС, составить документ с их описанием и классификацией.

Работа с IDEF0-диаграммами. Построение контекстной диаграммы и диаграмм декомпозиции.

- Ход занятия: изучение правил построения IDEF0-диаграмм, освоение техники декомпозиции, создание моделей верхнего уровня.
- Задание: построить контекстную диаграмму и диаграммы декомпозиции для заданной системы.

Работа с IDEF0-диаграммами. Построение диаграмм декомпозиции.

- Ход занятия: углубленное изучение декомпозиции процессов, построение диаграмм нижнего уровня, анализ взаимосвязей между элементами.
- Задание: создать детализированные диаграммы декомпозиции для выбранных процессов.

Язык UML. Построение диаграммы последовательности.

- Ход занятия: изучение синтаксиса UML, освоение построения диаграмм последовательности, анализ взаимодействий объектов.
- Задание: построить диаграмму последовательности для заданного бизнес-процесса.

Язык UML. Построение диаграммы объектов.

- Ход занятия: изучение структуры и элементов диаграммы объектов, создание статической модели системы.
- Задание: разработать диаграмму объектов для выбранной информационной системы.

Язык UML. Построение диаграммы классов.

- Ход занятия: изучение структуры классов, создание иерархий, определение отношений между классами.
- Задание: построить диаграмму классов для заданной системы.

Язык UML. Построение диаграмм вариантов использования.

- Ход занятия: изучение прецедентов системы, создание диаграмм вариантов использования, анализ функциональных требований.
- Задание: разработать диаграмму прецедентов для информационной системы.

Язык UML. Построение диаграммы деятельности.

- Ход занятия: изучение потоков работ, создание диаграмм деятельности, моделирование бизнес-процессов.
- Задание: построить диаграмму деятельности для заданного процесса.

Язык UML. Построение диаграммы кооперации.

- Ход занятия: изучение взаимодействий объектов, построение диаграмм кооперации, анализ ролей участников.
- Задание: создать диаграмму кооперации для выбранной системы.

Проектирование пользовательских интерфейсов. Работа с современными инструментами создания макетов интерфейсов.

- Ход занятия: освоение инструментов прототипирования, создание макетов

интерфейсов, тестирование прототипов.
- Задание: разработать прототип пользовательского интерфейса.

Разработка прототипов интерфейсов с учетом принципов юзабилити.
- Ход занятия: применение принципов юзабилити, создание интерактивных прототипов, оценка удобства использования.
- Задание: разработать интерактивный прототип интерфейса.

Стандарты качества ИС. Изучение требований национальных и международных стандартов.
- Ход занятия: анализ стандартов качества, изучение требований к ИС, освоение методов контроля качества.
- Задание: изучить стандарты качества для заданной системы, составить план контроля.

Метрики качества. Оценка показателей эффективности ИС.
- Ход занятия: выбор метрик качества, сбор и анализ данных, оценка эффективности системы.
- Задание: определить метрики для оценки качества ИС.

Анализ рисков и характеристик качества программного обеспечения.
- Ход занятия: идентификация рисков, оценка вероятности и влияния, разработка мер по снижению рисков.
- Задание: составить реестр рисков для проекта, разработать план реагирования.

Разработка требований безопасности информационной системы.
- Ход занятия: анализ угроз безопасности, формирование требований к защите, разработка механизмов безопасности.
- Задание: разработать требования безопасности для ИС.

Разработка технического задания на проектирование.
- Ход занятия: изучение структуры ТЗ, освоение составления разделов, оформление документации.
- Задание: составить техническое задание на разработку ИС.

Разработка технических условий эксплуатации ИС. Определение состава оборудования и программных средств разработки информационной системы.
- Ход занятия: выбор оборудования и ПО, составление технических условий, анализ требований к эксплуатации.
- Задание: разработать технические условия для ИС.

Разработка технических условий эксплуатации ИС. Определения состава оборудования и программных средств разработки информационной системы.
- Ход занятия: изучение требований к оборудованию и ПО, анализ совместимости компонентов, составление технических условий эксплуатации.
- Задание: составить перечень необходимого оборудования и ПО для заданной ИС.

Разработка проектной документации на информационную систему.
- Ход занятия: изучение структуры проектной документации, освоение правил оформления, составление основных разделов.
- Задание: разработать проектную документацию для информационной системы.

Разработка пользовательской документации на информационную систему.
- Ход занятия: изучение структуры пользовательской документации, освоение методов описания интерфейсов, составление руководств пользователя.
- Задание: составить руководство пользователя для информационной системы.

Проектирование спецификации информационной системы по индивидуальному заданию.
- Ход занятия: изучение требований к спецификации, анализ функциональных требований, составление технической спецификации.
- Задание: разработать спецификацию ИС по индивидуальному заданию.

Разработка документации по безопасности информационной системы.

- Ход занятия: изучение требований безопасности, анализ угроз, составление документации по защите информации.
- Задание: разработать документацию по обеспечению безопасности ИС.

Разработка документации по управлению качеством информационной системы.

- Ход занятия: изучение стандартов качества, освоение методов контроля, составление плана обеспечения качества.
- Задание: разработать план управления качеством для ИС.

Разработка руководства по инсталляции программного средства индивидуальному заданию.

- Ход занятия: изучение процесса установки ПО, освоение составления инструкций, оформление документации по инсталляции.
- Задание: составить руководство по установке программного продукта.

Разработка руководства пользователя программного средства по индивидуальному заданию.

- Ход занятия: изучение структуры руководства пользователя, освоение методов описания функционала, составление инструкций.
- Задание: разработать руководство пользователя для конкретного ПО.

Оформление сертификата на информационную систему.

- Ход занятия: изучение требований к сертификации, освоение оформления документов, подготовка комплекта сертификационной документации.
- Задание: подготовить документы для получения сертификата на ИС.

Варианты практических задач к экзамену

1. Провести анализ компонентов информационной системы для интернет-магазина. Составить схему взаимодействия подсистем и описать их функции.
2. Выполнить сравнительный анализ каскадной и спиральной моделей жизненного цикла для разработки банковской системы.
3. Провести предпроектное обследование предприятия торговли. Составить отчет с рекомендациями по автоматизации.
4. Разработать организационно-функциональную структуру компании с учетом цифровой трансформации производственного предприятия.
5. Собрать и документировать требования к ИС для системы управления персоналом.
6. Построить контекстную IDEF0-диаграмму и диаграммы декомпозиции для системы учета товаров.
7. Создать детальную IDEF0-диаграмму декомпозиции для процесса обработки заказов.
8. Разработать диаграмму последовательности для процесса оформления покупки в ИС.
9. Построить диаграмму объектов для системы управления складом.
10. Создать диаграмму классов для модуля управления пользователями.
11. Разработать диаграмму вариантов использования для системы бронирования.
12. Построить диаграмму деятельности для процесса обработки заявки.
13. Создать диаграмму кооперации для взаимодействия компонентов платежной системы.
14. Разработать прототип пользовательского интерфейса для системы учета рабочего времени.
15. Создать интерактивный прототип интерфейса с учетом принципов юзабилити для образовательного портала.
16. Проанализировать соответствие ИС требованиям национальных стандартов качества.
17. Оценить метрики эффективности ИС для системы управления проектами.
18. Провести анализ рисков для информационной системы обработки персональных данных.
19. Разработать требования безопасности для ИС государственного учреждения.
20. Составить техническое задание на разработку системы электронного документооборота.

21. Разработать технические условия эксплуатации ИС для производственного предприятия.
22. Создать проектную документацию для системы управления торговлей.
23. Разработать пользовательскую документацию для модуля отчетности.
24. Составить спецификацию требований к ИС для автоматизации бухгалтерии.
25. Разработать документацию по безопасности для банковской системы.
26. Создать план управления качеством для ИС образовательного учреждения.
27. Разработать руководство по инсталляции программного продукта.
28. Составить руководство пользователя для системы управления проектами.
29. Подготовить комплект документов для сертификации ИС.
30. Разработать план миграции данных при внедрении новой ИС.
31. Создать план тестирования компонентов ИС.
32. Разработать схему резервного копирования для ИС.
33. Составить план обновления программного обеспечения ИС.
34. Разработать план интеграции ИС с внешними системами.
35. Создать план обучения пользователей работе с новой ИС.

Критерии оценивания:

Критерии оценивания практических заданий

1. **Корректность анализа компонентов ИС**

- Полное соответствие структуры ИС реальным требованиям
- Правильность определения взаимосвязей между компонентами
- Точность описания функций каждого элемента системы

2. **Качество моделирования**

- Правильность построения диаграмм
- Соответствие нотаций стандартам
- Полнота отражения бизнес-процессов

3. **Соответствие методологии**

- Правильное применение выбранной модели жизненного цикла
- Корректность построения организационной структуры
- Соответствие стандартам проектирования

4. **Полнота требований**

- Всесторонность сбора информации
- Четкость формулировок требований
- Учет интересов всех заинтересованных сторон

5. **Качество документации**

- Структурированность материалов
- Полнота описания всех аспектов
- Соответствие стандартам оформления

6. **Практическая реализуемость**

- Реалистичность предложенных решений
- Учет ограничений системы
- Возможность внедрения разработанных компонентов

7. **Эффективность решений**

- Оптимизация использования ресурсов
- Рациональность выбранных подходов
- Минимизация рисков

8. **Соблюдение сроков**

- Выполнение задания в установленные рамки
- Своевременное предоставление промежуточных результатов
- Пунктуальность при сдаче работы

9. **Качество презентации**

- Четкость изложения материала
- Логичность подачи информации
- Умение отвечать на вопросы

10. **Самостоятельность выполнения**

- Оригинальность подходов
- Независимость в принятии решений
- Способность к критическому мышлению

Балльная шкала оценивания:

- **«Отлично» (оценка 5)** — Выполнено верно, с высоким качеством и эффективностью, грамотно оформленный и оригинальный код, используется современный подход.
- **«Хорошо» (оценка 4)** — Правильное выполнение задания, присутствуют некоторые недочёты, допускаются несущественные ошибки, умеренно качественный код.
- **«Удовлетворительно» (оценка 3)** — Основное задание выполнено, имеются значительные погрешности, низкая эстетичность и качество оформления, небольшой процент оригинальных решений.
- **«Неудовлетворительно» (оценка 2)** — Значительная часть задания не выполнена, большое количество ошибок, недопустимо низкое качество оформления и низкий уровень соответствия требованиям.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
МДК. 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ

для специальности
09.02.07 «Информационные системы и программирование»

Екатеринбург
2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данный фонд оценочных средств разработан для дисциплины «МДК. 05.02 Разработка кода информационных систем», направленной на формирование профессиональных компетенций в области разработки информационных систем.

Основная цель пособия — обеспечение качественного освоения студентами теоретических знаний и практических навыков в сфере проектирования информационных систем, включая методы анализа, моделирования и документирования.

Структура пособия

Материал делится на две основные части:

Теоретическая часть

Посвящена изучению ключевых понятий, принципов и моделей, которые служат основой для дальнейшего практического освоения темы. Включает обзор теоретических материалов, норм и стандартов, описание архитектурных подходов, паттернов проектирования, методологий, а также технических основ серверной разработки, безопасности и развертывания. Цель — формирование глубинного понимания процесса разработки информационных систем, подготовка к компетентному решению реальных задач.

Практическая часть

Направлена на закрепление теоретических знаний через выполнение лабораторных работ, практических заданий и проектов. Каждое практическое задание сопровождается детальным описанием плана выполнения, примерами и образцами решений, а также рекомендациями по самостоятельной проверке и оценке результата. Практика позволяет отработать навыки настройки среды, работы с кодом, взаимодействия с базами данных, создания API и безопасного развертывания приложений.

Учебные цели и задачи

Основные учебные цели включают:

- Формирование системного и комплексного подхода к разработки серверных и клиент-серверных приложений с использованием современных технологий и методологий.
- Освоение принципов архитектуры, проектирования, безопасности и развёртывания серверных программных систем.
- Развитие навыков построения масштабируемых, отказоустойчивых и поддерживаемых веб-приложений и API.
- Изучение передачи, обработки и хранения данных, интеграции сервисов и применения асинхронных технологий.

Задачи курса:

- Освоение практических и теоретических основ архитектурных моделей (одно-, двух-, трёх- и многоуровневых систем).
- Изучение объектно-ориентированных и структурных подходов, паттернов проектирования и методологий разработки.
- Совершенствование умений работы в средах разработки, системах контроля версий и управления конфигурациями.
- Формирование навыков создания и тестирования REST API, аутентификации и авторизации пользователей.
- Изучение методов контейнеризации, оркестрации и обеспечения безопасности микросервисов.

Контроль знаний

Контроль знаний осуществляется поэтапно:

- Текущие проверки в виде опросов и вопросов в конце занятий для оценки усвоения теории.
- Практические лабораторные работы с пошаговыми заданиями, примерами решений и обсуждением результатов.
- Итоговая аттестация в форме экзамена, сочетающего теоретические вопросы и практические задачи на проектирование и реализацию серверных компонентов.

Оценочная деятельность

Оценка проводится по следующим критериям:

- Качество и полнота выполнения практических лабораторных работ и индивидуальных проектов.
- Глубина понимания теории и уверенность в применении архитектурных и программных решений.
- Активное участие в обсуждениях, опросах и групповых задачах.
- Результативность и творческий подход при реализации курсового или итогового проекта, включая соблюдение стандартов кодирования и документирования.

Краткое изложение планов хода практических занятий и примерные задания

Среды и платформы разработки

Ход занятия: Во время занятия студенты знакомятся с компонентами среды разработки, включая редактор кода, систему сборки, отладки и деплоймента. Особое внимание уделяется кроссплатформенности, поддержке различных технологий и возможностям автоматизации процессов разработки. Рассматриваются преимущества и недостатки популярных сред, их настройка и интеграция с инструментами управления проектом.

Задание: Настроить выбранную среду разработки, создать минимальный проект серверного приложения, выполнить его сборку, запуск и отладку. Отследить процесс деплоя и продемонстрировать базовую работу с инструментами IDE.

Клиент-серверная архитектура

Ход занятия: Модели распределения вычислений и данных: одноуровневая, двухуровневая, трехуровневая и многоуровневая. Обсуждается устройство различных архитектурных моделей: от простейших одностраничных приложений до сложных многоуровневых систем. Выделяются особенности, преимущества и ограничения каждой модели, примеры применения. Анализируется распределение логики и данных по уровням, что обеспечивает масштабируемость и улучшает поддержку приложений.

Задание: Рассмотреть конкретный пример клиент-серверного приложения, определить его архитектурный уровень, составить подробную схему компонентов и описать взаимодействие между слоями.

Роли компонентов приложения

Ход занятия: Распределение ответственности между интерфейсом, бизнес-логикой и данными. Студенты изучают назначение каждого слоя сервера, методы организации взаимодействия компонентов, принципы модульности и раздельной ответственности. Рассматриваются подходы обеспечения качества, надежности и производительности через четкое разделение задач.

Задание: Проектировать структуру серверной части небольшого приложения, распределить функции между слоями, описать поток данных и обработку.

Сервисно-ориентированные архитектуры

Ход занятия: Рассматриваются концепции построения сервисных систем, преимущества разделения на независимые компоненты, проблемы масштабирования и безопасности. Обсуждается влияние SOA и микросервисов на архитектуру приложений.

Задание: Спроектировать упрощенный сервисный API, определить интерфейсы взаимодействия между компонентами, описать сценарии масштабируемости.

Объектно-ориентированные и структурные подходы

Ход занятия: Анализируются основные концепции ООП и структурного программирования, обсуждается разница подходов и их применение при проектировании серверных систем для достижения гибкости и повторного использования.

Задание: Реализовать на псевдокоде набор классов и структур для решения задачи серверной логики с использованием ООП.

Паттерны проектирования

Ход занятия: Изучаются шаблоны проектирования для решения распространенных задач программирования, рассматриваются их преимущества и примеры реализации в серверных приложениях.

Задание: Применить паттерн MVC при разработке небольшого API-интерфейса, реализовать основные компоненты.

Методологии и структура проектов

Ход занятия: Рассматриваются подходы к разработке программных проектов, методы интеграции изменений, важность структурирования кода, роли в команде и автоматизация процессов.

Задание: Создать структуру репозитория, настроить рабочий процесс разработки с использованием CI/CD инструментов.

Стандарты оформления документации

Ход занятия: Ознакомление с требованиями к документации, форматами описания интерфейсов (OpenAPI/Swagger), стандартами оформления кода и технических заданий.

Задание: Подготовить документацию для API с использованием OpenAPI, оформить описание методов, параметров и ответов.

Системы контроля версий

Ход занятия: Демонстрация создания и объединения веток, разрешения конфликтов, организационные аспекты совместной работы и политики доступа.

Задание: Создать несколько веток, внести изменения, слить изменения без конфликтов и с ними, настроить права доступа к репозиторию.

Управление конфигурациями

Ход занятия: Рассмотрение способов изоляции конфигураций от кода для удобства смены окружений и воспроизводимости развертывания.

Задание: Создать конфигурации для разных сред (dev, prod), реализовать загрузку параметров через переменные окружения.

Взаимодействие с базой данных

Ход занятия: Обсуждение хранения данных, различий реляционных и NoSQL-баз, типичных операций вставки, обновления, удаления и выборки данных.

Задание: Спроектировать и реализовать CRUD-операции для выбранной сущности, оформить интерфейс взаимодействия с базой.

ORM: концепции и применение

Ход занятия: Ознакомление с преимуществами ORM, примеры реализации, обсуждение случаев применения и ограничений.

Задание: Разработать модели и выполнять выборки, вставки и обновления через ORM-интерфейс.

Работа с REST API

Ход занятия: Проектирование RESTful API, построение маршрутов, обработка запросов и ответов, использование JSON и других форматов, тестирование с Postman.

Задание: Создать комплекс API для CRUD-сущности, протестировать все методы и обработку ошибок.

Аутентификация

Ход занятия: Анализ протоколов, практическая настройка системы аутентификации, управление токенами и сессиями.

Задание: Реализовать JWT-аутентификацию для API, проверить работу с разными пользователями.

Авторизация

Ход занятия: Обсуждение схем контроля доступа, реализация проверки прав на уровне ресурсов и методов.

Задание: Настроить роли пользователей, ограничить или разрешить доступ к определенным ресурсам API.

Безопасность микросервисов

Ход занятия: Методы обеспечения конфиденциальности и целостности данных в распределенных системах.

Задание: Внедрить шифрование коммуникаций между сервисами, протестировать уязвимости.

Асинхронное программирование

Ход занятия: Рассмотрение подходов async/await, событийных циклов, преимуществ параллелизма в серверных приложениях.

Задание: Реализовать асинхронную обработку нескольких запросов, замерить и проанализировать производительность.

Контейнеризация приложений

Ход занятия: Практическое создание Dockerfile, сборка и отладка контейнеров с серверным приложением.

Задание: Написать Dockerfile для приложения, собрать образ и запустить контейнер локально.

Оркестрация контейнеров

Ход занятия: Знакомство с основами Kubernetes, развёртывание, балансировка нагрузки, обновление без простоев.

Задание: Развернуть приложение в Kubernetes, настроить автоскейлинг и выполнить обновление.

Сессии и состояние пользователей

Ход занятия: Механизмы хранения пользовательской информации между запросами, обработка сессий.

Задание: Реализовать сессионное хранение данных, обеспечить безопасность и корректность управления

Логирование и мониторинг

Ход занятия: Настройка систем логирования, настройка alert’ов, мониторинг производительности.

Задание: Подключить библиотеку логирования, реализовать сбор и анализ логов, настроить простые оповещения.

Балансировка нагрузки и масштабируемость

Ход занятия: Разбор принципов балансировки, обеспечение отказоустойчивости и масштабируемости.

Задание: Смоделировать балансировку нагрузки для сервера, оценить производительность при увеличении нагрузки.

Интеграция сервисов: протоколы и форматы

Ход занятия: Рассмотрение характеристик протоколов, библиотек и форматов обмена.

Задание: Реализовать обмен сообщениями по выбранному протоколу с сериализацией данных в JSON или protobuf, проверить корректность взаимодействия.

Работа с веб-сокетами

Ход занятия: Конфигурация веб-сокетов, управление соединениями и событиями, устойчивость соединения.

Задание: Создать чат или систему уведомлений с обновлением в реальном времени, обработать отключение и переподключение клиента.

Комплект оценочных средств для проведения экзамена

ЗАДАНИЕ

В одном из театров города X, некий таинственный путешественник дает представления, где демонстрирует искусство черной магии. Каждое из таких представлений дается бесплатно и собирает полный зал. Поэтому, чтобы упростить процедуру регистрации на мероприятия, необходимо создать приложение, которое позволяет посетителям выполнить регистрацию самостоятельно.

- Предполагается, что для *регистрации* посетителю нужно указать:
- фамилию;
- дату рождения;
- номер телефона;
- представление, на которое зритель бы хотел попасть;

О каждом *представлении*, в свою очередь, должно быть известно следующее:

- название представления;
- максимальное количество мест;
- количество свободных мест (не задано в таблице);
- дата и время проведения;
- краткое описание;

Таблица 1. Информация о *представлениях*

1	Языки пламени	50	18.06.2022 19:00	Встречайте первый сеанс темной магии! Вы удивитесь тем чудесам, которые Маэстро выполняет на сцене! Первое представление - демонстрация стихии огня. Огонь из ничего, огонь из всего, огонь во все! Посетите данное представление, и вы узнаете, что огонь может приобретать самые причудливые формы.
2	Золотая лихорадка	35	15.09.2022 20:00	Второй сеанс темный магии обещает вам сказочные богатства! Только 15

				сентября, вы сможете с помощью темной магии Маэстро, превратить любые предметы в золото! Не пропустите! Также, представление обещает несколько сцен с исчезновением, и пару магических трюков с перемещением предметов силой мысли!
3	Добро пожаловать на сеанс!	50	30.10.2022 20:00	Встречайте третий сеанс темной магии! В этот раз, вы сможете попросить переместить в зал любого человека и с помощью темной магии этот человек попадет на сцену! Пусть он убедится в серьезности темной магии!

Руководство театра ожидает, что ваша разработка:

- позволит просматривать информацию о предстоящих мероприятиях в удобном виде. Если представление уже прошло, то его не должно быть видно;
- позволяет зарегистрировать гостя на интересующее представление (один человек может зарегистрироваться дважды, указав один и тот же номер телефона. Руководство театра надеется на порядочность посетителей и предполагает, что этого не произойдет). После регистрации гость получает секретный код (в качестве секретного кода может быть использован код регистрации, либо сгенерированный любым другим образом код);
- позволяет отменить участие по указанному коду (код выдается при регистрации). Перед отменой участия, необходимо для проверки показать информацию о посетителе;

При разработке стоит учесть следующее:

- невозможно зарегистрироваться на мероприятие, если нет свободных мест;
- гости не могут изменять информацию о мероприятиях и других гостях;
- будет нехорошо, если например, программа завершится с ошибкой при попытке регистрации пользователя с пустыми полями;
- темная магия - вещь серьезная. Лицам, не достигшим 18 лет в регистрации должно быть отказано.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
МДК. 05.03 ТЕСТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

для специальности
09.02.07 «Информационные системы и программирование»

Екатеринбург
2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данный фонд оценочных средств разработан для дисциплины «МДК. 05.03 Тестирование информационных систем», направленной на формирование профессиональных компетенций в области тестирования информационных систем.

Основная цель пособия — обеспечение качественного освоения студентами теоретических знаний и практических навыков в сфере тестирования программных продуктов, включая методы анализа, виды, методики, а также особенности организации процессов тестирования и фиксации результатов.

Структура пособия

Теоретическая часть

Посвящена изучению ключевых понятий и методов тестирования информационных систем, типам ошибок, критериям и принципам тестирования, видам (в том числе автоматизированным) и этапам тестирования, а также основам функционального и нефункционального тестирования. Цель — формирование основательного понимания процессов тестирования и подготовки к выполнению практических заданий.

Практическая часть

Направлена на закрепление теоретических знаний через выполнение лабораторных работ, практических заданий и проектов. Каждое практическое задание сопровождается детальным описанием плана выполнения, примерами и образцами решений, а также рекомендациями по самостоятельной проверке и оценке результата. Практика позволяет формировать навыки проведения тестирования разных видов, анализа результатов, документирования ошибок и работы с инструментами автоматизации тестирования.

Учебные цели и задачи

Основные учебные цели включают:

- освоение системного подхода к организации процессов тестирования;
- изучение видов тестирования и методик их применения на практике;
- развитие навыков выявления и анализа ошибок программных модулей;
- формирование умений документирования и отчётности по результатам тестирования;
- освоение инструментов и технологий автоматизации тестирования.

Задачи курса:

- изучить теоретические основы и методологию тестирования;
- овладеть навыками задания критериев, разработки и проведения тестов;
- осваивать проведение статического и динамического тестирования;
- совершенствовать умения работы с системами отслеживания ошибок и управления тестированием;
- формировать способности к анализу качества программных продуктов с применением автоматизированных средств.

Контроль знаний

Контроль знаний осуществляется поэтапно:

- текущие проверки в форме опросов и тестов по пройденной теории;
- проверка выполнения практических лабораторных заданий с анализом результатов;
- итоговая аттестация в виде экзамена, включающего теоретическую часть и практические задания по тестированию.

Оценочная деятельность

Оценка проводится по следующим критериям:

- полнота и качество выполнения лабораторных и домашних заданий;
- теоретическая подготовленность и способность применять знания на практике;
- активность в учебном процессе, участие в обсуждениях и дополнительных занятиях;

Комплект оценочных средств

Задание направлено на:

1. создание библиотеки классов
2. создание методов в соответствии с заданием (задания для создания методов предлагаются разной сложности . смотрите приложение)
3. создание тестовых наборов и реализацию модульного тестирования методов

КРИТЕРИИ ВЫСТАВЛЕНИЯ ОЦЕНОК

Результаты контроля определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Количество баллов	Оценка
10-14	Отлично
7-9	Хорошо
5-6	Удовлетворительно
3 и ниже	Неудовлетворительно

КРИТЕРИИ ОЦЕНИВАНИЯ

1. Метод реализован (0 - метод отсутствует; 1 - добавлена сигнатура метода; 2,3 - имеется частичное решение; 4 - задача решена полностью);
2. Создан корректный набор тестовых сценариев (0 - тесты отсутствуют; 1,2 - имеется набор тестов, однако есть определенные недочеты; 3 - набор тестов составлен правильно);
3. Создан документация к тест-кейсам (0 – документация отсутствует; 1,2 - имеется документация, однако есть определенные недочеты; 3 – документация выполнена по стандартам);
4. Метод имеет XML-комментарий (0 - комментарий отсутствует, либо не информативен; 1 - комментарий составлен как надо);
5. Общая оценка "читаемости" предоставленного кода: фор-матирование кода, имена переменных, использование подпрограмм и т. п. (0 – код не читаем , либо неинформативен; 1 – код «читаем»);
6. Общая оценка рациональности выбранных методов решений (0 – выбранные способы решения не рациональны; 1, 2 – выбранные способы решения рациональны)

Инструкция по выполнению заданий

- Время для выполнения задания – 4 часа.

ЗАДАНИЕ №1

1. Создать библиотеку классов, в которой реализован метод, выполняющий разворот строки.

Сигнатура метода

public static string ReverseString(string textString)

Входное значение: в качестве параметра передается строка, содержащая ноль или более символов.

Возвращаемое значение: Возвращается “перевернутая” строка, в которой последний сим-вол станет первым, предпоследний вторым и т. д.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №2

1. Создать библиотеку классов, в которой реализован метод OnlyLetters, который бы определял, что строка состоит только лишь из букв.

Сигнатура метода

public static bool OnlyLetters(string textString)

Входное значение: метод принимает в качестве параметра строку. Предполагается, что строка может быть пустой или содержать произвольный набор символов.

Возвращаемое значение: метод возвращает true, если входная строка содержит только буквы. Если имеется хотя бы один символ, которой не является буквой, то метод должен вернуть false. В случае пустой строки метод должен вернуть false.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №3

1. Создать библиотеку классов, в которой реализован метод OnlyDigits, который бы определял, что строка состоит только лишь из цифр.

Сигнатура метода

public static bool OnlyDigits(string textString)

Входное значение: метод принимает в качестве параметра строку. Предполагается, что строка может быть пустой или содержать произвольный набор символов.

Возвращаемое значение: метод возвращает true, если входная строка содержит только цифры. Если имеется хотя бы один символ, которой не является цифрой, то метод должен вернуть false. В случае пустой строки метод также должен вернуть false.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №4

1. Создать библиотеку классов, в которой реализован метод, выполняющий проверку на валидность переданного номера телефона. Строка является валидным номером телефона, если строка начинается с « +7», да-лее в правильном номере телефона следует 10 символов, являющихся цифрами.

Сигнатура метода

public static bool IsPhoneNumber(string textString)

Входное значение: в качестве параметра передается строка, которая может быть пустой или содержать любые символы.

Возвращаемое значение: метод возвращает true, если переданная строка соответствует условиям выше и является корректным номером телефона. В противном случае возвращается false.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №5

1. Создать библиотеку классов, в которой реализован метод Censore, который бы заменял во входной строке все неприличные слова на звез-дочки ***. Список неприличных слов задается. Регистр неприличных слов не важен

Сигнатура метода

public static string Censore(string textString, List<string> cens)

Входное значение

В качестве параметров передаются:

- проверяемая строка textString, которая может быть пустой или содержать некоторый текст;
- Список cens – это список неприличных слов. Регистр не-приличных слов не важен

Возвращаемое значение: возвращается отфильтрованный текст, в котором все неприличные слова заменены на звездочки ***

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №6

1. Создать библиотеку классов, в которой реализован метод, который бы определял количество слов word в тексте textString. Предполагается, что словом является любая последовательность букв, ограниченная двумя или одним пробелом (в случае начала или конца строки). Предполагается, что слова проверяются без учета регистра символов.

Сигнатура метода

public static int WordCount(string textString, string word)

Входные значения: метод принимает две строки: textString и word. Строка textString содержит исходный текст, строка word представляет собой слово, которое нужно искать. Для упрощения задачи гарантируется, что переменная textString содержит только буквы и пробелы; Значение переменной word всегда состоит из букв и содержит по крайней мере два символа.

Возвращаемое значение: метод возвращает целое число - количество слов word в тексте textString.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №7

1. Создать библиотеку классов, в которой реализован метод IsPalindrome, который бы проверял, является ли переданный текст textString палиндромом.

Палиндромом будем считать такой текст, который читается одинаково “слева-направо” и “справа-налево”. При этом, в самом тексте не должны учитываться пробельные символы. Например, палиндромом будет являться текст: “А роза упала на лапу Азора”.

Сигнатура метода

public static bool IsPalindrome(string textString)

Входное значение: в качестве параметра передается строка, которая может быть пустой или содержать некоторый текст. Текст может включать в себя буквы, цифры и пробелы.

Выходное значение: если переданный текст является палиндромом, то возвращается true. В случае пустой строки следует также вернуть true. В остальных случаях возвращается false.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №8

1. Создать библиотеку классов, в которой реализован метод GetLetters, который бы возвращал список всех букв, использованных в пере-данном тексте textString. Особенности списка, который формируется на выходе метода:

- список не должен содержать повторяющихся элементов;
- буквы в списке должны быть приведены к верхнему регистру;
- список должен быть отсортирован в алфавитном порядке (по возрастанию).

Допускается использовать для сортировки библиотечные функции.

Сигнатура метода

public static List<char> GetLetters(string textString)

Входное значение: в качестве параметра передается строка, которая может быть пустой или содержать некоторый текст. Текст может включать в себя произвольные символы.

Возвращаемое значение: возвращается список символов List<char>. Список должен со-держать буквы искомого текста и удовлетворять требованиям, указанным выше.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №9

1. Создать библиотеку классов, в которой реализован метод IsPangram, которых бы определял, является ли указанный текст textString панграммой.

Панграмма - короткий текст, использующий все или почти все буквы алфавита, по возможности не повторяя их (Wikipedia).

Будем считать панграммой строку, которая содержит все буквы русского алфавита.

Сигнатура метода

public static bool IsPangram(string textString)

Входное значение: в качестве параметра передается строка, которая может быть пустой или содержать некоторый текст. Текст может включать в се-бя произвольные символы.

Возвращаемое значение: возвращает true, если переданный текст является панграммой. В противном случае возвращается false.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №10

1. Создать библиотеку классов, в которой реализован метод для определения самого длинного слова в тексте. Предполагается, что словом является любая последовательность букв, ограниченная двумя или одним небуквенным символом (один символ в случае начала или конца строки). Если текст содержит несколько слов максимальной длины, то следует вернуть то слово, которое встречается в тексте первым.

Сигнатура метода

public static string FindLongestWord(string text)

Входное значение

В качестве параметра передается строка, которая может быть пустой или содержать некоторый текст. Гарантируется, что текст содержит только буквы, пробелы и знаки пунктуации ,, ., ! и ?. Обратите внимание, что символы пунктуации не являются частью слова.

Возвращаемое значение: возвращается строка, содержащее самое длинное слово в тексте, приведенное к нижнему регистру. В случае отсутствия слов, следует вернуть пустую строку.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №11

1. Создать библиотеку классов, в которой реализован метод, который бы возвращал список всех слов, которые встречаются в указанном тексте более одного раза.

Сигнатура метода

public static List<string> FindNonUniqueWords(string text)

Входное значение: в качестве параметра передается строка, которая может быть пустой или содержать некоторый текст. Гарантируется, что текст содержит только буквы, пробелы, переносы \n и знаки пунктуации ,, ., ! и ?.

Возвращаемое значение: метод возвращает список List<string> всех слов, которые встречаются в тексте два или более раз. Слова следует привести к нижнему регистру, а список следует отсортировать в алфавитном порядке.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №12

1. Создать библиотеку классов, в которой реализован метод FindMinimum, который бы возвращал число, соответствующее минимальному числу в заданном массиве целых чисел

Сигнатура метода

public static int FindMinimum(List<int> arrayNumbers)

Входное значение: в качестве параметра передается массив целых чисел

Возвращаемое значение: метод возвращает целое число, которое является минимальным числом в заданном массиве.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №13

1. Создать библиотеку классов, в которой реализован метод для работы со строкой. Пусть имеется строка вида "N/M. Реализуйте метод, который бы в случае, когда N > M заменил N на минимальное значение (среди чисел M и N). Например,

"1/10" => "1/10"

"251/100" => "100/100"

Сигнатура метода

public string FixWrongResult(string textString)

Входное значение: в качестве параметра передается строка. Гарантируется (не нужно это проверять, значение “заложено” условием задачи), что строка имеет вид "ЧИСЛО1/ЧИСЛО2". Гарантируется, что оба числовых значения корректны, положительны и умещаются в стандартный тип int.

Возвращаемое значение: метод возвращает строку, представляющую “исправленный результат”.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №14

1. Создать библиотеку классов, в которой реализован метод PasswordStrengthCheker, который позволяет определить степень сложности пароля.

- если пароль имеет длину больше 7 символов сложность повышается на 1 балл;
- если пароль содержит цифры, то сложность повышается ещё на 1 балл;
- если пароль содержит латинские буквы в нижнем регистре, то сложность повышается ещё на 1 балл;
- если пароль содержит Заглавные латинские буквы, то сложность повышается ещё на 1 балл;
- если пароль содержит спецсимволы, то сложность повышается ещё на 1 балл
- если пароль содержит кириллические символы, то выдается исключение «Кириллические символы запрещены при вводе пароля»

Сигнатура метода

public int PasswordStrengthCheker(string textString)

Входное значение: в качестве параметра передается строка - пароль

Возвращаемое значение: метод возвращает целое число, соответствующее сложности па-роля

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №15

1. Создать библиотеку классов, в которой реализован метод, который по контрольной цифре определяет корректность введенного номера.

Задана строка – номер из 12 цифр определить корректность но-мера по «контрольной» последней цифре.

Для проверки контрольной цифры необходимо умножить каждую цифру числа на соответствующий номер позиции цифры в этой строке. После этого получившееся число поделить на 11. Ес-ли полученный остаток равен 10, тогда последняя цифра числа должна быть равна 1, в противном случае она равна полученному остатку.

Например: для числа 279146358279 необходимо посчитать сум-му 1 * 2 + 2 * 7 + 3 * 9 + 4 * 1 +… .. + 12 * 9 . Поделить получен-ное число на 11. Если полученный остаток равен 10, тогда по-следняя цифра числа должна быть равна 1, в противном случае она равна полученному остатку.

Сигнатура метода

public bool CheckNumber (string textString)

Входное значение: в качестве параметра передается строка

Возвращаемое значение: метод возвращает true, если контрольная цифра задана верно, false – в случае неверной контрольной цифры

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №16

1. Создать библиотеку классов, в которой реализован метод, который по контрольной цифре определяет корректность введенного номера.

Число задается в виде строки из цифр. Необходимо получить контрольную цифру для этого числа. Подсчитываются суммы цифр, стоящих на четных и нечетных местах. Сумма цифр, стоящих на четных местах (, суммируется с утроенной суммой цифр, стоящих на нечетных местах.

Если цифра единиц полученного результата равна нулю, то контрольная цифра — 0. Если последняя цифра результата не нуль, то контрольная цифра равна дополнению этой цифры до 10.

Например:

Задана строка из цифр 345678905675

Цифры, стоящие на нечетных местах (справа-налево): 4, 6, 8, 0, 6, 5. Сумма этих цифр равна 29.

Цифры, стоящие на четных местах (справа-налево): 3, 5, 7, 9, 5, 7. Сумма этих цифр равна 36.

29*3+36=123. Искомая контрольная цифра 7, т.к. 7+3 = 10.

Сигнатура метода

public int CheckDigit(string textString)

Входное значение: в качестве параметра передается строка из цифр

Возвращаемое значение: метод возвращает целое число, которое является контрольной цифрой для заданной строки, высчитанной по описанному алгоритму

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №17

1. Создать библиотеку классов, в которой реализован метод, выполняющий проверку корректности СНИЛС

Страх-вой номер индивидуального лицевого счета страхового свидетельства обязательного пенсионного страхования(он же СНИЛС) проверяется на валидность контрольным числом.

СНИЛС имеет вид: «XXX-XXX-XXX YY», где XXX-XXX-XXX — собственно номер, а YY — контрольное число. Алгоритм формирования контрольного числа СНИЛС таков:

- Проверка контрольного числа Страхового номера проводится толь-ко для номеров больше номера 001-001-998
- Контрольное число СНИЛС рассчитывается следующим образом:

1) Каждая цифра СНИЛС умножается на номер своей позиции (позиции отсчитываются с конца)

2) Полученные произведения суммируются

3) Если сумма меньше 100, то контрольное число равно самой сумме

4) Если сумма равна 100 или 101, то контрольное число равно 00

5) Если сумма больше 101, то сумма делится по остатку на 101 и контрольное число определяется остатком от деления аналогично пунктам 2.3 и 2.4

Например: Указан СНИЛС 112-233-445 95

Проверяем правильность контрольного числа:

цифры номера 1 1 2 2 3 3 4 4 5

номер позиции 9 8 7 6 5 4 3 2 1

Сумма = 1×9 + 1×8 + 2×7 + 2×6 + 3×5 + 3×4 + 4×3 + 4×2 + 5×1 = 95

95 ÷ 101 = 0, остаток 95. Контрольное число 95 — указано верно

Сигнатура метода

public bool CheckMedicalCode(string textString)

Входное значение: в качестве параметра передается строка из цифр

Возвращаемое значение: метод возвращает true, если СНИЛС задан верно, false – в случае неверного СНИЛС

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №18

1. Создать библиотеку классов, в которой реализован метод, выполняю-щий расчет контрольной цифры IMEI

IMEI присваивается телефону во время изготовления на заводе. Он служит для идентификации устройства в сети и хранится в прошивке аппарата. Как правило, 15-я цифра IMEI является так называемым контрольным числом, которое позволяет проверить правильность всех остальных цифр.

Вычисление контрольной цифры IMEI

Например IMEI=354190023896443.

1. Сложить все цифры в нечетных позициях;
2. Заменить цифры на четных местах на :

0=0
1=2
2=4
3=6
4=8
5=1
6=3
7=5
8=7
9=9

3. К полученному числу прибавить результат, полученный в п.1;
4. Если полученное число кратно 10, тогда контрольное число IMEI равно 0. В противном случае контрольная сумма равна числу, кото-рое нужно прибавить к результату, чтобы получить ближайшее большее «круглое» число.

Пример

IMEI=35-419002-389644-3.

Сложить все цифры в нечетных позициях:

3+4+9+0+3+9+4 = 32

Заменить цифры на четных местах:

5,1,0,2,8,6,4 => 1,2,0,4,7,3,8

и сложить их:

1+2+0+4+7+3+8 = 25

К полученному числу прибавить результат п.1.

25 + 32 = 57

Контрольная сумма равна числу, которое нужно прибавить к результату, чтобы получить ближайшее большее «круглое» число: 60 — 57 = 3. Итого контрольное число равно 3.

Сигнатура метода

public int CheckDigitIMEI(string textString)

Входное значение

В качестве параметра передается строка из цифр

Возвращаемое значение: метод возвращает true, если контрольное цифра задана верно, false – в случае неверной контрольной цифры

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №19

1. Создать библиотеку классов, в которой реализован метод, который позволяет определить корректно ли введен код EAN-13 товара

Для этого необходимо посчитать контрольную сумму , которая вычисляется по следующему алгоритму:

1) каждая из цифр номера умножается на соответствующие коэф-фициенты из таблицы.

k13	k12	k11	k10	k9	k8	k7	k6	k5	k4	k3	k2	k1
1	3	1	3	1	3	1	3	1	3	1	3	1

4600051000057 (сигареты «Прима») — код EAN-13.

4×1+ 6×3+ 0×1+ 0×3+ 0×1+ 5×3+ 1×1+ 0×3+ 0×1+ 0×3+ 0×1+ 5×3+ 7×1=

4+ 18+ 0+ 0+ 0+ 15+ 1+ 0+ 0+ 0+ 0+ 15+ 7= 60.

2) Если полученная сума дает остаток от деления на 10 равный нулю, то номер признаётся правильным.

3) 60%10=0 — номер правильный.

Сигнатура метода

public bool CheckNumber(string textString)

Входное значение: в качестве параметра передается строка из цифр (код)

Возвращаемое значение

Метод возвращает true, если код задан верно, false – в случае неверной контрольной цифры

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

ПМ.05 Проектирование и разработка информационных систем»
для специальности 09.02.07
«Информационные системы и программирование»

Екатеринбург

2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические рекомендации по выполнению курсовой работы являются частью учебно-методического комплекса (УМК) по ПМ.05. «Проектирование и разработка информационных систем» для специальности 09.02.07 «Информационные системы и программирование».

Курсовой проект по ПМ.05. «Проектирование и разработка информационных систем» включает в себя разделы : проектирование информационной системы, разработку кода информационной системы, тестирование разработанной информационной системы.

Методические рекомендации определяют цели, задачи, порядок выполнения, а также содержат требования к лингвистическому и техническому оформлению курсовой работы, практические советы по подготовке и прохождению процедуры защиты.

1. ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА

1.1 Цель курсового проектирования

Результаты обучения:
уметь:
- проводить анализ предметной области;
- осуществлять поиск, обобщать, анализировать необходимую информацию;
- оформлять программную и техническую документацию, с использованием стандартов оформления программной документации;
- разрабатывать мероприятия для решения поставленных в курсовой работе задач;
- осуществлять выбор модели построения информационной системы и программных средств;
- решать прикладные вопросы программирования и языка сценариев для создания программ;
- проектировать и разрабатывать систему по заданным требованиям и спецификациям;
- разрабатывать графический интерфейс приложения;
- создавать проект по разработке приложения и формулировать его задачи;
- кодировать на языках программирования;
- использовать последнюю версию программного обеспечения среды разработки и инструменты, чтобы изменить существующие коды и писать новый код "клиент-сервер";
- определять и интегрировать соответствующие библиотеки и фреймворки в программное решение;
- разрабатывать тест-кейсы, применять методики тестирования;
- проверять результаты тест-кейсов;
- устранять и исправлять ошибки;
- документировать результаты тестирования (составлять отчет о процессе тестирования);
- фиксировать результаты тестирования;

знать:
- инструменты и методы модульного тестирования;
- современные методики тестирования;
- принципы устранения распространенных проблем программных приложений;
- важность тщательного тестирования решения;
- важность документирования испытаний;
- Национальную и международную систему стандартизации и сертификации и систему обеспечения качества продукции, методы контроля качества;
- Объектно-ориентированное программирование;
- Спецификации языка программирования, принципы создания графического пользовательского интерфейса (GUI);

- Важность рассмотрения всех возможных вариантов и получения наилучшего решения на основе анализа и интересов клиента;
- Файлового ввода-вывода, создания сетевого сервера и сетевого клиента;
- Платформы для создания, исполнения и управления информационной системой;
- Современные объектно-ориентированные языки программирования;
- Программные средства и платформы инфраструктуры информационных технологий организаций;
- Важность использования системного анализа и методологий проектирования (например, унифицированного языка моделирования (UnifiedModellingLanguage), программной платформы MVC (Model-View-Control), фреймворки, шаблоны проектирования);
- Новые технологии и принимать решение о целесообразности их применения;
- Важность оптимизации архитектуры системы с учетом модульности и повторного использования;
- Важность использования методологий разработки системы (например, объектно-ориентированные технологии);
- Важность рассмотрения всех нормальных и ненормальных сценариев и обработки исключений;
- Важность соблюдения стандартов (например, правила кодирования, руководство по стилю (оформление системной и программной документации), дизайна пользовательского интерфейса, управление каталогами и файлами);
- Важность точного и постоянного контроля версий (управление версиями);
- использование существующего кода в качестве основы для анализа и модификации;
- важность выбора наиболее подходящих средств разработки из предложенных вариантов.

2. ОБЩИЕ ПОЛОЖЕНИЯ

2.1. Требования к содержанию курсовой работы

Курсовой проект– заключительный этап изучения профессионального модуля ПМ.05. «Проектирование и разработка информационных систем» для специальности 09.02.07 «Информационные системы и программирование».

Цель курсового проекта – систематизация и закрепление теоретических знаний, полученных за время обучения, а также формирование навыков самостоятельного практического применения современных методов и средств тестирования и создания автоматизированной информационной системы, построенной на основе базы данных.

Курсовой проект основывается на обобщении практических/лабораторных работ, выполненных студентом, по дисциплинам «МДК. 05.01 Проектирование и дизайн информационных систем», «МДК 05.02 Разработка кода информационных систем», «МДК 05.03 Тестирование информационных систем» для специальности 09.02.07 «Информационные системы и программирование».

Разработка кода приложения должна реализовываться через ТЕСТИРОВАНИЕ с использованием методики Test-Driven Development (TDD).

2.2. Структура пояснительной записки

Титульный лист
Ведомость проекта
Содержание
Введение
1. Аналитическая часть
   1.1.Цели, функции и задачи системы
   1.2.Описание предметной области системы
   1.3.Определение пользователей системы

1.4.Проектные решения по реализации тестирования приложения
1.5.Проектные решения по реализации интерфейса приложения
2. Проектная часть
2.1.Модель информационной системы
2.2. Файловая структура приложения
2.3. Реализация кода приложения
2.4. Разработка библиотек
2.5. Работа с системой контроля версий
2.6. Документация по работе с информационной системой
3. Глоссарий
4. Список источников

Заключение

Приложения

2.3. ТИТУЛЬНЫЙ ЛИСТ

– Титульный лист должен включать: Название образовательного учреждения; Тему курсового проекта, группу, ФИО студента, ФИО руководителя проектирования, город и год.
– Бланк титульного листа приводится в Приложении 1.

2.4. ВЕДОМОСТЬ ПРОЕКТА

– Бланк ведомости проекта приводится в Приложении 1.

2.5. СОДЕРЖАНИЕ

– Содержание должно генерироваться автоматически

2.6. ВВЕДЕНИЕ

Во введении следует обосновать актуальность избранной темы курсовой работы, раскрыть ее теоретическую и практическую значимость, сформулировать цели и задачи работы.

Введение должно подготовить читателя к восприятию основного текста работы. Оно состоит из обязательных элементов, которые необходимо правильно сформулировать. **В первом предложении называется тема курсовой работы.**

Актуальность исследования (почему это следует изучать?) Актуальность исследования рассматривается с позиций социальной и практической значимости. В данном пункте необходимо раскрыть суть исследуемой проблемы и показать степень ее проработанности в различных трудах по разработке информационных систем. Здесь же можно перечислить источники информации, используемые для исследования.

Цель исследования (какой результат будет получен?) Цель должна заключаться в решении исследуемой проблемы путем ее анализа и практической реализации.

Объект исследования (что будет исследоваться?). Объект предполагает работу с понятиями. В данном пункте дается определение экономическому явлению, на которое направлена исследовательская деятельность. Опишите место информационной системы в жизни общества, где может использоваться система и почему, какие задачи позволит решить);

Предмет исследования (как, через что будет идти поиск?) Здесь необходимо дать определение планируемым к исследованию конкретным свойствам объекта или способам изучения экономического явления. Предмет исследования направлен на практическую деятельность и отражается через результаты этих действий.

Задачи курсового проектирования (как идти к результату?), пути достижения цели. Задачи соотносятся с гипотезой. Определяются они исходя из целей работы.

Формулировки задач необходимо делать как можно более тщательно, поскольку описание их решения должно составить содержание глав и параграфов работы. Как правило, формулируются 3 - 4 задачи.

Пример описания задач:

1. «на основе теоретического анализа литературы и Internet-источников произвести анализ предметной области...».
2. «Определить... » (выделить основные условия, факторы, причины, влияющие на объект исследования).
3. «Раскрыть... » (выделить основные условия, факторы, причины, влияющие на предмет исследования).
4. «Разработать... » (средства, условия, формы, программы).
5. «Протестировать…» (что разработали) и дать рекомендации...

Структура курсового проекта – это завершающая часть введения (что в итоге в работе представлено).

В завершающей части в назывном порядке перечисляются структурные части работы, например: «Структура работы соответствует логике исследования и включает в себя введение, практическую часть, заключение, список литературы, 5 приложений».

Здесь допустимо дать развернутую структуру курсовой работы и кратко изложить содержание глав. (Чаще содержание глав курсовой работы излагается в заключении).

Таким образом, введение должно подготовить к восприятию основного текста работы.

Краткие комментарии по формулированию элементов введения представлены в таблице 1.

Таблица 1.Комментарии по формулированию элементов введения

Элемент введения	**Комментарий к формулировке**
Актуальность темы	*Почему это следует изучать?* Раскрыть суть исследуемой проблемы и показать степень ее проработанности.
Цель исследования	*Какой результат будет получен?* Должна заключаться в решении исследуемой проблемы путем ее анализа и практической реализации.
Объект исследования	*Что будет исследоваться?* Дать определение явлению или проблеме, на которое направлена исследовательская деятельность.
Предмет исследования	*Как и через что будет идти поиск?* Дать определение планируемым к исследованию конкретным свойствам объекта или способам изучения явления или проблемы.
Задачи курсового проектирования	*Как идти к результату?* Определяются исходя из целей работы и в развитие поставленных целей. Формулировки задач необходимо делать как можно более тщательно, поскольку описание их решения должно составить содержание глав и параграфов работы. Рекомендуется сформулировать 3 – 4 задачи.
Структура курсового проекта (завершающая часть введения)	*Что в итоге в работе представлено.* Краткое изложение перечня и/или содержания глав работы.

2.7. АНАЛИТИЧЕСКАЯ ЧАСТЬ

Разделы, которые должна включать аналитическая часть пояснительной записки:

2.7.1. Цели, функции и задачи системы

Необходимо ПОДРОБНО описать назначение и функционал информационной системы, из каких модулей (интерфейсных окон) состоит информационная система.

2.7.2. Описание предметной области системы

Необходимо подробное описание области, для которой разрабатывается проект. Какая информация взята в основу проекта (какие исходные данные) и почему (какие опросы были произведены, что было изучено для того, чтобы определить исходные данные для проекта). Здесь же могут описаны формулы для расчета (если таковые есть).

2.7.3. Определение пользователей системы

Необходимо перечислить для кого предназначена ИС и почему, как планируется разграничивать доступ пользователей к ИС, какими средствами будете пользоваться для реализации разграничения доступа. ЗДЕСЬ НУЖНО добавить диаграмму вариантов использования.

Диаграмма вариантов использования (диаграмма прецендентов) направлена на описание функций, доступных разным группам пользователей, описание принципов разграничения доступа к данным пользователями);

Требования к диаграмме вариантов использования:

- в диаграмме должны быть описаны все участники системы (актеры).
- должны быть описаны возможные варианты сценариев работы с системой с точки зрения пользователя, возможные пути использования системы
- актеры должны быть оформлены в виде "проволочных человечков"
- прецеденты (варианты использования) должны оформляться с помощью отглагольных существительных или глаголами в неопределенной форме
- связь между актером и прецендентом изображается в виде "Сплошной" линии
- связь между прецендентами изображается в виде "Пунктирной" линии (с обязательным указанием типа зависимости: <<include>> или <<extend>>

2.7.4. Проектные решения по реализации тестирования приложения

Требования к разделу:

- Необходимо перечислить все функции приложения, которые планируется протестировать (в соответствии с таблицей 2)

Таблица 2. Функционал приложения

Requirements ID	**Описание функции**
Req_1	Добавление данных в таблицу «Имя таблиц»
Req_1	
…	

- Необходимо сформировать план тестирования (Test Plan) – проектную документацию описывающую и регламентирующую процесс тестирования. Должен быть сформирован документ, описывающий весь объем работ по тестированию, начиная с описания объекта до стратегии. Test Plan должен быть оформлен в соответствии с таблицей 3.

Таблица 3. Test Plan

test case ID	**описание объекта тестирования (requirements)**	**стратегия тестирования (вид тестирования)**	**Последовательность проведения тестирования в разрезе запланированных фаз разработки**	**Критерии начала тестирования**	**Критерии окончания тестирования**

TC_1	неправильный формат ввода данных в поле «Логин» при авторизации				
TC_2	нарушение области допустимых значений при вводе показателей				
	Создание жильца без ФИО				

– Необходимо сформировать Traceability matrix —двумерную таблиц, содержащую соответсвие функциональных требований продукта (functional requirements) и подготовленных тестовых сценариев (test cases). Traceability matrix должна быть оформлена в соответствии с таблицей 4

Таблица 4. Traceability matrix

	test case ID	TC_1	TC_2	TC_3	TC_4	TC_5	TC_6	TC_7	TC_8
Requirements ID									
Req_1		+							
Req_2			+		+				
Req_3				+			+		
Req_4			+		+				
Req_5						+			
Req_6							+		
Req_7					+			+	
Req_8									+

2.7.5. Проектные решения по реализации интерфейса приложения (разработка руководства по стилю)

Разработка руководства по стилю

Необходимо разработать руководство по стилю. Рабочие ссылки на руководство по стилю и само руководство по стилю должны быть представлены в пояснительной записке к информационной системе, а также продемонстрированы в презентации при публичной защите курсового проекта.

Требования к руководству по стилю:

- Необходимо описать требования к шрифтам, которые используются в проекте;
- Все экранные формы пользовательского интерфейса должны иметь заголовок и логотип. Поэтому в руководстве по стилю должны быть описаны требования к логотипу проекта (логотип должен быть добавлен в руководство по стилю в виде изображения);
- Описаны требования к иконке проекта;
- Описаны требования к цветовому оформлению (требования к заднему фону, кнопкам и др. элементам оформления);

- Описаны требования к размещению элементов и их размерам; (Например, значение высоты для маленьких, обычных и больших графических элементов (кнопок,текстовых полей, и т. п.) , внутренний отступ графических элементов (padding) , внешний отступ относительно других элементов (margin) , размер скругления границ (corner radius))

Прототип приложения

Необходимо представить прототип приложения.

Необходимо описать принципы, которыми руководствовались при создании интерфейса (Описать какие элементы интерфейса более заметны и почему? Обоснование размещения элементов в модальные окнах, на втором плане. По каким принципам группировали элементы?).

Проектные решения по реализации интерфейса приложения должны быть выполнены в Figma. Рабочие ссылки на прототип приложения должны быть представлены в пояснительной записке к информационной системе, а также продемонстрированы в презентации при публичной защите курсового проекта.

2.8. ПРОЕКТНАЯ ЧАСТЬ

2.8.1. Модель информационной системы

2.8.2. Схема базы данных

Необходимо создать физическую модель базы данных (ERD-диаграмму), реализованную средствами СУБД

Если ERD-диаграмма очень большая (10 и более таблиц), то рекомендуется вынести ее в приложение, сделав ссылку на нее в проектной части пояснительной записки.

2.8.3. Словарь данных

Необходимо создать словарь данных. Словарь данных создается для однозначного понимания того, что изображено в ERD-диаграмме. Необходимо сформировать словарь данных, который обеспечит понимание структуры и назначения каждого поля, описанного в сущностях спроектированной базы данных. Шаблон оформления словаря данных для одной сущности представлен в таблице 5.

Таблица 5 - Шаблон для оформления словаря данных

Ключ	**Имя поля**	**Тип данных**	**Нулевые значения**	**Дополнительные**	**Описание**
PK	question_id	INTEGER	N	Автозаполнение с шагом 1	ID Вопроса
	question	NVARCHAR(50)	N		Вопрос
FK	prof_id	INTEGER	N		ID Профессии

2.8.4. Файловая структура приложения

Требования к описанию файловой структуры:

- Исходя из функционала, описанного в ТЗ проекта, необходимо описать файловую структуру приложения и добавить скрин структуры папок приложения.
- Необходимо описать назначение каждой из папок приложения: папки Model, папки View, папки Controllers, и других.
- Необходимо создать диаграммы классов, чтобы наглядно описать созданные контроллеры. Диаграммы классов должны использоваться в пояснительной записке для описания принципа работы приложения.
- Необходимо описать методы, реализованные в созданных контроллерах (Например, методы добавления данных в таблицу, методы по фильтрации данных массива).

Каждый из созданных методов должен быть протестирован (для этих целей создаются дополнительные библиотеки).

– Необходимо представить XAML-документацию к созданным методам.

2.8.5. Реализация кода приложения

Требования к описанию разделу:

– Необходимо создать «Диаграммы последовательности» для описания процессов взаимодействия модулей программного продукта;

– Необходимо описать алгоритмы, используемые при разработке кода приложения.

2.8.6. Разработка библиотек

Требования к библиотекам:

- Каждый из созданных методов должен быть протестирован.
- Для каждого из созданных методов необходимо разработать позитивные и негативные тестовые случаи (test case); Объем тестовых данных должен охватывать все возможные ситуации при выполнении заданных операций;
- Для каждого теста необходимо разработать тестовую документацию. Документация должна быть оформлена в соответствии с шаблоном, представленным в таблице 6.

Таблица 6. Шаблон для оформления тестовой документации

Тестовый пример #	
Приоритет тестирования	
Заголовок/название теста	
Краткое изложение теста	
Этапы теста	
Тестовые данные	
Ожидаемый результат	
Фактический результат	
Статус	
Предварительное условие	
Постусловие	
Примечания/комментарии	

2.8.7. Работа с системой контроля версий

Требования к разделу:

- Необходимо проявить навыки работы с системой контроля версий GitHub, GitLub, Gogs или другие;
- Необходимо подробно описать процесс работы с системой контроля версий;

- Необходимо наглядного показать процесс контроля над версионностью программного продукта при работе с системой контроля версий с помощью скриншотов/команд;
- Необходимо представить ссылку на проект, размещенный в системе контроля версий.

2.8.8. Документация по работе с информационной системой

Руководство пользователя

В руководстве представляются приводятся подробные инструкции по действиям пользователей при выполнении различных операций по использованию системы для решения различных прикладных задач.

Требования к руководству пользователя:

- Необходимо, чтобы руководство пользователя описывало последовательность действий для выполнения всех функций вашей системы.
- Необходимо использовать живые примеры и скриншоты вашей системы для более наглядного пояснения шагов работы с различным функционалом.
- Необходимо разделить руководство на подразделы, использовать ссылки на рисунки, нумерованные и маркированные списки для описания шагов и т.д.

Руководство администратора

Требования к руководству администратора:

- Для выполнения операций редактирования, добавления, удаления данных информационной системы, в системе должны быть предусмотрены специальные категории пользователей;
- Необходимо предусмотреть личный кабинет администратора, вход в который осуществляется после авторизации;
- Обязательные элементы личного кабинета администратора: страница авторизации (поля Логин, Пароль, кнопка «Войти»);
- Необходимо, чтобы руководство администратора описывало последовательность действий для выполнения всех функций вашей системы.
- Необходимо использовать живые примеры и скриншоты вашей системы для более наглядного пояснения шагов работы с различным функционалом.
- Необходимо разделить руководство администратора на подразделы, использовать ссылки на рисунки, нумерованные и маркированные списки для описания шагов и т.д.

Руководство по установке программы

Требования к руководству по установке программы:

- Необходимо создать инсталлятор приложения;
- Необходимо выполнить подробное описание процесса развертывания приложения для реализации его полной работоспособности;
- Необходимо описать процесс подключения к БД ;
- Необходимо наглядного пояснить процесс установки приложения с помощью скриншотов;
- Необходимо разделить руководство по установке программы на подразделы, использовать ссылки на рисунки, нумерованные и маркированные списки для описания шагов и т.д.

2.8.9. ГЛОССАРИЙ

В пояснительную записку должны быть включены:

- словарь терминов,
- список используемых сокращений, которые были использованы при написании текста пояснительной записки.

Например,

1. БД – база данных
2. АРМ - автоматизированного рабочего места
3. СУБД – система управления базами данных

2.8.10. СПИСОК ИСТОЧНИКОВ

В пояснительной записке необходимо опираться на источники, которыми Вы руководствовались при ее составлении. В них обязательно должны входить ГОСТы, стандарты используемых методологий.

В список источников и литературы включаются источники, изученные Вами в процессе подготовки работы, в т. ч. те, на которые Вы ссылаетесь в тексте курсовой работы.

Список используемой литературы должен содержать 20 – 25 источников (не менее 10 книг и 10-15 материалов периодической печати), с которыми работал автор курсовой работы.

Список используемой литературы включает в себя:

- нормативные правовые акты;
- научную литературу и материалы периодической печати;
- практические материалы.

Источники размещаются в алфавитном порядке. Для всей литературы применяется сквозная нумерация.

При ссылке на литературу в тексте курсовой работы следует записывать не название книги (статьи), а присвоенный ей в указателе «Список источников и литературы» порядковый номер в квадратных скобках. Ссылки на литературу нумеруются по ходу появления их в тексте записки. Применяется сквозная нумерация.

Пример оформления списка источников:

1. PlantUML в двух словах [Электронный ресурс] - Режим доступа: https://plantuml.com/ru/
2. SQL Server connection strings [Электронный ресурс] – Режим доступа: https://www.connectionstrings.com/sql-server/
3. Н. А. Тюкачев, В. Г. Хлебостроев C#. Основы программирования. Учебное пособие (+ CD) - СПб.: ЭБС ЛАНЬ, 2017. – 272 с.
4. Создание запросов LINQ на языке C# [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/csharp/linq/write-linq-queries

2.8.11. ЗАКЛЮЧЕНИЕ

В заключении необходимо подвести итоги по теме. Заключение носит форму синтеза полученных в работе результатов. Его основное назначение - резюмировать содержание работы, подвести итоги проведенного исследования. В заключении излагаются полученные выводы и их соотношение с целью исследования, конкретными задачами, гипотезой, сформулированными во введении.

2.8.12. ПРИЛОЖЕНИЯ

Приложения могут отсутствовать в пояснительной записке. Целесообразно вынести в приложения материал, который «не вписывается» в общий контекст описания проектной работы

В большинстве случаев, в Приложение «выносят» справочные материалы по проекту или материалы, несущие описательный характер и показывающие объем проделанной работы.

3. ОБЩИЕ ПРАВИЛА ОФОРМЛЕНИЯ КУРСОВОГО ПРОЕКТА

3.1. Оформление текстового материала

Текстовая часть работы должна быть представлена в компьютерном варианте на бумаге формата А4. Шрифт – Times New Roman, размер шрифта – 14, одинарный интервал, выравнивание по ширине. Страницы должны иметь поля (рекомендуемые): нижнее – 2,5; верхнее – 2; левое – 3; правое – 1,5.

Объем курсовой работы – 40-60 страниц. Все страницы работы должны быть подсчитаны, начиная с титульного листа и заканчивая последним приложением. Нумерация страниц должна быть сквозная, начиная с введения и заканчивая последним приложением. Номер страницы ставится в правой части листа нижнего поля.

Весь текст работы должен быть разбит на составные части. Разбивка текста производится делением его на разделы и подразделы (смотри п.1.2. «Структура пояснительной записки».

В содержании работы не должно быть совпадения формулировок названия одной из составных частей с названием самой работы, а также совпадения названий глав и параграфов. Названия разделов и подразделов должны отражать их основное содержание и раскрывать тему работы.

При делении работы на разделы (согласно ГОСТ 2.105-95) их обозначают порядковыми номерами – арабскими цифрами без точки и записывают с абзацного отступа. При необходимости подразделы могут делиться на пункты. ***Номер пункта*** должен состоять из номеров раздела, подраздела и пункта, разделённых точками. В конце номера раздела, пункта точку не ставят.

Если раздел или подраздел состоит из одного пункта, он также нумеруется. Пункты при необходимости, могут быть разбиты на подпункты, которые должны иметь порядковую нумерацию в пределах каждого пункта, например*: 4.2.1.1, 4.2.1.2, 4.2.1.3* и т. д.Каждый пункт, подпункт и перечисление записывают с абзацного отступа. Разделы, подразделы должны иметь заголовки.

Наименование разделов должно быть кратким и записываться в виде заголовков (в красную строку) ЗАГЛАВНЫМИ БУКВАМИ, без подчеркивания и без точки в конце. Заголовки должны четко и кратко отражать содержание разделов, подразделов, пунктов.

Нумерация страниц основного текста и приложений, входящих в состав работы, должна быть сквозная.

В основной части работы должны присутствовать таблицы, схемы, графики с соответствующими ссылками и комментариями.

В работе должны применяться научные и специальные термины, обозначения и определения, установленные соответствующими стандартами, а при их отсутствии – общепринятые в специальной и научной литературе.

3.2. Оформление иллюстраций

Рисунки и диаграммы должны иметь прямое отношение к тексту, без лишних изображений и данных, которые нигде не поясняются. Количество иллюстраций в работе должно быть достаточным для пояснения излагаемого текста. Иллюстрации следует размещать как можно ближе к соответствующим частям текста. На все иллюстрации должны иметь ссылки в тексте работы. Наименования, приводимые в тексте и на иллюстрациях, должны быть одинаковыми.

Ссылки на иллюстрации разрешается помещать в скобках в соответствующем месте текста, без указания *см.* (смотри). Ссылки на ранее упомянутые иллюстрации записывают, сокращенным словом *смотри*, например, *см. рисунок 3.*

Размещаемые в тексте иллюстрации следует нумеровать арабскими цифрами, например: *Рисунок 1, Рисунок 2* и т.д. Допускается нумеровать иллюстрации в пределах раздела. В этом случае номер иллюстрации должен состоять из номера раздела и порядкового номера иллюстрации, например *Рисунок 1.1.*

Надписи, загромождающие рисунок, чертеж или схему, необходимо помещать в тексте или под иллюстрацией.

3.3. Оформление таблиц

Цифровой материал, как правило, оформляют в виде таблиц. Название таблицы должно отражать её содержание, быть точным и кратким. Лишь в порядке исключения таблица может не иметь названия.

Таблицы в пределах всей записки нумеруют арабскими цифрами сквозной нумерацией, перед которыми записывают слово *Таблица*. Допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой.

Пример:

Таблица 7 - Классификация связей информационной системы

Родительская таблица	Дочерняя таблица	Тип связи	Обоснование
Проживающий	Оплата	1:M	На один приказ может оформляться множество платежей, по месяцам
Проживающий	Заселение	1:M	На одного студента может подписываться несколько приказов, по годам.

На все таблицы должны быть ссылки в тексте, при этом слово таблица в тексте пишут полностью, например: *в таблице 4.*

Таблицу, в зависимости от ее размера, помещают под текстом, в котором впервые дана ссылка на нее, или на следующей странице, а при необходимости, в приложении. Допускается помещать таблицу вдоль стороны листа.

Если строки или графы таблицы выходят за формат страницы, ее делят на части, помещая одну часть под другой, при этом в каждой части таблицы повторяют ее шапку и боковик.

При переносе таблицы на другой лист (страницу), шапку таблицы повторяют и над ней указывают: *Продолжение таблицы 5.* Название таблицы помещают только над первой частью таблицы.

В графах таблиц не допускается проводить диагональные линии с разноской заголовков вертикальных глав по обе стороны диагонали.

Основные заголовки следует располагать в верхней части шапки таблицы над дополнительными и подчиненными заголовками вертикальных граф. Заголовки граф, как правило, записывают параллельно строкам таблицы. При необходимости допускается перпендикулярное расположение заголовков граф.

Все слова в заголовках и надписях шапки и боковика таблицы пишут полностью, без сокращений. Допускаются лишь те сокращения, которые приняты в тексте, как при числах, так и без них. Следует избегать громоздкого построения таблиц с «многоэтажной» шапкой. Все заголовки надо писать по возможности просто и кратко.

3.4. Оформление приложений

Приложения оформляют как продолжение основного текста на последующих листах или в виде самостоятельного документа.

В основном тексте на все приложения должны быть даны ссылки.

Приложения располагают в последовательности ссылок на них в тексте. Каждое приложение должно начинаться с нового листа (страницы) с указанием в правом верхнем углу страницы слова *Приложение* и номера.

Приложения обозначают арабскими цифрами, за исключением цифры 0. Обозначение приложений римскими цифрами не допускается.

Приложение должно иметь заголовок, который записывают с прописной буквы отдельной строкой.

ВНИМАНИЕ! Выполненная курсовая работа сдается руководителю на проверку.

Перед сдачей работы Вы должны проверить соблюдение всех необходимых требований по ее содержанию и оформлению. Несоблюдение требований может повлиять на оценку или курсовая работа может быть возвращена для доработки, а также повторного выполнения.

3.5. Требования к лингвистическому оформлению курсового проекта

Курсовая работа должна быть написана логически последовательно, литературным языком. Повторное употребление одного и того же слова, если это возможно, допустимо через 50 – 100 слов. Не должны употребляться как излишне пространные и сложно построенные предложения, так и чрезмерно краткие лаконичные фразы, слабо между собой связанные, допускающие двойные толкования и т. д.

При написании курсовой работы не рекомендуется вести изложение от первого лица единственного числа: «я наблюдал», «я считаю», «по моему мнению» и т. д. Корректнее использовать местоимение «мы». Допускаются обороты с сохранением первого лица множественного числа, в которых исключается местоимение «мы», то есть фразы строятся с употреблением слов «наблюдаем», «устанавливаем», «имеем». Можно использовать выражения «на наш взгляд», «по нашему мнению», однако предпочтительнее выражать ту же мысль в безличной форме, например:

- *изучение предметной области свидетельствует о том, что …,*
- *на основе выполненного анализа можно утверждать …,*
- *проведенные исследования подтвердили…;*
- *представляется целесообразным отметить;*
- *установлено, что;*
- *делается вывод о…;*
- *следует подчеркнуть, выделить;*
- *можно сделать вывод о том, что;*
- *необходимо рассмотреть, изучить, дополнить;*
- *в работе рассматриваются, анализируются...*

При написании курсовой работы необходимо пользоваться языком научного изложения. Здесь могут быть использованы следующие слова и выражения:

для указания на последовательность развития мысли и временную соотнесенность:

- *прежде всего, сначала, в первую очередь;*
- *во – первых, во – вторых и т. д.;*
- *затем, далее, в заключение, итак, наконец;*
- *до сих пор, ранее, в предыдущих исследованиях, до настоящего времени;*
- *в последние годы, десятилетия;*

для сопоставления и противопоставления:

- *однако, в то время как, тем не менее, но, вместе с тем;*
- *как…, так и…;*
- *с одной стороны…, с другой стороны, не только…, но и;*
- *по сравнению, в отличие, в противоположность;*

для указания на следствие, причинность:

- *таким образом, следовательно, итак, в связи с этим;*

– *отсюда следует, понятно, ясно;*
– *это позволяет сделать вывод, заключение;*
– *свидетельствует, говорит, дает возможность;*
– *в результате;*

для дополнения и уточнения:

– *помимо этого, кроме того, также и, наряду с…, в частности;*
– *главным образом, особенно, именно;*

для иллюстрации сказанного:

– *например, так;*
– *проиллюстрируем сказанное следующим примером, приведем пример;*
– *подтверждением выше сказанного является;*

для ссылки на предыдущие высказывания, мнения, исследования и т.д.:

– *было установлено, рассмотрено, выявлено, проанализировано;*
– *как говорилось, отмечалось, подчеркивалось;*
– *аналогичный, подобный, идентичный анализ, результат;*
– *по мнению Х, как отмечает Х, согласно теории Х;*

для введения новой информации:

– *рассмотрим следующие случаи, дополнительные примеры;*
– *перейдем к рассмотрению, анализу, описанию;*
– *остановимся более детально на…;*
– *следующим вопросом является…;*
– *еще одним важнейшим аспектом изучаемой проблемы является…;*

для выражения логических связей между частями высказывания:

– *как показал анализ, как было сказано выше;*
– *на основании полученных данных;*
– *проведенное исследование позволяет сделать вывод;*
– *резюмируя сказанное;*
– *дальнейшие перспективы исследования связаны с….*

Письменная речь требует использования в тексте большого числа развернутых предложений, включающих придаточные предложения, причастные и деепричастные обороты. В связи с этим часто употребляются составные подчинительные союзы и клише:

– *поскольку, благодаря тому что, в соответствии с…;*
– *в связи, в результате;*
– *при условии, что, несмотря на…;*
– *наряду с…, в течение, в ходе, по мере.*

Необходимо определить основные понятия по теме исследования, чтобы использование их в тексте курсовой работы было однозначным. Это означает: то или иное понятие, которое разными учеными может трактоваться по-разному, должно во всем тексте данной работы от начала до конца иметь лишь одно, четко определенное автором курсовой работы значение.

В курсовой работе должно быть соблюдено единство стиля изложения, обеспечена орфографическая, синтаксическая и стилистическая грамотность в соответствии с нормами современного русского языка.

Приложение 2. Оформление титульного листа, ведомости проекта

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ
ГАПОУ СО «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

КУРСОВОЙ ПРОЕКТ
по МДК 05.03 «Тестирование информационных систем»

ТЕМА: «_____»

0	9	0	2	0	7	- КП -										

ОЦЕНКА: _____
РУКОВОДИТЕЛЬ КУРСОВОГО ПРОЕКТА:
_____(_____)
СТУДЕНТ гр. _____(_____)

Екатеринбург
2025

№ стр.	Формат	Обозначение	Наименование	Кол. листов	№ экз.	Примеч.
1	А4	ЕМК 09.02.07. КП 3518-ИП81-18 ПЗ	Пояснительная записка	50	-	
2		ЕМК 09.02.07.КП 3518- ИП81-18 ИС	«Тема»	1		CD-диск
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						

					ЕМК 09.02.07 КП 3518-ИП81-18	
Изм.	Лист	№ докум.	Подпись	Дата		
Разраб.		ФИО студента.			Ведомость курсового проекта	Лит. / Лист 79 / Листов 50
Провер.		ФИО консультанта				
Реценз.						ГАПОУ СО «ЕМК» Группа ИП-81
Н. Контр.						
Утверд.						

Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.					Ведомость курсового проекта	Лит.	Лист 80 / Листов 82
Провер.							
Реценз.							
Н. контр.		.					
Утв.							