

Документ подписан министром науки и высшего образования Российской Федерации
Информация о владельце: федеральное государственное бюджетное образовательное учреждение высшего образования
ФИО: Выборнова Любовь Александровна «Поволжский государственный университет сервиса» (ФГБОУ ВО «ПВГУС»)
Должность: Ректор
Дата подписания: 31.08.2026 09:48:13
Уникальный программный ключ:
c3b3b9c625f6c113afa2a2c42baff9e05a38b76e

Высшая школа интеллектуальных систем и кибертехнологий

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«ЯЗЫКИ ПРОГРАММИРОВАНИЯ»**

Программа бакалавриата по направлению подготовки
09.03.02 «Информационные системы и технологии»

Образовательная программа:
«Информационные системы и технологии»

Квалификация выпускника: бакалавр

Тольятти 2025

Рабочая программа дисциплины «Программирование» разработана в соответствии с федеральным государственным образовательным стандартом высшего образования - бакалавриат по направлению подготовки 09.03.02 Информационные системы и технологии, утвержденным приказом Министерства науки и высшего образования Российской Федерации от 17 ноября 2020 г. N 1427.

Составители:

старший преподаватель Коновалова А.Д.
(ученая степень, ученое звание) (ФИО)

РПД обсуждена на заседании Высшей школы интеллектуальных систем и кибертехнологий

« » _____ 202_ г., протокол № 2 _____

к.э.н., доцент

Директор ВШИСиКТ

Филлипова О.А.

(уч.степень, уч.звание)

(ФИО)

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1.1. Цель освоения дисциплины

Целью освоения дисциплины является:

– формирование у обучающихся общепрофессиональных компетенций, обеспечивающих разработку алгоритмов и программ, пригодных для практического применения в области информационных систем и технологий.

– формирование профессиональных компетенций по разработке и тестированию прототипов информационных систем.

1.2. Перечень планируемых результатов обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине	Основание (ПС) *для профессиональных компетенций
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	ИОПК-6.1. Применяет методы проектирования программного обеспечения ИОПК-6.2. Использует современные языки программирования для разработки алгоритмов и программ	Знает: основные парадигмы и модели выполнения языков программирования; принципы проектирования программного обеспечения; синтаксические, типовые, модульные и объектные механизмы современных языков. Умеет: сопоставлять языки программирования по модели типов, памяти, модульности и средствам абстракции; выбирать язык и проектные приемы под требования задачи; реализовывать алгоритмы на современных языках. Владеет: навыками проектирования структуры программного решения и реализации алгоритмов с применением языковых абстракций и библиотек.	—
ПК-2 Способен к выполнению работ по сопровождению и разработке прототипов ИС	ИПК-2.2. Разрабатывает прототип ИС в соответствии с требованиями ИПК-2.3. Осуществляет тестирование прототипа ИС на проверку корректности архитектурных решений	Знает: принципы прототипирования информационных систем, способы реализации функциональных требований средствами языков программирования, подходы к проверке корректности архитектурных решений. Умеет: разрабатывать прототип ИС по требованиям на выбранном языке программирования, организовывать модули и интерфейсы, проектировать и выполнять тесты прототипа. Владеет: навыками создания, интеграционного тестирования и сравнительной оценки прототипов ИС, реализованных с применением различных языковых и технологических средств.	06.015 Профессиональный стандарт «Специалист по информационным системам».

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к Блоку 1 «Дисциплины (модули)» образовательной программы (Модуль специализации. Специализация 2 «Проектирование и разработка корпоративных информационных систем»).

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3.1. Объем и структура дисциплины

Общая трудоемкость дисциплины составляет 5/5 з.е. (180/180 час.), их распределение по видам работ и формам обучения представлено в таблице.

Виды учебных занятий и работы обучающихся	Трудоемкость, час
Общая трудоемкость дисциплины, час	180/180
Контактная работа обучающихся с преподавателем по видам учебных занятий (всего), в т.ч.:	48/18
занятия лекционного типа (лекции)	18/6
занятия семинарского типа (семинары, практические занятия, практикумы, коллоквиумы и иные аналогичные занятия)	18/6
лабораторные работы	12/4
Самостоятельная работа всего, в т.ч.:	105/155
Самоподготовка по темам (разделам) дисциплины	105/155
Контроль (часы на экзамен, зачет)	27/9
Промежуточная аттестация	Экзамен

Примечание: -/- объем часов соответственно для очной, очно-заочной форм обучения

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

В процессе освоения дисциплины может применяться электронное обучение и дистанционные образовательные технологии.

В процессе освоения дисциплины обучающиеся обеспечены доступом к электронной информационно-образовательной среде и электронно-библиотечным системам.

3.1. Содержание дисциплины, структурированное по темам

Планируемые результаты освоения: код формируемой компетенции и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы				Формы текущего контроля (наименование оценочного средства)
		Контактная работа			Самостоятельная работа, час	
		Лекции, час	Лабораторные работы, час	Практические занятия, час		
ОПК-6: ИОПК-6.1 ИОПК-6.2 ПК-2: ИПК-2.2 ИПК-2.3	ТЕМА 1. ВВЕДЕНИЕ В C# И .NET. ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ Содержание лекции: 1. Обзор экосистемы .NET и места C# в ней. Эволюция языка: от C# 1.0 до современных версий. 2. Мультипарадигмальность C#: императивное, объектно-ориентированное, функциональное и компонентно-ориентированное программирование. 3. Обзор Common Language Runtime (CLR) и компиляции (Roslyn, JIT). Критерии выбора C# для корпоративных ИС.	2/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 1. Установка .NET SDK и IDE. Создание консольного приложения «Hello, World!».		2/1			Отчет по лабораторной работе
	Практическая работа 1. Построение матрицы критериев для выбора C# vs Java vs Python для заданной корпоративной ИС с учетом экосистемы, производительности и инструментов.			2/1		Отчёт по практической работе
	Самостоятельная работа.				11/12	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.1 ИОПК-6.2 ПК-2: ИПК-2.2 ИПК-2.3	ТЕМА 2. СИСТЕМА ТИПОВ C# И УПРАВЛЕНИЕ ПАМЯТЬЮ Содержание лекции: 1. Единая система типов .NET: значимые и ссылочные типы. Упаковка и распаковка. 2. Стек и управляемая куча. Принципы работы сборщика мусора. 3. Работа с неуправляемыми ресурсами. Ключевые слова ref, out, in для передачи параметров.	2/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 2. Эксперименты с типами: сравнение производительности значимых и ссылочных типов. Написание кода с упаковкой и оценка влияния на память.		2/1			Отчет по лабораторной работе
	Практическая работа 2. Анализ фрагментов кода на предмет ошибок управления памятью. Выбор между class и struct для конкретной модели данных.			2/1		Отчёт по практической работе
	Самостоятельная работа.				11/12	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.1 ИОПК-6.2	ТЕМА 3. ПРОЦЕДУРНЫЕ СРЕДСТВА C# И МОДУЛЬНОСТЬ Содержание лекции:	2/1				Лекция-визуализация

ПК-2: ИПК-2.2 ИПК-2.3	1. Методы (функции), параметры, перегрузка. Именованные и опциональные аргументы. ref, out, in и возвращаемые ссылки. 2. Пространства имен и сборки. Модификаторы доступа. 3. Организация кода в проектах и решениях. Обработка исключений и создание собственных типов исключений.					(в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 3. Реализация библиотеки классов для математических операций. Разделение на проект с логикой и консольное приложение для тестирования.		2/1			Отчет по лабораторной работе
	Практическая работа 3. Проектирование модульной структуры небольшой системы. Организация пространств имен и контроль доступа.			2/1		Отчёт по практической работе
	Самостоятельная работа.				11/12	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.1 ИОПК-6.2 ПК-2: ИПК-2.2 ИПК-2.3	ТЕМА 4. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ В C# Содержание лекции: 4. Классы и объекты. Инкапсуляция, наследование, полиморфизм. 5. Абстрактные классы и интерфейсы. Отличие интерфейсов от абстракций, множественное наследование интерфейсов. 6. Композиция vs наследование. Свойства и их использование. 7. Ключевые слова base, virtual, override, sealed. Generic-классы и методы.	2/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 4. Реализация иерархии классов для предметной области. Создание абстрактного класса и нескольких наследников.		2/1			Отчет по лабораторной работе
	Практическая работа 4. Рефакторинг кода для применения паттерна «Стратегия» или «Фабричный метод» с использованием интерфейсов.			2/1		Отчёт по практической работе
	Самостоятельная работа.				12/18	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.1 ИОПК-6.2 ПК-2: ИПК-2.2 ИПК-2.3	ТЕМА 5. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ C# Содержание лекции: 1. Лямбда-выражения и анонимные методы. 2. Делегаты и события. LINQ: синтаксис запросов и цепочки методов. 3. Неизменяемые типы. Паттерн «Строитель» с использованием функциональных цепочек. Сопоставление с образцом.	2/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 5. Обработка коллекций данных с использованием LINQ. Реализация конвейера обработки данных.		2/0			Отчет по лабораторной работе
	Практическая работа 5. Рефакторинг императивного кода с циклами.			2/1		Отчёт по практической работе

	Самостоятельная работа.				12/18	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.1 ИОПК-6.2 ПК-2: ИПК-2.2 ИПК-2.3	ТЕМА 6. ДИНАМИЧЕСКИЕ ВОЗМОЖНОСТИ И РЕФЛЕКСИЯ В C# Содержание лекции: 1. Ключевое слово dynamic. Рефлексия: получение информации о типах во время выполнения. 2. Создание экземпляров и вызов методов через Туре. Атрибуты: создание кастомных атрибутов и их использование для валидации или маршрутизации. 3. Сценарии использования: сериализация, DI-контейнеры, инструменты тестирования. Выражения.	2/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 6. Создание простой утилиты для валидации объектов на основе кастомных атрибутов.		2/0			Отчет по лабораторной работе
	Практическая работа 6. Выбор между статическим и динамическим подходом для конкретной задачи интеграции с внешним API.			2/0		Отчёт по практической работе
	Самостоятельная работа.				12/18	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.1 ИОПК-6.2 ПК-2: ИПК-2.2 ИПК-2.3	ТЕМА 7. БИБЛИОТЕКИ, ПАКЕТЫ И ВЗАИМОДЕЙСТВИЕ В .NET Содержание лекции: 1. Управление пакетами NuGet. Зависимости и семантическое версионирование. 2. Работа с JSON. Взаимодействие с внешними ресурсами. 3. Асинхронное программирование. Сериализация и форматы обмена данными.	2/0				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 7. Создание консольного клиента.		0/0			Отчет по лабораторной работе
	Практическая работа 7. Проектирование интеграционного интерфейса между двумя модулями на C# с использованием асинхронных методов и JSON/Protobuf.			2/0		Отчёт по практической работе
	Самостоятельная работа.				12/20	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.1 ИОПК-6.2 ПК-2: ИПК-2.2 ИПК-2.3	ТЕМА 8. РАЗРАБОТКА ПРОТОТИПА ИС НА C# Содержание лекции: 1. Проектирование прототипа корпоративной ИС на C#. Выбор типа приложения. 2. Организация слоев: уровень представления, бизнес-логика, и хранения данных. Работа с Entity Framework Core. 3. Внедрение зависимостей в .NET. Миграции и модель данных.	2/0				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 8. Создание минимального Web API с одним контроллером, внедрением		0/0			Отчет по лабораторной

	простого сервиса и подключением к локальной базе данных.					работе
	Практическая работа 8. Разработка вертикального среза прототипа.			2/1		Отчёт по практической работе
	Самостоятельная работа.				12/25	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.1 ИОПК-6.2 ПК-2: ИПК-2.2 ИПК-2.3	ТЕМА 9. ТЕСТИРОВАНИЕ ПРОТОТИПА НА C# Содержание лекции: 1. Уровни тестирования. Фреймворки для модульного тестирования. 2. Тест-двойники: создание моков. Интеграционное тестирование Web API. 3. Оценка архитектурной корректности: проверка зависимостей между модулями. Покрытие кода тестами.	2/0				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 9. Написание модульных и интеграционных тестов для курсового прототипа.		0/0			Отчет по лабораторной работе
	Практическая работа 9. Разработка набора тестов для проверки архитектурного решения.			2/0		Отчёт по практической работе
	Самостоятельная работа.				12/20	Самостоятельное изучение учебных материалов
ИТОГО		18/6	12/4	18/6	105/155	

Примечание: -/- объем часов соответственно для очной, очно-заочной форм обучения

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

4.1. Общие методические рекомендации по освоению дисциплины, образовательные технологии

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

При проведении учебных занятий по дисциплине обеспечивается развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей, преподавание дисциплины в форме курса, составленного на основе результатов научных исследований, проводимых университетом, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей).

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

- балльно-рейтинговая технология оценивания;
- электронное обучение;

Для оценки знаний, умений, навыков и уровня сформированности компетенции по дисциплине применяется балльно-рейтинговая система контроля и оценки успеваемости студентов. В основу балльно-рейтинговой системы положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущего контроля успеваемости. Максимальное количество баллов в семестре – 100.

4.2. Методические указания для обучающихся по освоению дисциплины на занятиях лекционного типа

Лекционный курс предполагает систематизированное изложение основных вопросов тематического плана. В ходе лекционных занятий раскрываются базовые вопросы в рамках каждой темы дисциплины. Обозначаются ключевые аспекты тем, а также делаются акценты на наиболее сложные и важные положения изучаемого материала.

Лекционные занятия проводятся в поточной аудитории с применением мультимедийного проектора в виде учебной презентации или в ЭИОС университета.

В ходе лекционных занятий рекомендуется конспектирование учебного материала.

Возможно ведение конспекта лекций в виде интеллект-карт.

Отдельные темы предлагаются для самостоятельного изучения (конспектируются).

Материалы лекций являются опорной основой для подготовки обучающихся к практическим занятиям / лабораторным работам и выполнения заданий самостоятельной работы, а также к мероприятиям текущего контроля успеваемости и промежуточной аттестации по дисциплине.

4.3. Методические указания для обучающихся по освоению дисциплины на лабораторных работах

Подготовку к каждой лабораторной работе студент должен начать с ознакомления с планом занятия, который отражает содержание предложенной темы. Каждая выполненная работа с оформленным отчетом по ней подлежит защите преподавателю.

При оценивании лабораторных работ учитывается следующее:

- качество выполнения экспериментально-практической части работы и степень соответствия результатов работы заданным требованиям;
- качество оформления отчета по работе;
- качество устных ответов на контрольные вопросы при защите работы.

Лабораторные работы организуются, в том числе в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

4.4. Методические указания для обучающихся по освоению дисциплины на занятиях семинарского типа/ на практических занятиях

Практические (семинарские) занятия представляют собой детализацию лекционного теоретического материала, проводятся в целях закрепления курса и охватывают все основные разделы. Основной формой проведения семинаров и практических занятий является обсуждение наиболее проблемных и сложных вопросов по отдельным темам, а также решение задач и разбор примеров и ситуаций в аудиторных условиях.

Практические (семинарские) занятия обучающихся обеспечивают:

- проверку и уточнение знаний, полученных на лекциях;
- получение умений и навыков составления докладов и сообщений, обсуждения вопросов по учебному материалу дисциплины;
- подведение итогов занятий по рейтинговой системе, согласно технологической карте дисциплины.

Практические занятия организуются, в том числе в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

4.5. Методические указания по самостоятельной работе обучающихся

Самостоятельная работа обеспечивает подготовку обучающегося к аудиторным занятиям и мероприятиям текущего контроля и промежуточной аттестации по изучаемой дисциплине. Результаты этой подготовки проявляются в активности обучающегося на занятиях и в качестве выполненных практических заданий и других форм текущего контроля.

Самостоятельная работа студентов включает:

1. Изучение учебной литературы по курсу.
2. Работу с ресурсами Интернет
3. Самостоятельное изучение учебных материалов

При выполнении заданий для самостоятельной работы рекомендуется проработка материалов лекций по каждой пройденной теме, а также изучение рекомендуемой литературы.

Для обучающихся по заочной форме обучения самостоятельная работа является основным видом учебной деятельности.

Для обеспечения самостоятельной работы обучающихся используется электронный учебный курс, созданный в ЭИОС университета <http://sdo.tolgas.ru/>.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Вся литература, включенная в данный перечень, представлена в виде электронных ресурсов в электронной библиотеке университета (ЭБС). Литература, используемая в печатном виде, представлена в научной библиотеке университета в объеме не менее 0,25 экземпляров на одного обучающегося.

Основная литература

1. Троелсен, Э. Язык программирования C# 10 и платформа .NET 8 / Э. Троелсен, Ф. Джепикс. – 9-е изд. – М. : Вильямс, 2024. – 1424 с. – ISBN 978-5-907-51595-8.
2. Шилдт, Г. C# 10. Полное руководство / Г. Шилдт. – М. : Диалектика, 2023. – 960 с. – ISBN 978-5-907-51545-3.
3. Фримен, А. ASP.NET Core MVC с примерами на C# 9 для профессионалов / А. Фримен. – М. : Вильямс, 2022. – 832 с. – ISBN 978-5-907-51511-8.

Дополнительная литература

4. Албахари, Д. C# 12 в кратком изложении / Д. Албахари. – СПб. : Питер, 2024. – 1056 с. – ISBN 978-5-4461-2388-3.
5. Стиллмен, Э. Изучаем C# через разработку игр на Unity / Э. Стиллмен. – 5-е изд. – М. : Эксмо, 2024. – 544 с. – ISBN 978-5-04-203074-1.
6. Рихтер, Д. CLR via C# / Д. Рихтер. – 4-е изд. – М. : Питер, 2023. – 896 с. – ISBN 978-5-4461-2340-1.
7. Сейкон, Р. Entity Framework Core в действии / Р. Сейкон. – М. : ДМК-Пресс, 2022. – 528 с. – ISBN 978-5-93700-145-4.
8. Эванс, Э. Предметно-ориентированное проектирование (DDD) / Э. Эванс. – М. : Вильямс, 2023. – 448 с. – ISBN 978-5-8459-2045-0.

5.2. Профессиональные базы данных, информационно-справочные системы, интернет-ресурсы

1. КонсультантПлюс [Электронный ресурс]: справочная правовая система. – Режим доступа: <http://www.consultant.ru/>.
2. Электронная библиотечная система Поволжского государственного университета сервиса [Электронный ресурс]. – Режим доступа: <http://elib.tolgas.ru/>.
3. Электронно-библиотечная система Znanium.com [Электронный ресурс]. – Режим доступа: <http://znanium.com/>.
4. Электронно-библиотечная система «Издательство Лань» [Электронный ресурс]. – Режим доступа: <https://e.lanbook.com/>.
5. Научная электронная библиотека eLIBRARY.RU [Электронный ресурс]. – Режим доступа: <http://elibrary.ru/defaultx.asp>.
6. Открытое образование [Электронный ресурс]. – Режим доступа: <https://openedu.ru/>.
7. Базы данных Всероссийского института научной и технической информации (ВИНИТИ РАН) по естественным, точным и техническим наукам [Электронный ресурс]. – Режим доступа: <http://www.viniti.ru>.

5.3. Программное обеспечение

Информационное обеспечение учебного процесса по дисциплине осуществляется с использованием следующего программного обеспечения (лицензионного и свободно распространяемого), в том числе отечественного производства:

№ п/п	Наименование	Условия доступа
1	Microsoft Windows	из внутренней сети университета (лицензионный договор)
2	Microsoft Office	из внутренней сети университета (лицензионный договор)
3	КонсультантПлюс	из внутренней сети университета (лицензионный договор)
4	СДО MOODLE	из любой точки, в которой имеется доступ к сети Интернет (лицензионный договор)
5	Google	из любой точки, в которой имеется доступ к сети Интернет (свободно распространяемое)
6	Yandex	из любой точки, в которой имеется доступ к сети Интернет (свободно распространяемое)
7	Среда разработки и компилятор/интерпретатор языка программирования, используемого на занятиях	свободно распространяемое или установленное в компьютерном классе программное обеспечение; конкретный язык определяется учебным заданием

6. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных учебным планом и рабочей программой дисциплины, оснащенные оборудованием и техническими средствами обучения.

Занятия лекционного типа. Учебные аудитории для занятий лекционного типа укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации (стационарные или переносные наборы демонстрационного оборудования (проектор, экран, компьютер/ноутбук), учебно-наглядные пособия (презентации по темам лекций), обеспечивающие тематические иллюстрации, соответствующие данной программе дисциплины.

Занятия семинарского типа. Учебные аудитории для занятий семинарского типа укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации (стационарные или переносные наборы демонстрационного оборудования (проектор, экран, компьютер/ноутбук). Для проведения лабораторных работ используется учебная аудитория «Лаборатория информационных технологий, информатики и методов программирования // Учебная аудитория для проведения занятий лекционного типа, семинарского типа, курсового проектирования, консультаций, текущего контроля и промежуточной аттестации // Помещение для хранения и профилактического обслуживания учебного оборудования (Г-413)», оснащенная следующим оборудованием: рабочие места обучающихся; Рабочее место преподавателя; Учебно-наглядные пособия, обеспечивающие тематические иллюстрации. Компьютер Асус Intel G530\iH61\GeForceGT430\4096 Mb DDR3\500Gb\LCD 22" - 14 шт., Компьютер в сборе iP2.0 LCD - 1 шт., принтер HP 1200 - 1 шт., мобильный экран проекционный DraperLuma - 1 шт., мобильный проектор Sanyo PLC, Сетевое оборудование D-Link DES-1024D, 24-портовый; локальная сеть 100 мб с доступом в Интернет 50 мб. Лаборатория оснащена кондиционером.

Лабораторные работы. Для проведения лабораторных работ используется учебная аудитория «Лаборатория информационных технологий, информатики и методов программирования // Учебная аудитория для проведения занятий лекционного типа, семинарского типа, курсового проектирования, консультаций, текущего контроля и промежуточной аттестации // Помещение для хранения и профилактического обслуживания учебного оборудования (Г-413)», оснащенная следующим оборудованием: рабочие места обучающихся; Рабочее место преподавателя; Учебно-наглядные пособия, обеспечивающие тематические иллюстрации. Компьютер Аксус Intel G530\iH61\GeForceGT430\4096 Mb DDR3\500Gb\LCD 22" - 14 шт., Компьютер в сборе iP2.0 LCD - 1 шт., принтер HP 1200 - 1 шт., мобильный экран проекционный DraperLuma - 1 шт., мобильный проектор Sanyo PLC, Сетевое оборудование D-Link DES-1024D, 24-портовый; локальная сеть 100 мб с доступом в Интернет 50 мб. Лаборатория оснащена кондиционером.

Промежуточная аттестация. Для проведения промежуточной аттестации по дисциплине используются компьютерные классы, оснащенные компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду университета и/или учебные аудитории, укомплектованные мебелью и техническими средствами обучения.

Самостоятельная работа. Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде университета. Для организации самостоятельной работы обучающихся используются:

- компьютерные классы университета;
- библиотека (медиазал), имеющая места для обучающихся, оснащенные компьютерами с доступом к базам данных и сети Интернет.

Электронная информационно-образовательная среда университета (ЭИОС). Каждый обучающийся в течение всего периода обучения обеспечен индивидуальным неограниченным доступом к электронной информационно-образовательной среде университета (ЭИОС) <http://sdo.tolgas.ru/> из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети "Интернет", как на территории университета, так и вне ее.

ЭИОС университета обеспечивает:

- доступ к учебным планам, рабочим программам дисциплин (модулей), программам практик, электронным учебным изданиям и электронным образовательным ресурсам, указанным в рабочих программах дисциплин (модулей), программах практик;
- формирование электронного портфолио обучающегося, в том числе сохранение его работ и оценок за эти работы.

В случае реализации образовательной программы с применением электронного обучения, дистанционных образовательных технологий ЭИОС дополнительно обеспечивает:

фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательной программы;

проведение учебных занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением электронного обучения, дистанционных образовательных технологий;

взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействия посредством сети "Интернет".

7. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБУЧЕНИЯ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

При необходимости рабочая программа дисциплины может быть адаптирована для обеспечения образовательного процесса инвалидов и лиц с ограниченными возможностями здоровья, в том числе для дистанционного обучения. Для этого требуется заявление студента (его законного представителя) и заключение психолого-медико-педагогической комиссии (ПМПК).

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида, могут предлагаться следующие варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.
- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.
- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

8. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (ФОНД ОЦЕНОЧНЫХ СРЕДСТВ) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

8.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Для оценки знаний, умений, навыков и уровня сформированности компетенции по дисциплине применяется балльно-рейтинговая система контроля и оценки успеваемости студентов. В основу балльно-рейтинговой системы положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущего контроля успеваемости. Максимальное количество баллов в семестре – 100.

Шкала оценки результатов освоения дисциплины, сформированности результатов обучения

Форма промежуточной аттестации	Уровень сформированности	100-балльная шкала, %	Шкала освоения, %	5-балльная дифференцированная оценка/балл
экзамен	допороговый	ниже 61	ниже 61	«неудовлетворительно» / 2
	пороговый	61–85,9	61–69,9	«удовлетворительно» / 3
	пороговый	61–85,9	70–85,9	«хорошо» / 4
	повышенный	86–100	86–100	«отлично» / 5

По итогам текущей успеваемости студенту может быть выставлена оценка по промежуточной аттестации в соответствии с набранными за семестр баллами (по накопительному рейтингу). Студентам, набравшим в ходе текущего контроля успеваемости по дисциплине от 61 до 100 баллов и выполнившим все обязательные виды запланированных учебных занятий, по решению преподавателя без прохождения промежуточной аттестации выставляется оценка в соответствии со шкалой оценки результатов освоения дисциплины.

Результат обучения считается сформированным (повышенный уровень), если теоретическое содержание курса освоено полностью; при устных собеседованиях студент исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 86 до 100, что соответствует повышенному уровню сформированности результатов обучения.

Результат обучения считается сформированным (пороговый уровень), если теоретическое содержание курса освоено полностью; при устных собеседованиях студент последовательно, четко и логически стройно излагает учебный материал; справляется с задачами, вопросами и другими видами заданий, требующих применения знаний; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 61 до 85,9, что соответствует пороговому уровню сформированности результатов обучения.

Результат обучения считается несформированным, если студент при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет задания, не демонстрирует необходимых умений, качество выполненных заданий не соответствует установленным требованиям, качество их выполнения оценено числом баллов ниже 61, что соответствует допороговому уровню.

Формы текущего контроля успеваемости

Формы текущего контроля	Количество контрольных точек	Количество баллов за 1 контр. точку	Макс. возм. кол-во баллов
Посещение лекционного занятия (конспект лекций)	9	2	18
Выполнение лабораторных работ	9	4	36
Выполнение практических работ	9	4	36
Творческий рейтинг (участие в конференциях, олимпиадах, проектной деятельности и т.п.)	1	10	10
Итого по дисциплине			100 баллов

Система оценивания представлена в электронном учебном курсе по дисциплине <http://sdo.tolgas.ru/>.

8.2. Типовые контрольные задания или иные материалы для ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

8.2.1. Типовые задания к практическим (семинарским) занятиям

Практическая работа 1. Построение матрицы критериев для выбора C# vs Java vs Python для заданной корпоративной ИС с учетом экосистемы, производительности и инструментов

Задание 1. Построить матрицу критериев выбора языка программирования для корпоративной ИС с учётом экосистемы, производительности и доступных инструментов.

Задание 2. Определить доминирующую парадигму программирования для заданного фрагмента кода.

Задание 3. Обосновать выбор C# для проекта по управлению заявками, сравнив его с Java и Python по 5–7 критериям.

Задание 4. Подготовить устное обоснование выбора технологического стека для заданной предметной области.

Практическая работа 2. Анализ фрагментов кода на предмет ошибок управления памятью. Выбор между class и struct для конкретной модели данных

Задание 1. Проанализировать фрагмент кода на C# и определить ошибки управления памятью.

Задание 2. Выявить упаковку и распаковку в предоставленном коде и предложить исправления.

Задание 3. Обосновать выбор между class и struct для модели данных «Точка» и «Пользователь» с учётом частоты создания экземпляров и объёма данных.

Задание 4. Сравнить поведение значимых и ссылочных типов при передаче в метод с параметрами ref, out и in.

Практическая работа 3. Проектирование модульной структуры небольшой системы. Организация пространств имен и контроль доступа

Задание 1. Спроектировать модульную структуру небольшой системы (калькулятор с историей) с разделением на пространства имён и проекты.

Задание 2. Организовать контроль доступа к классам и методам с использованием модификаторов public, internal, private.

Задание 3. Разработать интерфейс библиотеки классов для математических операций и продемонстрировать его использование в консольном приложении.

Задание 4. Реализовать обработку исключений с созданием собственных типов исключений.

Практическая работа 4. Рефакторинг кода для применения паттерна «Стратегия» или «Фабричный метод» с использованием интерфейсов

Задание 1. Выполнить рефакторинг кода, заменив условные операторы на паттерн «Стратегия» с использованием интерфейсов.

Задание 2. Реализовать паттерн «Фабричный метод» для создания объектов семейства классов.

Задание 3. Разработать иерархию классов для предметной области «Транспортные средства» с применением виртуальных и абстрактных методов.

Задание 4. Спроектировать интерфейс и реализовать его для работы с коллекцией объектов в памяти.

Практическая работа 5. Рефакторинг императивного кода с циклами

Задание 1. Выполнить рефакторинг императивного кода с циклами и условиями на функциональный стиль с использованием LINQ и лямбда-выражений.

Задание 2. Реализовать цепочку методов для фильтрации, сортировки и группировки коллекции объектов по заданному условию.

Задание 3. Преобразовать список строк в словарь с подсчётом длины каждого слова с использованием LINQ.

Задание 4. Разработать конвейер обработки данных с применением функциональных цепочек.

Практическая работа 6. Выбор между статическим и динамическим подходом для конкретной задачи интеграции с внешним API

Задание 1. Обосновать выбор между статическим и динамическим подходом для задачи интеграции с внешним API.

Задание 2. С использованием рефлексии вывести информацию о всех публичных методах и свойствах заданного класса.

Задание 3. Разработать кастомный атрибут и написать утилиту для валидации объектов на основе этого атрибута.

Задание 4. Создать экземпляр класса и вызвать его метод по имени строки с использованием рефлексии.

Практическая работа 7. Проектирование интеграционного интерфейса между двумя модулями на C# с использованием асинхронных методов и JSON/Protobuf

Задание 1. Спроектировать интеграционный интерфейс между двумя модулями на C# с использованием асинхронных методов.

Задание 2. Выбрать и обосновать формат обмена данными (JSON или Protobuf) для заданного сценария интеграции.

Задание 3. Подключить через NuGet библиотеку для работы с JSON и выполнить сериализацию/десериализацию объекта.

Задание 4. Реализовать асинхронный метод, выполняющий HTTP-запрос к внешнему ресурсу и обрабатывающий ответ.

Практическая работа 8. Разработка вертикального среза прототипа

Задание 1. Разработать вертикальный срез прототипа ИС (REST API с одним контроллером) по заданным функциональным требованиям.

Задание 2. Организовать слои бизнес-логики и хранения данных с внедрением зависимостей (DI-контейнер).

Задание 3. Подключить Entity Framework Core и выполнить миграцию для создания модели данных.

Задание 4. Реализовать CRUD-операции через Web API и протестировать их с помощью Swagger.

Практическая работа 9. Разработка набора тестов для проверки архитектурного решения

Задание 1. Разработать набор модульных тестов для проверки бизнес-логики прототипа.

Задание 2. Создать тест-двойники (моки) для изоляции зависимостей от базы данных.

Задание 3. Написать интеграционные тесты для проверки взаимодействия Web API с реальной базой данных.

Задание 4. Разработать набор тестов для проверки архитектурной корректности: отсутствие циклических зависимостей между модулями.

8.2.2. Типовые задания для лабораторных работ

Лабораторная работа 1. Установка .NET SDK и IDE. Создание консольного приложения «Hello, World!»

Задание 1. Установить .NET SDK и выбранную IDE. Создать консольное приложение «Hello, World!» и запустить его.

Задание 2. Модифицировать программу для вывода текущей даты и времени с использованием встроенного класса DateTime.

Задание 3. Создать проект с несколькими классами, демонстрирующими разные парадигмы программирования.

Задание 4. Скомпилировать и запустить приложение через командную строку (dotnet run) без использования IDE.

Лабораторная работа 2. Эксперименты с типами: сравнение производительности значимых и ссылочных типов. Написание кода с упаковкой и оценка влияния на память

Задание 1. Написать программу, демонстрирующую разницу между значимыми и ссылочными типами при присваивании.

Задание 2. Создать эксперимент с упаковкой и распаковкой: поместить значимый тип в object и обратно, измерив влияние на производительность.

Задание 3. Реализовать класс с деструктором и методом Dispose для освобождения неуправляемых ресурсов.

Задание 4. Сравнить производительность передачи параметров по значению, по ссылке (ref) и через in для структуры большого размера.

Лабораторная работа 3. Реализация библиотеки классов для математических операций. Разделение на проект с логикой и консольное приложение для тестирования

Задание 1. Реализовать библиотеку классов для математических операций с обработкой ошибок.

Задание 2. Создать отдельный консольный проект, подключить к нему библиотеку и написать тестовый вывод.

Задание 3. Организовать код в пространства имён: один для математических функций, второй — для вспомогательных утилит.

Задание 4. Разделить решение на три проекта: библиотека логики, консольное приложение и проект с модульными тестами.

Лабораторная работа 4. Реализация иерархии классов для предметной области. Создание абстрактного класса и нескольких наследников

Задание 1. Создать абстрактный класс и несколько наследников с переопределением виртуальных методов.

Задание 2. Реализовать интерфейс для класса и выполнить сортировку списка объектов.

Задание 3. Создать generic-класс с методами для хранения коллекции объектов любого типа.

Задание 4. Реализовать композицию, обеспечивая инкапсуляцию и делегирование вызовов.

Лабораторная работа 5. Обработка коллекций данных с использованием LINQ. Реализация конвейера обработки данных

Задание 1. Дана коллекция целых чисел. С помощью LINQ выбрать чётные числа, отсортировать по убыванию и вывести в консоль.

Задание 2. Реализовать конвейер обработки данных: прочитать список строк, отфильтровать, преобразовать, сгруппировать.

Задание 3. Написать лямбда-выражение для поиска максимального элемента в коллекции.

Задание 4. Создать делегат и присвоить ему лямбда-выражение для выполнения арифметической операции.

Лабораторная работа 6. Создание простой утилиты для валидации объектов на основе кастомных атрибутов

Задание 1. Создать класс с несколькими публичными и приватными полями и вывести информацию о всех членах класса с помощью рефлексии.

Задание 2. Разработать кастомный атрибут и написать утилиту для проверки свойств объекта.

Задание 3. Используя `dynamic`, создать объект-обёртку для обращения к свойствам через строковые ключи.

Задание 4. Написать программу, загружающую сборку и вызывающую метод по строке-имени.

Лабораторная работа 7. Создание консольного клиента

Задание 1. Установить пакет `Newtonsoft.Json` через `NuGet` и выполнить сериализацию/десериализацию объекта.

Задание 2. Написать консольный клиент, отправляющий HTTP-запрос к публичному API и выводящий результат.

Задание 3. Реализовать асинхронный метод для загрузки содержимого страницы по URL.

Задание 4. Создать клиент-серверное взаимодействие с передачей данных в формате JSON.

Лабораторная работа 8. Создание минимального Web API с одним контроллером, внедрением простого сервиса и подключением к локальной базе данных

Задание 1. Создать Web API с одним контроллером, возвращающим список товаров.

Задание 2. Подключить `Entity Framework Core` и создать локальную базу данных `SQLite`.

Задание 3. Реализовать внедрение зависимости для сервиса в контроллер.

Задание 4. Выполнить миграцию и добавить тестовые данные, проверить работу CRUD-операций через `Swagger`.

Лабораторная работа 9. Написание модульных и интеграционных тестов для курсового прототипа

Задание 1. Написать модульные тесты для сервиса с использованием мока репозитория.

Задание 2. Реализовать интеграционные тесты для проверки контроллера с реальным подключением к базе.

Задание 3. Разработать тест, проверяющий возврат статуса 404 при удалении несуществующего объекта.

Задание 4. Измерить покрытие кода тестами и написать отчёт по результатам проверки архитектурной корректности.

8.3. Типовые контрольные задания или иные материалы для проведения ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Форма проведения промежуточной аттестации по дисциплине: *зачет и экзамен (по результатам накопительного рейтинга или в форме компьютерного тестирования).*

Устно-письменная форма по экзаменационным билетам предполагается, как правило, для сдачи академической задолженности.

Перечень вопросов и заданий для подготовки к экзамену

ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

1. Понятие парадигмы программирования. Мультипарадигмальность C# как основное преимущество языка.
2. Архитектура .NET: CLR, CTS, CLS. Жизненный цикл выполнения программы на C# (компиляция Roslyn, JIT, выполнение).
3. Единая система типов .NET: значимые и ссылочные типы. Упаковка и распаковка. Влияние на производительность.
4. Управление памятью в .NET: стек и управляемая куча. Принципы работы сборщика мусора (поколения, алгоритмы).
5. Работа с неуправляемыми ресурсами: IDisposable, using, финализаторы.
6. Передача параметров в методы C#: ref, out, in. Возвращаемые ссылки (ref return).
7. Методы в C#: перегрузка, именованные и опциональные аргументы. Ключевое слово params.
8. Пространства имен и сборки. Модификаторы доступа и их роль в инкапсуляции.
9. Обработка исключений в C#. Иерархия исключений. Создание собственных исключений.
10. Объектно-ориентированное программирование в C#: классы и объекты. Инкапсуляция, наследование, полиморфизм.
11. Абстрактные классы и интерфейсы в C#: сравнение, области применения. Множественное наследование интерфейсов.
12. Композиция vs наследование: сравнение подходов, выбор стратегии проектирования.
13. Свойства в C#. Автоматические свойства. Вычисляемые свойства.
14. Ключевые слова base, virtual, override, sealed, new. Переопределение и сокрытие методов.
15. Generic-классы и Generic-методы. Ограничения параметров-типов (where T : class, struct, new() и др.).
16. Делегаты в C#: объявление, создание экземпляров, комбинирование. Анонимные методы.
17. События в C#. Механизм подписки и отписки. Связь событий и делегатов.
18. Лямбда-выражения в C#. Синтаксис и применение. Замыкания.
19. LINQ в C#: архитектура, синтаксис запросов, цепочки методов. Отложенное и немедленное выполнение.
20. LINQ to Objects: основные методы расширения (Where, Select, SelectMany, OrderBy, GroupBy, Aggregate, Any, All, First, Single и др.).
21. Сопоставление с образцом (pattern matching) в C#: is, switch, паттерны свойств, позиционные паттерны.
22. Неизменяемость в C#. Паттерн «Строитель» с функциональными цепочками. Создание иммутабельных типов.
23. Рефлексия в C#: класс Type, получение информации о сборках, типах, членах.
24. Создание экземпляров и вызов методов через рефлексия. Динамическое связывание.
25. Атрибуты в C#: синтаксис создания, использование, получение через рефлексия. Сценарии применения.
26. Ключевое слово dynamic. Отличия от var. Сценарии использования в интеграции с COM и динамическими API.
27. Управление зависимостями в .NET: NuGet. Файл проекта. Семантическое версионирование.

28. Асинхронное программирование в C#: Task, async/await. Синхронизация контекста.
29. Работа с данными в .NET: System.Text.Json, Newtonsoft.Json. Сериализация в различные форматы.
30. Взаимодействие с внешними ресурсами: HttpClient, асинхронные запросы, обработка ответов.
31. Стандартная библиотека базовых классов .NET (BCL). Основные пространства имен (System, System.Collections.Generic, System.Linq, System.IO, System.Threading.Tasks и др.).
32. Модульность в .NET: проекты, решения, сборки. Типы сборок (исполняемые, библиотечные).

ПК-2. Способен к выполнению работ по сопровождению и разработке прототипов ИС

33. Прототипирование информационных систем: назначение, виды прототипов, критерии готовности.
34. Выбор технологического стека для прототипа: факторы, влияющие на решение (требования проекта, команда, инфраструктура).
35. Архитектура прототипа: слои (представление, бизнес-логика, хранение данных). Организация зависимостей.
36. Внедрение зависимостей (Dependency Injection) в .NET. Встроенный DI-контейнер.
37. Проектирование модели данных прототипа. Связи между сущностями (one-to-many, many-to-many).
38. Entity Framework Core: подходы (Code-First, Database-First). Миграции и управление схемой данных.
39. Реализация вертикального среза функциональности: разработка контроллера, сервиса и репозитория.
40. Создание Web API в .NET: атрибуты маршрутизации, привязка моделей, форматы ответов.
41. Уровни тестирования: модульное, интеграционное, системное. Роль каждого уровня в проверке прототипа.
42. Модульное тестирование в .NET: фреймворки (NUnit, xUnit), структура теста (Arrange-Act-Assert).
43. Тест-двойники: моки, стабы, шпионы. Использование библиотек-мокеров (Moq, NSubstitute).
44. Интеграционное тестирование: проверка взаимодействия слоев, тестирование базы данных, тестирование API.
45. Проверка архитектурной корректности. Инструменты для анализа зависимостей между модулями.
46. Покрытие кода тестами. Пороговые значения и методы увеличения покрытия.
47. Документирование архитектурных решений и результатов тестирования прототипа.
48. Сравнительная оценка прототипов на разных языках и технологических платформах.
49. Подготовка прототипа к демонстрации: оформление, инструкция по запуску, демонстрационный сценарий.
50. Практическое задание: разработка набора модульных и интеграционных тестов для проверки архитектурного решения прототипа.

Типовые практические задания при промежуточной аттестации

1. Сравнить две реализации одного алгоритма на разных языках по заданным критериям (производительность, читаемость, сопровождаемость).
2. Выбрать язык и технологический стек для прототипа ИС и обосновать решение по заданным функциональным и нефункциональным требованиям.
3. Спроектировать модульный интерфейс и показать его реализацию средствами выбранного языка.
4. Разработать тесты, проверяющие взаимодействие модулей прототипа.
5. По результатам теста определить нарушение архитектурного ограничения и предложить исправление.

6. Реализовать иерархию классов для заданной предметной области с использованием абстрактных классов и интерфейсов.
7. Создать минимальное Web API с одним контроллером, внедрением простого сервиса и подключением к локальной базе данных.
8. Разработать конвейер обработки данных с использованием LINQ и функциональных цепочек.
9. Спроектировать кастомный атрибут и реализовать утилиту для валидации объектов на основе этого атрибута с использованием рефлексии.