



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГЕОДЕЗИИ И КАРТОГРАФИИ
(МИИГАиК)

Факультет геоинформатики и информационной безопасности
Кафедра информационно-измерительных систем



**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5387FD2AF5B17FED816FDE9A1D4973B1
Владелец: Степанченко Алексей Леонидович
Действителен: с 21.07.2025 по 14.10.2026

«УТВЕРЖДАЮ»

Проректор
А. Л. Степанченко
16 февраля 2024 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ПРАКТИКЕ
«Б2.О.03(У) УЧЕБНАЯ (ОЗНАКОМИТЕЛЬНАЯ) ПРАКТИКА ПО
ПРОГРАММИРОВАНИЮ»**

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.03 Прикладная информатика

Направленность(профиль) подготовки: Прикладная информатика

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2024

Срок получения образования: 4 года

Объем:
в зачетных единицах: 5 з.е.
в академических часах: 180 ак.ч.

Москва, 2024

Разработчики:

Преподаватель кафедры Информационно-измерительных систем
Рамазанов Э. Р.

Фонд оценочных средств по практике составлен в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика, утвержденного приказом Минобрнауки от 19.09.2017 № 922

Согласование и утверждение

№	Подразделение	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1	факультет Геоинформатики и информационной безопасности	Председатель ученого совета факультета	Матерухин А. В.	Рассмотрено	14.12.2023, № 7-23

1. Показатели и критерии оценивания компетенций

Планируемые результаты освоения программы	Уровень овладения	Критерии оценивания (перечень сформированных ИДК/ЗУВ по уровням освоения)
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Повышенный	ИДК ОПК-7.1 Анализирует поставленную задачу и требования к программному продукту, определяет необходимые алгоритмические и программные решения, а также оценивает их применимость для практического использования Знать: ОПК-7.1/Зн1 Основные парадигмы программирования и алгоритмические структуры, методы анализа и декомпозиции задач, критерии выбора алгоритмических решений, основы проектирования программных систем на языке Python, методы оценки вычислительной сложности алгоритмов и их практической применимости. Уметь: ОПК-7.1/Ум5 Проводить анализ исходных данных и условий задачи, формулировать требования к алгоритму и программной реализации, определять ограничения и оценивать применимость выбранного решения. ОПК-7.1/Ум6 Построить блок-схему заданного алгоритма, исходя из его описания Владеть: ОПК-7.1/Нв1 Навыками системного анализа предметной области и формализации требований, методами алгоритмического проектирования и выбора структур данных в Python, инструментами профилирования и оценки производительности Python-приложений, навыками технико-экономического обоснования выбранных программных решений. ОПК-7.1/Нв5 Навыками анализа поставленной задачи, включающим формализацию исходных данных в виде алгоритмической модели задачи, выбор рациональных методов решения и обоснование предпочтительной алгоритмической стратегии.

ИДК ОПК-7.2 Разрабатывает эффективные алгоритмы для решения задач, а также выбирает оптимальные структуры данных для их реализации

Знать:

ОПК-7.2/Зн1 Основные алгоритмические парадигмы (жадные алгоритмы, динамическое программирование, метод "разделяй и властвуй", рекурсивные алгоритмы), базовые и продвинутые структуры данных Python (списки, кортежи, множества, словари, очереди, стеки, деревья), методы анализа временной и пространственной сложности алгоритмов с использованием нотации O-большое, критерии выбора оптимальных структур данных для различных типов задач, классические алгоритмы сортировки, поиска, обхода графов и области их применения.

ОПК-7.2/Зн5 Классы алгоритмических стратегий; стандартные структуры данных (массив, список, дерево, граф, куча и др.); понятия сложности алгоритма.

Уметь:

ОПК-7.2/Ум1 Разрабатывать эффективные алгоритмические решения с учетом ограничений по времени и памяти, выбирать подходящие структуры данных на основе анализа требований задачи, оценивать и сравнивать эффективность различных алгоритмических подходов, реализовывать классические алгоритмы и адаптировать их под конкретные задачи, применять встроенные структуры данных Python и библиотеки collections для оптимизации решений.

ОПК-7.2/Ум2 Применять современные информационные технологии для анализа данных

Владеть:

ОПК-7.2/Нв1 Навыками разработки и реализации эффективных алгоритмов на языке Python, техниками оптимизации алгоритмов и выбора оптимальных структур данных, методами профилирования и анализа производительности алгоритмов, инструментами визуализации алгоритмов и структур данных.

ОПК-7.2/Нв2 Навыками анализа данных для решения задач профессиональной деятельности

ИДК ОПК-7.3 Пишет программный код на одном или нескольких языках программирования, следуя принципам модульности, читаемости, документирования и соответствия стандартам кодирования

Знать:

ОПК-7.3/Зн1 Синтаксис и семантику языка программирования Python, принципы структурного и модульного программирования, стандарты оформления кода PEP 8 и PEP 257, системы документирования кода (docstrings, type hints), принципы написания чистого и поддерживаемого кода, паттерны проектирования, применимые в Python.

Уметь:

ОПК-7.3/Ум1 Писать читаемый и эффективный код на языке Python с соблюдением принципов DRY и KISS, структурировать программу с использованием функций, классов и модулей, документировать код с использованием строк документации и комментариев, аннотацию типов для повышения надежности кода, использовать инструменты автоматической проверки качества кода (pylint, flake8, black).

ОПК-7.3/Ум5 Реализовывать алгоритмы в виде программных модулей, документировать код и соблюдать стандарты оформления

Владеть:

ОПК-7.3/Нв1 Навыками разработки модульных Python-приложений с четким разделением ответственности, инструментами рефакторинга и оптимизации кода, методами написания самодокументируемого кода, навыками работы с системами контроля версий для командной разработки.

ОПК-7.3/Нв4 Навыками пользования современными информационными технологиями с учетом их особенностей и возможной области применения

ОПК-7.3/Нв5 Навыками программирования и реализации алгоритмов на языках C++ и Python с соблюдением принципов читаемости и модульности

ИДК ОПК-7.4 Проводит отладку и тестирование программного кода, выявляет и устраняет ошибки, а также оптимизирует производительность программы для обеспечения её стабильной и эффективной работы в реальных условиях

Знать:

ОПК-7.4/Зн1 Виды ошибок в программах (синтаксические, логические, времени выполнения), методологии и стратегии отладки программного кода, основы модульного и интеграционного тестирования, инструменты отладки и профилирования в Python, принципы обработки исключений и написания отказоустойчивого кода.

Уметь:

ОПК-7.4/Ум1 Использовать встроенные и внешние отладчики для поиска и устранения ошибок, разрабатывать unit-тесты с использованием модулей unittest и pytest, применять методы логирования для диагностики проблем в работе программы, проводить профилирование кода и выявлять узкие места производительности, обрабатывать исключения и создавать информативные сообщения об ошибках.

Владеть:

ОПК-7.4/Нв1 Навыками систематической отладки и рефакторинга программного кода, техниками написания тестируемого кода и создания тестовых сценариев, инструментами автоматизированного тестирования и непрерывной интеграции, методами оптимизации производительности Python-приложений, навыками использования систем контроля версий для отслеживания и исправления ошибок.

Базовый	ИДК ОПК-7.1 Анализирует поставленную задачу и требования к программному продукту, определяет необходимые алгоритмические и программные решения, а также оценивает их применимость для практического использования Знать: ОПК-7.1/Зн1 Основные парадигмы программирования и алгоритмические структуры, методы анализа и декомпозиции задач, критерии выбора алгоритмических решений, основы проектирования программных систем на языке Python, методы оценки вычислительной сложности алгоритмов и их практической применимости. Уметь: ОПК-7.1/Ум5 Проводить анализ исходных данных и условий задачи, формулировать требования к алгоритму и программной реализации, определять ограничения и оценивать применимость выбранного решения. ОПК-7.1/Ум6 Построить блок-схему заданного алгоритма, исходя из его описания
Базовый	ИДК ОПК-7.2 Разрабатывает эффективные алгоритмы для решения задач, а также выбирает оптимальные структуры данных для их реализации Знать: ОПК-7.2/Зн1 Основные алгоритмические парадигмы (жадные алгоритмы, динамическое программирование, метод "разделяй и властвуй", рекурсивные алгоритмы), базовые и продвинутые структуры данных Python (списки, кортежи, множества, словари, очереди, стеки, деревья), методы анализа временной и пространственной сложности алгоритмов с использованием нотации O-большое, критерии выбора оптимальных структур данных для различных типов задач, классические алгоритмы сортировки, поиска, обхода графов и области их применения. ОПК-7.2/Зн5 Классы алгоритмических стратегий; стандартные структуры данных (массив, список, дерево, граф, куча и др.); понятия сложности алгоритма. Уметь: ОПК-7.2/Ум1 Разрабатывать эффективные алгоритмические решения с учетом ограничений по времени и памяти, выбирать подходящие структуры данных на основе анализа требований задачи, оценивать и сравнивать эффективность различных алгоритмических подходов, реализовывать классические алгоритмы и адаптировать их под конкретные задачи, применять встроенные структуры данных Python и библиотеки collections для оптимизации решений.

ОПК-7.2/Ум2 Применять современные информационные технологии для анализа данных

ИДК ОПК-7.3 Пишет программный код на одном или нескольких языках программирования, следуя принципам модульности, читаемости, документирования и соответствия стандартам кодирования

Знать:

ОПК-7.3/Зн1 Синтаксис и семантику языка программирования Python, принципы структурного и модульного программирования, стандарты оформления кода PEP 8 и PEP 257, системы документирования кода (docstrings, type hints), принципы написания чистого и поддерживаемого кода, паттерны проектирования, применимые в Python.

Уметь:

ОПК-7.3/Ум1 Писать читаемый и эффективный код на языке Python с соблюдением принципов DRY и KISS, структурировать программу с использованием функций, классов и модулей, документировать код с использованием строк документации и комментариев, аннотацию типов для повышения надежности кода, использовать инструменты автоматической проверки качества кода (pylint, flake8, black).

ОПК-7.3/Ум5 Реализовывать алгоритмы в виде программных модулей, документировать код и соблюдать стандарты оформления

ИДК ОПК-7.4 Проводит отладку и тестирование программного кода, выявляет и устраняет ошибки, а также оптимизирует производительность программы для обеспечения её стабильной и эффективной работы в реальных условиях

Знать:

ОПК-7.4/Зн1 Виды ошибок в программах (синтаксические, логические, времени выполнения), методологии и стратегии отладки программного кода, основы модульного и интеграционного тестирования, инструменты отладки и профилирования в Python, принципы обработки исключений и написания отказоустойчивого кода.

Уметь:

ОПК-7.4/Ум1 Использовать встроенные и внешние отладчики для поиска и устранения ошибок, разрабатывать unit-тесты с использованием модулей unittest и pytest, применять методы логирования для диагностики проблем в работе программы, проводить профилирование кода и выявлять узкие места производительности, обрабатывать исключения и создавать информативные сообщения об ошибках.

Пороговый	ИДК ОПК-7.1 Анализирует поставленную задачу и требования к программному продукту, определяет необходимые алгоритмические и программные решения, а также оценивает их применимость для практического использования Знать: ОПК-7.1/Зн1 Основные парадигмы программирования и алгоритмические структуры, методы анализа и декомпозиции задач, критерии выбора алгоритмических решений, основы проектирования программных систем на языке Python, методы оценки вычислительной сложности алгоритмов и их практической применимости.
Пороговый	ИДК ОПК-7.2 Разрабатывает эффективные алгоритмы для решения задач, а также выбирает оптимальные структуры данных для их реализации Знать: ОПК-7.2/Зн1 Основные алгоритмические парадигмы (жадные алгоритмы, динамическое программирование, метод "разделяй и властвуй", рекурсивные алгоритмы), базовые и продвинутые структуры данных Python (списки, кортежи, множества, словари, очереди, стеки, деревья), методы анализа временной и пространственной сложности алгоритмов с использованием нотации O-большое, критерии выбора оптимальных структур данных для различных типов задач, классические алгоритмы сортировки, поиска, обхода графов и области их применения. ОПК-7.2/Зн5 Классы алгоритмических стратегий; стандартные структуры данных (массив, список, дерево, граф, куча и др.); понятия сложности алгоритма. ИДК ОПК-7.3 Пишет программный код на одном или нескольких языках программирования, следуя принципам модульности, читаемости, документирования и соответствия стандартам кодирования Знать: ОПК-7.3/Зн1 Синтаксис и семантику языка программирования Python, принципы структурного и модульного программирования, стандарты оформления кода PEP 8 и PEP 257, системы документирования кода (docstrings, type hints), принципы написания чистого и поддерживаемого кода, паттерны проектирования, применимые в Python.

		ИДК ОПК-7.4 Проводит отладку и тестирование программного кода, выявляет и устраняет ошибки, а также оптимизирует производительность программы для обеспечения её стабильной и эффективной работы в реальных условиях Знать: ОПК-7.4/Зн1 Виды ошибок в программах (синтаксические, логические, времени выполнения), методологии и стратегии отладки программного кода, основы модульного и интеграционного тестирования, инструменты отладки и профилирования в Python, принципы обработки исключений и написания отказоустойчивого кода.
--	--	--

2. Шкала оценивания

Оценка	Критерии оценивания (перечень сформированных ИДК/ЗУВ)
Отлично	ОПК-7.1 Анализирует поставленную задачу и требования к программному продукту, определяет необходимые алгоритмические и программные решения, а также оценивает их применимость для практического использования ОПК-7.1/Зн1 Основные парадигмы программирования и алгоритмические структуры, методы анализа и декомпозиции задач, критерии выбора алгоритмических решений, основы проектирования программных систем на языке Python, методы оценки вычислительной сложности алгоритмов и их практической применимости. ОПК-7.1/Ум5 Проводить анализ исходных данных и условий задачи, формулировать требования к алгоритму и программной реализации, определять ограничения и оценивать применимость выбранного решения. ОПК-7.1/Ум6 Построить блок-схему заданного алгоритма, исходя из его описания ОПК-7.1/Нв1 Навыками системного анализа предметной области и формализации требований, методами алгоритмического проектирования и выбора структур данных в Python, инструментами профилирования и оценки производительности Python-приложений, навыками технико-экономического обоснования выбранных программных решений. ОПК-7.1/Нв5 Навыками анализа поставленной задачи, включающим формализацию исходных данных в виде алгоритмической модели задачи, выбор рациональных методов решения и обоснование предпочтительной алгоритмической стратегии.

ОПК-7.2 Разрабатывает эффективные алгоритмы для решения задач, а также выбирает оптимальные структуры данных для их реализации
ОПК-7.2/Зн1 Основные алгоритмические парадигмы (жадные алгоритмы, динамическое программирование, метод "разделяй и властвуй", рекурсивные алгоритмы), базовые и продвинутое структуры данных Python (списки, кортежи, множества, словари, очереди, стеки, деревья), методы анализа временной и пространственной сложности алгоритмов с использованием нотации O-большое, критерии выбора оптимальных структур данных для различных типов задач, классические алгоритмы сортировки, поиска, обхода графов и области их применения.

ОПК-7.2/Зн5 Классы алгоритмических стратегий; стандартные структуры данных (массив, список, дерево, граф, куча и др.); понятия сложности алгоритма.

ОПК-7.2/Ум1 Разрабатывать эффективные алгоритмические решения с учетом ограничений по времени и памяти, выбирать подходящие структуры данных на основе анализа требований задачи, оценивать и сравнивать эффективность различных алгоритмических подходов, реализовывать классические алгоритмы и адаптировать их под конкретные задачи, применять встроенные структуры данных Python и библиотеки collections для оптимизации решений.

ОПК-7.2/Ум2 Применять современные информационные технологии для анализа данных

ОПК-7.2/Нв1 Навыками разработки и реализации эффективных алгоритмов на языке Python, техниками оптимизации алгоритмов и выбора оптимальных структур данных, методами профилирования и анализа производительности алгоритмов, инструментами визуализации алгоритмов и структур данных.

ОПК-7.2/Нв2 Навыками анализа данных для решения задач профессиональной деятельности

ОПК-7.3 Пишет программный код на одном или нескольких языках программирования, следуя принципам модульности, читаемости, документирования и соответствия стандартам кодирования

ОПК-7.3/Зн1 Синтаксис и семантику языка программирования Python, принципы структурного и модульного программирования, стандарты оформления кода PEP 8 и PEP 257, системы документирования кода (docstrings, type hints), принципы написания чистого и поддерживаемого кода, паттерны проектирования, применимые в Python.

ОПК-7.3/Ум1 Писать читаемый и эффективный код на языке Python с соблюдением принципов DRY и KISS, структурировать программу с использованием функций, классов и модулей, документировать код с использованием строк документации и комментариев, аннотацию типов для повышения надежности кода, использовать инструменты автоматической проверки качества кода (pylint, flake8, black).

ОПК-7.3/Ум5 Реализовывать алгоритмы в виде программных модулей, документировать код и соблюдать стандарты оформления

ОПК-7.3/Нв1 Навыками разработки модульных Python-приложений с четким разделением ответственности, инструментами рефакторинга и оптимизации кода, методами написания самодокументируемого кода, навыками работы с системами контроля версий для командной разработки.

ОПК-7.3/Нв4 Навыками пользования современными информационными технологиями с учетом их особенностей и возможной области применения

ОПК-7.3/Нв5 Навыками программирования и реализации алгоритмов на языках C++ и Python с соблюдением принципов читаемости и модульности

	ОПК-7.4 Проводит отладку и тестирование программного кода, выявляет и устраняет ошибки, а также оптимизирует производительность программы для обеспечения её стабильной и эффективной работы в реальных условиях ОПК-7.4/Зн1 Виды ошибок в программах (синтаксические, логические, времени выполнения), методологии и стратегии отладки программного кода, основы модульного и интеграционного тестирования, инструменты отладки и профилирования в Python, принципы обработки исключений и написания отказоустойчивого кода. ОПК-7.4/Ум1 Использовать встроенные и внешние отладчики для поиска и устранения ошибок, разрабатывать unit-тесты с использованием модулей unittest и pytest, применять методы логирования для диагностики проблем в работе программы, проводить профилирование кода и выявлять узкие места производительности, обрабатывать исключения и создавать информативные сообщения об ошибках. ОПК-7.4/Нв1 Навыками систематической отладки и рефакторинга программного кода, техниками написания тестируемого кода и создания тестовых сценариев, инструментами автоматизированного тестирования и непрерывной интеграции, методами оптимизации производительности Python-приложений, навыками использования систем контроля версий для отслеживания и исправления ошибок.
Хорошо	ОПК-7.1 Анализирует поставленную задачу и требования к программному продукту, определяет необходимые алгоритмические и программные решения, а также оценивает их применимость для практического использования ОПК-7.1/Зн1 Основные парадигмы программирования и алгоритмические структуры, методы анализа и декомпозиции задач, критерии выбора алгоритмических решений, основы проектирования программных систем на языке Python, методы оценки вычислительной сложности алгоритмов и их практической применимости. ОПК-7.1/Ум5 Проводить анализ исходных данных и условий задачи, формулировать требования к алгоритму и программной реализации, определять ограничения и оценивать применимость выбранного решения. ОПК-7.1/Ум6 Построить блок-схему заданного алгоритма, исходя из его описания

ОПК-7.2 Разрабатывает эффективные алгоритмы для решения задач, а также выбирает оптимальные структуры данных для их реализации
ОПК-7.2/Зн1 Основные алгоритмические парадигмы (жадные алгоритмы, динамическое программирование, метод "разделяй и властвуй", рекурсивные алгоритмы), базовые и продвинутое структуры данных Python (списки, кортежи, множества, словари, очереди, стеки, деревья), методы анализа временной и пространственной сложности алгоритмов с использованием нотации O-большое, критерии выбора оптимальных структур данных для различных типов задач, классические алгоритмы сортировки, поиска, обхода графов и области их применения.

ОПК-7.2/Зн5 Классы алгоритмических стратегий; стандартные структуры данных (массив, список, дерево, граф, куча и др.); понятия сложности алгоритма.

ОПК-7.2/Ум1 Разрабатывать эффективные алгоритмические решения с учетом ограничений по времени и памяти, выбирать подходящие структуры данных на основе анализа требований задачи, оценивать и сравнивать эффективность различных алгоритмических подходов, реализовывать классические алгоритмы и адаптировать их под конкретные задачи, применять встроенные структуры данных Python и библиотеки collections для оптимизации решений.

ОПК-7.2/Ум2 Применять современные информационные технологии для анализа данных

ОПК-7.3 Пишет программный код на одном или нескольких языках программирования, следуя принципам модульности, читаемости, документирования и соответствия стандартам кодирования

ОПК-7.3/Зн1 Синтаксис и семантику языка программирования Python, принципы структурного и модульного программирования, стандарты оформления кода PEP 8 и PEP 257, системы документирования кода (docstrings, type hints), принципы написания чистого и поддерживаемого кода, паттерны проектирования, применимые в Python.

ОПК-7.3/Ум1 Писать читаемый и эффективный код на языке Python с соблюдением принципов DRY и KISS, структурировать программу с использованием функций, классов и модулей, документировать код с использованием строк документации и комментариев, аннотацию типов для повышения надежности кода, использовать инструменты автоматической проверки качества кода (pylint, flake8, black).

ОПК-7.3/Ум5 Реализовывать алгоритмы в виде программных модулей, документировать код и соблюдать стандарты оформления

ОПК-7.4 Проводит отладку и тестирование программного кода, выявляет и устраняет ошибки, а также оптимизирует производительность программы для обеспечения её стабильной и эффективной работы в реальных условиях

ОПК-7.4/Зн1 Виды ошибок в программах (синтаксические, логические, времени выполнения), методологии и стратегии отладки программного кода, основы модульного и интеграционного тестирования, инструменты отладки и профилирования в Python, принципы обработки исключений и написания отказоустойчивого кода.

ОПК-7.4/Ум1 Использовать встроенные и внешние отладчики для поиска и устранения ошибок, разрабатывать unit-тесты с использованием модулей unittest и pytest, применять методы логирования для диагностики проблем в работе программы, проводить профилирование кода и выявлять узкие места производительности, обрабатывать исключения и создавать информативные сообщения об ошибках.

Удовлетворительно	ОПК-7.1 Анализирует поставленную задачу и требования к программному продукту, определяет необходимые алгоритмические и программные решения, а также оценивает их применимость для практического использования ОПК-7.1/Зн1 Основные парадигмы программирования и алгоритмические структуры, методы анализа и декомпозиции задач, критерии выбора алгоритмических решений, основы проектирования программных систем на языке Python, методы оценки вычислительной сложности алгоритмов и их практической применимости. ОПК-7.2 Разрабатывает эффективные алгоритмы для решения задач, а также выбирает оптимальные структуры данных для их реализации ОПК-7.2/Зн1 Основные алгоритмические парадигмы (жадные алгоритмы, динамическое программирование, метод "разделяй и властвуй", рекурсивные алгоритмы), базовые и продвинутые структуры данных Python (списки, кортежи, множества, словари, очереди, стеки, деревья), методы анализа временной и пространственной сложности алгоритмов с использованием нотации O-большое, критерии выбора оптимальных структур данных для различных типов задач, классические алгоритмы сортировки, поиска, обхода графов и области их применения. ОПК-7.2/Зн5 Классы алгоритмических стратегий; стандартные структуры данных (массив, список, дерево, граф, куча и др.); понятия сложности алгоритма. ОПК-7.3 Пишет программный код на одном или нескольких языках программирования, следуя принципам модульности, читаемости, документирования и соответствия стандартам кодирования ОПК-7.3/Зн1 Синтаксис и семантику языка программирования Python, принципы структурного и модульного программирования, стандарты оформления кода PEP 8 и PEP 257, системы документирования кода (docstrings, type hints), принципы написания чистого и поддерживаемого кода, паттерны проектирования, применимые в Python. ОПК-7.4 Проводит отладку и тестирование программного кода, выявляет и устраняет ошибки, а также оптимизирует производительность программы для обеспечения её стабильной и эффективной работы в реальных условиях ОПК-7.4/Зн1 Виды ошибок в программах (синтаксические, логические, времени выполнения), методологии и стратегии отладки программного кода, основы модульного и интеграционного тестирования, инструменты отладки и профилирования в Python, принципы обработки исключений и написания отказоустойчивого кода.
Неудовлетворительно	Результаты обучения не сформированы на достаточном уровне

3. Контрольные мероприятия по практике

Вид контроля	Форма контроля/Оценочное средство
Текущий контроль	Тестирование
Промежуточная аттестация	Зачет с оценкой

№ п/п	Наименование раздела	Контролируемые ИДК	Вид контроля/ используемые оценочные материалы	
			Текущий	Промежут. аттестация

1	Подготовительный этап: Формирование проектных заданий - 20 час. Тема 1.1 Определение треков, стека и критериев оценки - 20 час.	ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4		Зачет с оценкой
2	Планирование и проектирование - 20 час. Тема 2.1 Консультации по планированию разработки и проектированию схемы взаимодействия компонентов системы - 10 час. Тема 2.2 Проектирование системы и распределение ролей - 10 час.	ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4	Тестирование	Зачет с оценкой
3	Разработка и реализация проекта - 130 час. Тема 3.1 Консультации по процессу разработки - 20 час. Тема 3.2 Практическая реализация системы - 110 час.	ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4		Зачет с оценкой
4	Завершающий этап: Подготовка к демонстрации и подведение итогов - 10 час. Тема 4.1 Формирование отчетных материалов - 10 час.	ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4		Зачет с оценкой

4. Порядок проведения промежуточной аттестации

Порядок проведения промежуточной аттестации регламентируется локальными нормативными актами Университета:

- Положением о формах, периодичности, порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по основным образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.6 «Организация и проведение промежуточной аттестации».
- Положением о порядке осуществления образовательного процесса, проведения текущего контроля успеваемости и промежуточной аттестации обучающихся с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.8 «Проведение промежуточной аттестации с применением ЭО и ДОТ».

Промежуточная аттестация: Зачет с оценкой, 2 семестр.

Зачет оценивается по пятибалльной шкале:

- зачено с оценкой «отлично» выставляется, если обучающийся имеет глубокие, исчерпывающие знания в объеме освоенной программы; знание основной (обязательной) литературы; дает правильные и уверенные ответы, свидетельствующие о наличии твердых знаний и навыков в использовании технических средств; полное, четкое, грамотное и логически стройное изложение материала; демонстрирует свободное применение теоретических знаний при анализе практических вопросов;
- зачено с оценкой «хорошо» выставляется обучающемуся в случае соблюдения тех же требований, но наличии в ответе обучающегося по некоторым перечисленным показателям

недостатков принципиального характера, что вызывает замечания или поправки преподавателя;

- зачтено с оценкой «удовлетворительно» выставляется обучающемуся в случае соблюдения тех же требований, но в случае наличия в ответе ошибок, необходимости существенной помощи в виде поправок и наводящих вопросов преподавателя;
- не зачтено («не удовлетворительно») – обучающийся не смог ответить на 2/3 вопросов; не справился с практическим заданием или выполнено менее 50 % практических заданий.

5. Оценочные материалы текущего контроля

Раздел 2. Планирование и проектирование

Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4

Тема 2.2. Проектирование системы и распределение ролей

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Что такое Minimum Viable Product?
2. Зачем нужно планировать разработку и разбивать ее на задачи?
3. Какой инструмент для управления задачами вы использовали в команде?
4. Что такое "декомпозиция" задачи?
5. Почему важно сначала реализовать MVP, а не весь функционал сразу?
6. Как вы в команде отслеживали прогресс выполнения задач?
7. С какими основными трудностями в планировании вы столкнулись?
8. Что такое бэклог проекта?
9. Почему важно определить четкие критерии готовности для каждой задачи?
10. Как вы определяли приоритеты задач при планировании?
11. Что такое схема взаимодействия компонентов системы?
12. Какие основные компоненты были в архитектуре вашего проекта?
13. Как Frontend-приложение взаимодействует с Backend-приложением?
14. Что такое REST API?
15. Что такое endpoint в контексте REST API?
16. Зачем нужна схема базы данных?
17. Как вы спроектировали API для основной функции вашего проекта?
18. Почему важно спроектировать API до начала его реализации?
19. В вашем стеке есть PostgreSQL и Redis. Для каких данных вы бы использовали PostgreSQL, а для каких — Redis?
20. Как вы организовали взаимодействие контейнеров в docker-compose.yml?
21. Какие две основные роли были у разработчиков в вашей команде?
22. Какова была ваша роль в команде?
23. Опишите основную зону ответственности Backend-разработчика.
24. Опишите основную зону ответственности Frontend-разработчика.
25. Как вы взаимодействовали с другим разработчиком в вашей команде?
26. Какую роль в команде выполняли специалисты по ИБ?
27. Приведите пример задачи, которую вам поставил специалист по ИБ.
28. Как вы использовали Git для совместной работы?
29. Сталкивались ли вы с конфликтами слияния в Git и как вы их решали?
30. Почему важно, чтобы каждый член команды четко понимал свою роль и зону ответственности?

6. Оценочные материалы промежуточной аттестации

2 семестр, Зачет с оценкой

Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4

Вопросы/Задания:

1. Какую информацию должна содержать схема взаимодействия компонентов системы и какая цель проектирования такой схемы?
2. Опишите ключевые различия в задачах Backend- и Frontend-разработчика в рамках одного проекта. Как они договариваются об API?
3. Почему документирование REST API является полезным в рамках любого проекта?
4. Объясните, для каких типов данных в рамках одного проекта могут использоваться PostgreSQL и Redis.
5. Какую основную проблему решает использование docker-compose для локальной разработки и тестирования многокомпонентного приложения?
6. Опишите процесс работы с Git для команды из двух разработчиков.
7. Какой алгоритм вы использовали для генерации уникальной короткой ссылки и как бы решали проблему коллизий?
8. Как бы вы спроектировали схему базы данных в PostgreSQL для эффективного хранения информации о переходах для последующей аналитики?
9. Какую основную задачу, помимо сокращения URL, решает этот сервис?
10. Зачем в этом сервисе нужен Rate limiting?
11. Зачем в этом проекте используется микросервисная архитектура?
12. Какую задачу решает компонент "API Gateway"?
13. Что означает требование "End-to-end шифрование" и от какой угрозы оно защищает?
14. Зачем для видеохостинга нужна "система транскодинга в различные форматы"?
15. Какую роль выполняет CDN в видеохостинге?
16. Что такое Центр сертификации и какова его главная функция в системе управления сертификатами?
17. Зачем в этой системе нужна интеграция с HSM?
18. Почему "изоляция выполнения заданий" является критически важным требованием безопасности для сервиса проверки домашних заданий?

19. Какую информацию должен шифровать сервис записи к врачам в соответствии с требованиями безопасности?

20. Какой основной компонент этой системы курьерской доставки позволяет клиентам отслеживать свои заказы?

21. Какие данные в этом проекте для курьерской доставки требуют особой защиты согласно требованиям безопасности?

22. Как система управления криптовалютным кошельком получает историю транзакций и балансы? Опишите плюсы и минусы хранения этой информации в локальной базе данных PostgreSQL по сравнению с получением ее напрямую из блокчейна при каждом запросе.

23. Какова роль backend-сервера в системе управления криптовалютного кошелька, если подписание транзакций происходит на стороне клиента (в браузере)? Какие задачи, кроме простого отображения страниц, он выполняет?

24. Зачем в сервисе мониторинга банковских транзакций используется технология потоковой обработки, такая как Apache Kafka?

25. В чем разница между статическим и динамическим анализом?

26. Почему важно проводить анализ мобильного приложения в "изолированной среде выполнения"?

27. Какие два основных типа сканирования выполняет система сканирования инфраструктуры на уязвимости?

28. Для чего нужна "Система приоритизации уязвимостей (CVSS)"?

29. Какую основную задачу решает DLP-система?

30. Что происходит с файлами, которые система классификации и защиты конфиденциальных данных определяет как "подозрительные"?

31. В чем главное преимущество "централизованной системы управления" межсетевыми экранами по сравнению с ручной настройкой каждого устройства?

32. Зачем в системе управления межсетевыми экранами нужен "аудит всех изменений" правил?

33. Что такое MVP и какова его основная цель?

34. На этапе проектирования системы какую роль выполняет разработчик?

35. Какая технология является ключевой для передачи видео и аудио между браузерами в этом проекте?