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
Тюменский государственный университет

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5BFEE1D286E9C95A0C0D2C611E434BD5
Владелец: Степанченко Алексей Леонидович
Действителен: с 23.09.2026 по 17.12.2027

«УТВЕРЖДАЮ»
Проректор
А. Л. Степанченко
24 марта 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) «Б1.О.09 ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

Уровень высшего образования: магистратура

Направление подготовки: 09.04.03 Прикладная информатика

Направленность (профиль) подготовки: Сквозные цифровые технологии в управлении недвижимостью

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Год набора (приема на обучение): 2025

Срок получения образования: 1 года 10 месяца(-ев)

Объем: в зачетных единицах: 9 з.е.
в академических часах: 324 ак.ч.

Москва, 2025

Разработчики:
Профессор академического департамента, доктор
физико-математических наук Шевляков А. Н.

Фонд оценочных средств по дисциплине (модулю) составлен в соответствии с требованиями ФГОС ВО по направлению подготовки 09.04.03 Прикладная информатика, утвержденного приказом Минобрнауки от 19.09.2017 № 916

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	14.10.2024, № 10-24

1. Планируемые результаты обучения, соотнесенные с планируемыми результатами освоения образовательной программы

ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач

ОПК-2.1 Умеет разрабатывать оригинальные алгоритмы и программные средства с использованием современных интеллектуальных технологий

Знать:

ОПК-2.1/Зн1 Современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач

ОПК-2.1/Зн2 Методы разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач

ОПК-2.1/Зн3 Архитектурные принципы построения систем искусственного интеллекта, методы декомпозиции основных подсистем (компонентов) и реализации их взаимодействия

Уметь:

ОПК-2.1/Ум1 Обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач

ОПК-2.1/Ум2 Разрабатывать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач

Владеть:

ОПК-2.1/Нв1 Навыками использования инструментов и методов разработки программных компонентов систем, основанных на знаниях

ОПК-2.1/Нв2 Навыками разработки систем искусственного интеллекта и их подсистем на основе различных архитектурных принципов

ОПК-2.2 Применяет расширенный математический аппарат при построении и обосновании оригинальных алгоритмов

Знать:

ОПК-2.2/Зн1 Методы построения программных алгоритмов

Уметь:

ОПК-2.2/Ум1 Применять расширенный математический аппарат при построении и обосновании оригинальных алгоритмов

Владеть:

ОПК-2.2/Нв1 Навыками обоснования оригинальных программных решений и алгоритмов

ОПК-2.3 Разрабатывает программные средства на основе методов интеллектуального анализа данных для решения профессиональных задач

Знать:

ОПК-2.3/Зн1 Методы интеллектуального анализа данных для решения профессиональных задач

Уметь:

ОПК-2.3/Ум1 Разрабатывать программные средства на основе интеллектуального анализа данных

Владеть:
ОПК-2.3/Нв1 Навыками разработки программных средств с использованием системного подхода

2. Показатели и критерии оценивания компетенций

Планируемые результаты освоения программы	Уровень овладения	Критерии оценивания (перечень сформированных ИДК/ЗУВ по уровням освоения)
ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	Повышенный	ИДК ОПК-2.1 Умеет разрабатывать оригинальные алгоритмы и программные средства с использованием современных интеллектуальных технологий Знать: ОПК-2.1/Зн1 Современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач ОПК-2.1/Зн2 Методы разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач ОПК-2.1/Зн3 Архитектурные принципы построения систем искусственного интеллекта, методы декомпозиции основных подсистем (компонентов) и реализации их взаимодействия Уметь: ОПК-2.1/Ум1 Обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач ОПК-2.1/Ум2 Разрабатывать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач Владеть: ОПК-2.1/Нв1 Навыками использования инструментов и методов разработки программных компонентов систем, основанных на знаниях ОПК-2.1/Нв2 Навыками разработки систем искусственного интеллекта и их подсистем на основе различных архитектурных принципов

	Базовый	ИДК ОПК-2.1 Умеет разрабатывать оригинальные алгоритмы и программные средства с использованием современных интеллектуальных технологий Знать: ОПК-2.1/Зн3 Архитектурные принципы построения систем искусственного интеллекта, методы декомпозиции основных подсистем (компонентов) и реализации их взаимодействия ОПК-2.1/Зн2 Методы разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач ОПК-2.1/Зн1 Современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач Уметь: ОПК-2.1/Ум2 Разрабатывать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач ОПК-2.1/Ум1 Обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач
	Пороговый	ИДК ОПК-2.1 Умеет разрабатывать оригинальные алгоритмы и программные средства с использованием современных интеллектуальных технологий Знать: ОПК-2.1/Зн1 Современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач ОПК-2.1/Зн2 Методы разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач ОПК-2.1/Зн3 Архитектурные принципы построения систем искусственного интеллекта, методы декомпозиции основных подсистем (компонентов) и реализации их взаимодействия

3. Шкала оценивания

Оценка	Критерии оценивания (перечень сформированных ИДК/ЗУВ)

Отлично	ОПК-2.1 Умеет разрабатывать оригинальные алгоритмы и программные средства с использованием современных интеллектуальных технологий ОПК-2.1/Зн1 Современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач ОПК-2.1/Зн2 Методы разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач ОПК-2.1/Зн3 Архитектурные принципы построения систем искусственного интеллекта, методы декомпозиции основных подсистем (компонентов) и реализации их взаимодействия ОПК-2.1/Ум1 Обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач ОПК-2.1/Ум2 Разрабатывать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач ОПК-2.1/Нв1 Навыками использования инструментов и методов разработки программных компонентов систем, основанных на знаниях ОПК-2.1/Нв2 Навыками разработки систем искусственного интеллекта и их подсистем на основе различных архитектурных принципов
Хорошо	ОПК-2.1 Умеет разрабатывать оригинальные алгоритмы и программные средства с использованием современных интеллектуальных технологий ОПК-2.1/Зн1 Современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач ОПК-2.1/Зн2 Методы разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач ОПК-2.1/Зн3 Архитектурные принципы построения систем искусственного интеллекта, методы декомпозиции основных подсистем (компонентов) и реализации их взаимодействия ОПК-2.1/Ум1 Обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач ОПК-2.1/Ум2 Разрабатывать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач
Удовлетворительно	ОПК-2.1 Умеет разрабатывать оригинальные алгоритмы и программные средства с использованием современных интеллектуальных технологий ОПК-2.1/Зн1 Современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач ОПК-2.1/Зн2 Методы разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач ОПК-2.1/Зн3 Архитектурные принципы построения систем искусственного интеллекта, методы декомпозиции основных подсистем (компонентов) и реализации их взаимодействия

Неудовле-творительно	Результаты обучения не сформированы на достаточном уровне

4. Контрольные мероприятия по дисциплине

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	Введение в программную инженерию. Жизненный цикл программного продукта	ОПК-2.1 ОПК-2.2 ОПК-2.3	Тестирование	Зачет с оценкой
2	Разработка приложений и систем машинного обучения	ОПК-2.1 ОПК-2.2 ОПК-2.3	Тестирование	Зачет с оценкой
3	Системы контроля версий и командная разработка	ОПК-2.1 ОПК-2.2 ОПК-2.3	Тестирование	Зачет с оценкой
4	Основы Continuous Integration (CI)	ОПК-2.1 ОПК-2.2 ОПК-2.3	Тестирование	Зачет с оценкой
5	Архитектура программного обеспечения. Разработка API для систем машинного обучения	ОПК-2.1 ОПК-2.2 ОПК-2.3	Тестирование	Зачет с оценкой
6	Переиспользование программного кода	ОПК-2.1 ОПК-2.2 ОПК-2.3	Тестирование	Зачет с оценкой
7	Стиль и качество кода. Рецензирование кода (Code Review)	ОПК-2.1 ОПК-2.2 ОПК-2.3	Тестирование	Зачет с оценкой
8	Тестирование систем машинного обучения	ОПК-2.1 ОПК-2.2 ОПК-2.3	Тестирование	Зачет с оценкой

5. 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:
- зачтено 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;
- зачтено 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.

6. Оценочные материалы текущего контроля

Раздел 1. Введение в программную инженерию. Жизненный цикл программного продукта
Контролируемые ИДК: ОПК-2.1 ОПК-2.2 ОПК-2.3

Тема 1.1. Модели жизненного цикла и методологии разработки ПО
Форма контроля/оценочное средство: Тестирование
Вопросы/Задания:
1. Как называется модель разработки с последовательными этапами?
2. Как называется семейство гибких методологий разработки?
3. Как называется модель жизненного цикла, ориентированная на управление рисками?

Раздел 2. Разработка приложений и систем машинного обучения
Контролируемые ИДК: ОПК-2.1 ОПК-2.2 ОПК-2.3

Тема 2.1. Особенности инженерии ML-проектов (MLOps)
Форма контроля/оценочное средство: Тестирование
Вопросы/Задания:
1. Как называется практика объединения разработки и эксплуатации ML-систем?

2. Какой инструмент используется для версионирования данных в ML-проектах?
3. Какой инструмент используется для отслеживания экспериментов в ML?
4. Как называется процесс получения предсказаний от обученной модели?

Раздел 3. Системы контроля версий и командная разработка
Контролируемые ИДК: ОПК-2.1 ОПК-2.2 ОПК-2.3

Тема 3.1. Коллективная работа с Git и платформенные решения (GitHub/GitLab)
Форма контроля/оценочное средство: Тестирование
Вопросы/Задания:
1. Какая система контроля версий является наиболее популярной?
2. Как сокращенно называют запрос на слияние изменений в Git?
3. Как называется стратегия ветвления Git с долгоживущими ветками main и develop?

Раздел 4. Основы Continuous Integration (CI)
Контролируемые ИДК: ОПК-2.1 ОПК-2.2 ОПК-2.3

Тема 4.1. Автоматизация сборки и проверки кода с помощью CI-серверов
Форма контроля/оценочное средство: Тестирование
Вопросы/Задания:
1. Как называется процесс автоматической сборки и тестирования при каждом изменении кода?
2. Как называется этап проверки стиля кода в CI-пайплайне?
3. Назовите один из самых популярных CI-серверов.

Раздел 5. Архитектура программного обеспечения. Разработка API для систем машинного обучения
Контролируемые ИДК: ОПК-2.1 ОПК-2.2 ОПК-2.3

Тема 5.1. Проектирование архитектуры и способы поставки ML-моделей через API
Форма контроля/оценочное средство: Тестирование
Вопросы/Задания:
1. Какой архитектурный стиль чаще всего используется для создания API?
2. Какой формат данных является стандартом для REST API?
3. Как называется архитектурный стиль, предполагающий разбиение приложения на мелкие независимые сервисы?

Раздел 6. Переиспользование программного кода
Контролируемые ИДК: ОПК-2.1 ОПК-2.2 ОПК-2.3

Тема 6.1. Пакетирование, менеджеры зависимостей и повторное использование кода
Форма контроля/оценочное средство: Тестирование
Вопросы/Задания:
1. Как называется принцип программирования "не повторяйся"?
2. Как называется стандартный менеджер пакетов для Python?
3. Как называется стандарт версионирования ПО в формате Major.Minor.Patch?
4. Как называется инструмент для создания изолированных окружений Python?

Раздел 7. Стиль и качество кода. Рецензирование кода (Code Review)
Контролируемые ИДК: ОПК-2.1 ОПК-2.2 ОПК-2.3

Тема 7.1. Стандарты оформления кода и практика код-ревью

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Как называется процесс рецензирования кода коллегами?
2. Как называется инструмент статического анализа кода для проверки стиля?
3. Как называется накопление проблем в коде, вызванное компромиссными решениями?
4. Как называется стандарт оформления кода на Python?

Раздел 8. Тестирование систем машинного обучения

Контролируемые ИДК: ОПК-2.1 ОПК-2.2 ОПК-2.3

Тема 8.1. Многоуровневое тестирование в ML-проектах

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Как называются тесты, проверяющие работу отдельного модуля или функции?
2. Как называется явление изменения статистических характеристик входных данных модели с течением времени?
3. Как называется метрика, показывающая долю истинно положительных результатов среди всех положительных предсказаний?
4. Как называется набор данных, используемый для финальной оценки качества модели?
5. Как называется метрика, показывающая долю найденных истинно положительных объектов?
6. Как называется процесс развертывания модели в продуктивную среду?

7. Оценочные материалы промежуточной аттестации

2 семестр, Зачет с оценкой

Контролируемые ИДК: ОПК-2.1 ОПК-2.2 ОПК-2.3

Вопросы/Задания:

1. Дайте определение понятию «программная инженерия». Чем она отличается от программирования?

2. Перечислите и кратко охарактеризуйте основные этапы жизненного цикла программного обеспечения.

3. В чем заключаются ключевые различия между каскадной (Waterfall) и итеративной моделями разработки?

4. Опишите основные принципы и ценности гибких методологий разработки (Agile).

5. В чем заключается специфика жизненного цикла систем машинного обучения по сравнению с традиционным ПО?

6. Какие дополнительные артефакты (помимо кода) появляются в ML-проектах и почему они требуют управления версиями?

7. Что такое MLOps? Какие проблемы призвана решать эта практика?

8. Опишите типовой пайплайн разработки системы машинного обучения (от постановки задачи до эксплуатации).

9. Объясните принцип работы распределенных систем контроля версий на примере Git. В чем их преимущество перед централизованными?

10. Для чего в Git используются ветки (branches)? Опишите одну из популярных стратегий ветвления (Git Flow или GitHub Flow).

11. Что такое pull request (merge request)? Опишите его жизненный цикл и роль в командной разработке.

12. Как разрешить конфликт слияния (merge conflict) в Git? Опишите последовательность действий разработчика.

13. Дайте определение непрерывной интеграции (Continuous Integration). Какие практики она в себя включает?

14. Какие проблемы в разработке решает внедрение CI и какие риски снижает?

15. Опишите типовую структуру CI-пайплайна. Какие этапы являются обязательными?

16. Какую роль выполняют автоматические тесты и статические анализаторы кода в процессе непрерывной интеграции?

17. Что такое архитектура программного обеспечения? Назовите основные архитектурные стили, которые вы знаете.

18. В чем заключается идея микросервисной архитектуры? Каковы ее преимущества и недостатки?

19. Для чего при разработке систем машинного обучения используется REST API?

20. Опишите структуру типового запроса и ответа при обращении к API модели машинного обучения.

21. Раскройте смысл принципа DRY (Don't Repeat Yourself). Как он связан с модульностью и переиспользованием кода?

22. Для чего нужны менеджеры зависимостей (например, pip, poetry) и какую проблему они решают?

23. Что такое семантическое версионирование (SemVer)? Как по номеру версии определить характер изменений?

24. Какие существуют способы создания изолированных окружений для Python-проектов и зачем это нужно?

25. Дайте определение понятию «технический долг» (technical debt). Какие практики помогают его минимизировать?

26. С какой целью используются стандарты оформления кода (например, PEP 8) и инструменты их проверки (линтеры)?

27. Опишите процесс Code Review. На что обращает внимание ревьювер при проверке кода?

28. Каковы правила конструктивного общения при проведении Code Review (как писать комментарии и как на них реагировать)?

29. Чем тестирование систем машинного обучения отличается от тестирования традиционного ПО? Какие новые уровни проверки появляются?

30. Что такое дрейф данных (data drift) и дрейф концепции (concept drift)? Почему их мониторинг критически важен для ML-систем в продуктивной среде?