

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Максимов Алексей Борисович

Должность: директор департамента по образовательной политике

Дата подписания: 05.09.2026 14:03:59

Уникальный программный ключ:

8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)**

Факультет информационных технологий

УТВЕРЖДАЮ

Декан факультета

«Информационных технологий»

 / Д.Г. Демидов /
« 26 » февраля 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Разработка приложений с использованием искусственного интеллекта»

Направление подготовки

10.03.01 «Информационная безопасность»

Профиль

«Безопасность компьютерных систем»

Квалификация

Бакалавр

Формы обучения

очная

Москва, 2026 г.

Разработчик:

Заведующий кафедрой «Информационная безопасность»,
к.т.н., доцент
И.В.Калуцкий



Согласовано:

Заведующий кафедрой «Информационная безопасность»,
к.т.н., доцент
И.В.Калуцкий



Руководитель образовательной программы
Гневшев



А.Ю.

Содержание

1. Цели освоения дисциплины	4
2. Место дисциплины в структуре ООП специалитета	5
3. Структура и содержание дисциплины	6
3.1 Виды учебной работы и трудоемкость	6
3.2 Тематический план изучения дисциплины	7
3.3 Содержание дисциплины	7
3.4 Тематика лабораторных занятий	9
4. Учебно-методическое обеспечение	10
5. Материально-техническое обеспечение	11
6. Методические рекомендации	12
7. Фонд оценочных средств	13

1. Цели освоения дисциплины.

Цель освоения дисциплины «Разработка приложений с использованием искусственного интеллекта»:

Формирование у студентов знаний, умений и навыков проектирования, разработки, тестирования и эксплуатации приложений, использующих методы и технологии искусственного интеллекта, включая машинное обучение, генеративные модели, большие языковые модели и интеллектуальные программные сервисы.

Задачи дисциплины

- изучение архитектур приложений с ИИ-компонентами;
- освоение методов подготовки данных, обучения и оценки моделей;
- изучение способов интеграции ML-моделей и AI API в программные системы;
- освоение технологий LLM, эмбедингов, векторного поиска и RAG;
- получение навыков создания AI-сервисов на Python;
- изучение вопросов тестирования, мониторинга и эксплуатации ИИ-приложений;
- изучение угроз безопасности, связанных с использованием генеративного ИИ;
- разработка и защита итогового проекта.

Планируемые результаты обучения

В результате освоения дисциплины студент должен:

Знать:

- основные классы моделей искусственного интеллекта;
- архитектуру типовых AI-приложений;
- принципы работы машинного обучения и больших языковых моделей;
- методы подготовки данных и оценки качества моделей;
- основные угрозы безопасности приложений с ИИ.

Уметь:

- формулировать задачу разработки интеллектуального приложения;
- подготавливать данные для обучения и тестирования;
- выбирать подходящую модель и метрики качества;
- интегрировать ML-модели и внешние AI API;
- разрабатывать RAG-приложения;

- проводить функциональное и нагрузочное тестирование AI-сервисов;
- выявлять риски утечки данных, prompt injection и небезопасного вывода модели.

Владеть:

- Python-инструментами разработки ИИ-приложений;
- технологиями REST API;
- средствами контейнеризации Docker;
- технологиями векторного поиска и базами данных;
- методами журналирования, мониторинга и контроля качества моделей;
- методами обеспечения безопасности AI-приложений.

Формируемые компетенции

Код компетенции	Индикаторы достижения
ОПК-2. Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности;	ИОПК-2.1 Определяет риски применения ИИ и методы контроля; ИОПК-2.2 Проверяет качество, целостность и воспроизводимость данных; ИОПК-2.3 Оценивает защищённость AI-приложения и корректность его ответов.
ОПК-7. Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности;	ИОПК-7.1 Разрабатывает архитектуру приложения с ИИ-компонентами с учётом требований к объекту защиты; ИОПК-7.2 Интегрирует модели и AI-сервисы в программные системы; ИОПК-7.3 Проводит опытную эксплуатацию и документирует результаты испытаний.

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина «Разработка приложений с использованием искусственного интеллекта» относится к числу профессиональных учебных дисциплин обязательной части цикла (Б1.2) основной образовательной программы бакалавриата (Б1.2.5).

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ООП: «Основы проектирования информационных систем»,

«Программно-аппаратные средства обеспечения информационной безопасности», «Методы машинного обучения и технологии искусственного интеллекта», «Объектно-ориентированное программирование», «Безопасность систем баз данных», «Архитектура облачных приложений и систем», «Инструментальные средства технологий искусственного интеллекта».

3. Структура и содержание дисциплины.

Общая трудоемкость дисциплины составляет **2** зачетных единицы, т.е. **72** академических часа: из них **36** часов аудиторные занятия (лекции – **8** часов, лабораторные занятия – **28** часов) и самостоятельная работа - **36** часов, форма контроля – дифференцированный зачет в 8 семестре.

3.1 Виды учебной работы и трудоемкость

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			Семестр 8	Неделя семестра
1	Аудиторные занятия	72	72	
	В том числе:			
1.1	Лекции	8	8	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	28	28	
2	Самостоятельная работа	36	36	
3	Промежуточная аттестация	Дифференцированный зачет	Дифференцированный зачет	
	Итого:	72	72	

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/ п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самос тояте льная работ а
			Лек ции	Семинар ские/ практиче ские занятия	Лабор аторн ые заняти я	Практиче ская подгот овка	
1	Введение в разработку AI-приложений	12	2		4		6
2	Данные и модели машинного обучения	10	2		4		4
3	Генеративный ИИ и большие языковые модели	10	2		4		4
4	Архитектура интеллектуальных приложений и RAG	10	2		4		4
5	Интеграция, тестирование и эксплуатация AI-сервисов	10			6		4
6	Безопасность и надёжность приложений с ИИ	10			6		6
7	Итоговый проект	10			6		8
Итого		72	8		28		36

3.3 Содержание дисциплины

Тема 1. Введение в разработку AI-приложений

Понятие интеллектуального приложения. Отличия классического приложения от приложения с ИИ-компонентом. Типовая архитектура AI-приложения: пользовательский интерфейс, прикладная логика, модель, хранилище данных, API и средства мониторинга.

Обзор Python, Jupyter, REST API, Docker и библиотек машинного обучения.

Тема 2. Данные и модели машинного обучения

Типы данных и задачи классификации, регрессии, кластеризации и поиска аномалий. Подготовка и разметка данных. Очистка данных, обработка пропусков, формирование признаков.

Разделение данных на обучающую, валидационную и тестовую выборки. Переобучение. Метрики качества. Сериализация модели и подключение модели к приложению.

Тема 3. Генеративный ИИ и большие языковые модели

Понятие генеративного искусственного интеллекта. Принципы работы больших языковых моделей. Prompt engineering, системные инструкции, структурированный вывод и function calling.

Ограничения контекстного окна, галлюцинации, неполнота знаний модели и зависимость качества результата от входных данных. Подключение локальных и облачных моделей через API.

Тема 4. Архитектура интеллектуальных приложений и RAG

Эмбединги, семантический поиск и векторные базы данных. Предварительная обработка документов, разбиение текста на фрагменты, индексация и поиск контекста.

Архитектура Retrieval-Augmented Generation. Формирование промпта с найденными источниками. Контроль ссылок и обоснованности ответа. Основы разработки AI-агентов и вызова внешних инструментов.

Тема 5. Интеграция, тестирование и эксплуатация AI-сервисов

Разработка REST-сервиса на Python. Конфигурация приложения, управление секретами, обработка ошибок, логирование, кэширование и асинхронные вызовы.

Функциональное и интеграционное тестирование. Оценка качества ответов модели. Мониторинг задержки, стоимости запросов, нагрузки и деградации качества. Основы MLOps и LLMOps.

Тема 6. Безопасность и надёжность приложений с ИИ

Основные угрозы:

- prompt injection;
- jailbreak;
- утечка конфиденциальных данных;
- отравление обучающих данных;
- небезопасный вывод модели;
- несанкционированный вызов инструментов;
- злоупотребление API;
- нарушение целостности базы знаний.

Методы защиты:

- разграничение доступа;
- маскирование персональных данных;
- управление секретами;

- фильтрация входных и выходных данных;
- аудит действий;
- проверка ответа модели;
- ограничение инструментов;
- использование механизма human-in-the-loop.

Тема 7. Итоговый проект

Разработка прототипа приложения с использованием искусственного интеллекта.

Проект должен включать:

- постановку задачи;
- описание целевой аудитории;
- архитектурную схему;
- описание используемых данных;
- выбор модели или AI API;
- программную реализацию;
- тестирование;
- оценку качества;
- анализ рисков безопасности;
- инструкцию по развёртыванию;
- презентацию и защиту.

3.4 Тематика лабораторных занятий

№	Наименование лабораторной работы	Часы
1	Подготовка среды разработки и создание базового AI-сервиса	2
2	Подготовка набора данных и обучение модели классификации	4
3	Интеграция ML-модели в REST API на Python	4
4	Работа с LLM через API: промпты и структурированный вывод	4

№	Наименование лабораторной работы	Часы
5	Создание RAG-приложения с эмбедами и векторным хранилищем	6
6	Тестирование и оценка качества ответов AI-приложения	2
7	Контейнеризация и развёртывание AI-сервиса	4
8	Аудит безопасности AI-приложения	2
Итого		28

Возможные темы итоговых проектов

- интеллектуальный помощник по внутренней нормативной документации;
- RAG-сервис для поиска по технической документации;
- система классификации обращений пользователей;
- сервис обнаружения аномалий в журналах информационной системы;
- интеллектуальный анализ событий информационной безопасности;
- чат-бот службы технической поддержки;
- AI-сервис анализа конфигураций облачной инфраструктуры;
- система классификации фишинговых сообщений;
- интеллектуальный помощник специалиста SOC;
- приложение поиска уязвимостей в исходном коде.

4. Учебно-методическое обеспечение

4.1 Основная литература

1. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение.
2. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python.
3. Документация Python, FastAPI, scikit-learn, PyTorch и используемого поставщика AI API.
4. Материалы курса МГУ «Разработка приложений с использованием искусственного интеллекта», выбранного как основной аналог по тематике дисциплины.

5. Материалы курса ИРНИТУ «Программирование искусственного интеллекта в приложениях», использованного как дополнительный ориентир по прикладным моделям ИИ.

4.2 Дополнительные электронные ресурсы

- Python Documentation — <https://docs.python.org/3/>
- FastAPI Documentation — <https://fastapi.tiangolo.com/>
- scikit-learn User Guide — https://scikit-learn.org/stable/user_guide.html
- PyTorch Documentation — <https://pytorch.org/docs/>
- Hugging Face Documentation — <https://huggingface.co/docs>
- LangChain Documentation — <https://python.langchain.com/docs>
- PostgreSQL Documentation — <https://www.postgresql.org/docs/>
- pgvector — <https://github.com/pgvector/pgvector>
- Yandex Cloud Documentation — <https://cloud.yandex.ru/docs>
- OWASP Top 10 for Large Language Model Applications — <https://owasp.org/www-project-top-10-for-large-language-model-applications/>

5. Материально-техническое обеспечение

Для проведения лекций необходимы:

- мультимедийный проектор;
- экран или интерактивная панель;
- компьютер преподавателя;
- доступ к сети Интернет.

Для лабораторных занятий необходим компьютерный класс из расчёта одно рабочее место на одного студента.

Рекомендуемое программное обеспечение

- Windows или Linux;
- Python 3.11 и выше;
- JupyterLab;
- Git;
- Visual Studio Code;
- Docker;
- PostgreSQL;

- расширение pgvector;
- pandas;
- NumPy;
- scikit-learn;
- FastAPI;
- transformers;
- sentence-transformers.

Для практических работ допускается использование локальных моделей, облачных AI API и совместимых отечественных сервисов.

6. Методические рекомендации

6.1. Для преподавателя

Обучение рекомендуется организовать по принципу:

1. краткое теоретическое объяснение;
2. демонстрация рабочего примера;
3. выполнение лабораторного задания;
4. защита результата;
5. обсуждение ошибок и ограничений.

При оценке необходимо учитывать не только факт запуска приложения, но и:

- обоснованность архитектуры;
- корректность выбора модели;
- качество исходных данных;
- воспроизводимость эксперимента;
- наличие тестов;
- безопасность обработки информации;
- полноту технической документации.

6.2. Для обучающихся

Самостоятельная работа включает:

- изучение учебной литературы и документации;
- подготовку наборов данных;
- выполнение лабораторных заданий;

- анализ качества модели;
- разработку итогового проекта.

Все результаты рекомендуется хранить в системе контроля версий. Репозиторий проекта должен содержать файл с зависимостями, инструкцию запуска, описание API, тестовые данные, параметры модели, результаты тестирования и перечень известных ограничений.

7. Фонд оценочных средств

7.1. Методы контроля

- защита лабораторных работ — **40%**;
- тестирование и контрольные вопросы — **20%**;
- итоговый проект и его защита — **40%**.

7.2. Критерии оценивания

Оценка	Критерии
Отлично	Все обязательные работы выполнены. Приложение работоспособно, архитектура обоснована, проведены тестирование и анализ безопасности. Студент понимает ограничения применяемых моделей.
Хорошо	Основные работы выполнены, приложение функционирует. Имеются небольшие недостатки в тестировании, документации или обосновании архитектуры.
Удовлетворительно	Освоены базовые понятия, выполнена часть практических заданий. Приложение работает ограниченно, имеются существенные недочёты.
Неудовлетворительно	Не выполнены обязательные лабораторные работы или отсутствуют базовые знания и работоспособный проект.

7.3. Примерные вопросы к дифференцированному зачёту

1. Назовите основные компоненты архитектуры приложения с ИИ.
2. Чем отличаются обучающая, валидационная и тестовая выборки?
3. Что такое переобучение модели?
4. Какие метрики используются для оценки классификатора?

5. Какие задачи решаются с помощью классификации и регрессии?
6. Что такое большая языковая модель?
7. Что такое prompt engineering?
8. Что такое структурированный вывод модели?
9. Какие ограничения имеют LLM?
10. Что такое галлюцинации и как их выявлять?
11. Для чего используются эмбединги?
12. Опишите архитектуру RAG-приложения.
13. Для чего применяется векторная база данных?
14. Как интегрировать модель в REST API?
15. Какие показатели следует контролировать при эксплуатации AI-сервиса?
16. Что такое MLOps и LLMOps?
17. Назовите основные угрозы безопасности LLM-приложений.
18. Что такое prompt injection?
19. Как предотвратить утечку конфиденциальных данных в AI-приложении?
20. Какие методы применяются для верификации ответа модели?
21. Для чего нужны журналирование и аудит действий AI-сервиса?
22. Какие требования предъявляются к итоговому проекту?
23. Как оценить воспроизводимость эксперимента?
24. Какие преимущества и недостатки имеют локальные и облачные модели?
25. Опишите этапы разработки приложения с использованием ИИ.