Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(МИ ВлГУ)

Кафедра *ФПМ*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
23.05.2023

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Технологии, методы и средства использования генеративных моделей искусственного интеллекта для разработки программного обеспечения

Направление подготовки *01.03.02 Прикладная математика и информатика*

Профиль подготовки *Интеллектуальный анализ данных*

Семестр	Трудоемкость, час./зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль, час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
7	**216 / 6**	**32**	**32**		**5,2**	**0,35**	**69,55**	**110,8**	**Экз.(35,65)**
Итого	**216 / 6**	**32**	**32**		**5,2**	**0,35**	**69,55**	**110,8**	**35,65**

Муром, 2023 г.

1. Цель освоения дисциплины

Цель изучения дисциплины заключается в формировании у студентов системного представления о возможностях, ограничениях и рисках применения генеративных моделей ИИ в промышленной разработке программного обеспечения, а также в выработке устойчивых практических навыков эффективного использования ИИ-ассистентов для автоматизации задач проектирования, кодирования, тестирования и документирования на всех этапах жизненного цикла программного продукта.

Задачи дисциплины:

– изучение архитектуры и принципов работы больших языковых моделей, лежащих в основе современных средств генерации кода;

– освоение методов промпт-инжиниринга применительно к задачам написания алгоритмов, рефакторинга, синтеза тестов и технической документации;

– формирование навыков интеграции ИИ-инструментов в процессы прототипирования, архитектурного проектирования и обратной разработки;

– приобретение практического опыта верификации сгенерированного кода, выявления уязвимостей, логических ошибок и «галлюцинаций» моделей;

– изучение правовых и этических аспектов использования генеративного ИИ при разработке коммерческого ПО.

2. Место дисциплины в структуре ОПОП ВО

Предшествующими дисциплинами, формирующими начальные знания, являются: «Технологии и методы программирования», «Теория нейронных сетей», «Технологии машинного обучения», «Структуры и алгоритмы обработки данных», «Основы информационной безопасности» и «Специальные главы математики». Знания и умения, формируемые в процессе изучения дисциплины, могут стать основой для дальнейшего успешного освоения следующих дисциплин: «Проектирование информационных систем», «Цифровая обработка изображений», а также для успешного выполнения выпускной квалификационной работы.

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1 Применяет математические методы и системы программирования для решения прикладных задач	Знать математические принципы работы и функциональные возможности генеративных моделей, а также системы программирования на их основе, применяемые для разработки и реализации алгоритмов решения прикладных задач (ОПК-2.1) Уметь обоснованно выбирать и адаптировать генеративные модели и методы промпт-инжиниринга для автоматизации процессов проектирования, кодирования и тестирования программного обеспечения	тест

		при решении прикладных задач (ОПК-2.1) Владеть навыками практического использования ИИ-ассистентов и систем программирования на базе генеративных моделей для разработки, отладки и верификации алгоритмов решения прикладных задач (ОПК-2.1)	

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

4.1. Форма обучения: очная

Уровень базового образования: среднее общее.
Срок обучения 4г.

4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Теоретические основы и методы применения генеративных моделей ИИ в разработке программного обеспечения	7	16	16						72	работа на практических занятиях
2	Инструментальные средства генеративного ИИ и их интеграция в процессы разработки программного обеспечения	7	16	16						38,8	работа на практических занятиях
Всего за семестр		216	32	32				5,2	0,35	110,8	Экз.(35,65)
Итого		216	32	32				5,2	0,35	110,8	35,65

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 7

Раздел 1. Теоретические основы и методы применения генеративных моделей ИИ в разработке программного обеспечения

Лекция 1.

Введение в генеративные модели ИИ. Эволюция, классификация и роль в программной инженерии (2 часа).

Лекция 2.

Архитектура больших языковых моделей: трансформеры, механизмы внимания, этапы предобучения (2 часа).

Лекция 3.

Обзор генеративных моделей, открытые модели (2 часа).

Лекция 4.

Промпт-инжиниринг. Базовые принципы и техники составления запросов для генерации кода (2 часа).

Лекция 5.

Продвинутые методы промпт-инжиниринга применительно к алгоритмическим задачам (2 часа).

Лекция 6.

Генерация программного кода: автоматическое написание функций, классов и модулей (2 часа).

Лекция 7.

Рефакторинг и оптимизация кода с использованием генеративных моделей (2 часа).

Лекция 8.

Прототипирование программных интерфейсов с помощью генеративных моделей (2 часа).

Раздел 2. Инструментальные средства генеративного ИИ и их интеграция в процессы разработки программного обеспечения

Лекция 9.

Интеграция ИИ-ассистентов в среды разработки (2 часа).

Лекция 10.

Автоматизация тестирования с помощью генеративных моделей (2 часа).

Лекция 11.

Генерация технической документации средствами ИИ (2 часа).

Лекция 12.

Локальный запуск открытых генеративных моделей (2 часа).

Лекция 13.

Интеллектуальный поиск по кодовой базе с использованием технологии дополненной генерации (2 часа).

Лекция 14.

Верификация сгенерированного кода. Выявление ошибок и уязвимостей (2 часа).

Лекция 15.

Правовые и этические аспекты использования генеративного ИИ при разработке ПО (2 часа).

Лекция 16.

Перспективы развития программной инженерии с ИИ-поддержкой (2 часа).

4.1.2.2. Перечень практических занятий

Семестр 7

Раздел 1. Теоретические основы и методы применения генеративных моделей ИИ в разработке программного обеспечения

Практическое занятие 1

Классификация генеративных моделей. Анализ эволюции и современных направлений развития ИИ в программной инженерии (2 часа).

Практическое занятие 2

Исследование архитектуры трансформеров. Разбор механизмов внимания на схемах и упрощённых примерах (2 часа).

Практическое занятие 3

Сравнительный анализ функциональных возможностей моделей применительно к задачам разработки ПО (2 часа).

Практическое занятие 4

Базовые техники промпт-инжиниринга: составление запросов на генерацию кода простых алгоритмов (2 часа).

Практическое занятие 5

Продвинутые техники промпт-инжиниринга: применение Chain-of-Thought для решения алгоритмических задач (2 часа).

Практическое занятие 6

Генерация программного кода: написание функций, классов и модулей по спецификации (2 часа).

Практическое занятие 7

Рефакторинг и оптимизация кода с использованием генеративных моделей (2 часа).

Практическое занятие 8

Прототипирование программных интерфейсов: генерация структуры проекта по текстовому описанию требований (2 часа).

Раздел 2. Инструментальные средства генеративного ИИ и их интеграция в процессы разработки программного обеспечения

Практическое занятие 9

Установка и настройка ИИ-ассистента в среду разработки (2 часа).

Практическое занятие 10

Автоматическая генерация модульных и интеграционных тестов для учебного проекта (2 часа).

Практическое занятие 11

Генерация технической документации: Docstrings, OpenAPI-спецификации, README-файлы (2 часа).

Практическое занятие 12

Локальный запуск открытой генеративной модели (2 часа).

Практическое занятие 13

Построение системы интеллектуального поиска по учебной кодовой базе с использованием открытых библиотек (2 часа).

Практическое занятие 14

Верификация сгенерированного кода: выявление типовых ошибок, уязвимостей и галлюцинаций модели (2 часа).

Практическое занятие 15

Комплексное применение ИИ-инструментов при выполнении учебного проекта (2 часа).

Практическое занятие 16

Комплексное применение ИИ-инструментов при выполнении учебного проекта (2 часа).

4.1.2.3. Перечень лабораторных работ

Не планируется.

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. История развития искусственного интеллекта: от первых нейросетей до современных генеративных моделей и автономных агентов.
2. Сравнительный анализ открытых и проприетарных генеративных моделей: преимущества, недостатки, области применения.
3. Обзор российского рынка ИИ-решений для разработки ПО: текущее состояние и перспективы.
4. Психологические аспекты взаимодействия разработчика с ИИ-ассистентами и агентами: когнитивные искажения и эффект чрезмерного доверия.
5. Влияние генеративных моделей и ИИ-агентов на рынок труда в сфере программной инженерии.
6. Анализ успешных кейсов внедрения ИИ-инструментов и агентных систем в российских ИТ-компаниях.

7. Архитектура и принципы построения ИИ-агентов для автоматизации задач разработки ПО.
8. Обзор фреймворков с открытым исходным кодом для создания ИИ-агентов: LangChain, AutoGPT, CrewAI и их аналоги.
9. Мультиагентные системы в программной инженерии: кооперация агентов для решения комплексных задач.
10. Этические кодексы и стандарты ответственной разработки ИИ-систем и автономных агентов.
11. Изучение подходов к оценке качества и надёжности сгенерированного кода и действий ИИ-агентов.
12. Авторское право и интеллектуальная собственность при использовании ИИ-инструментов и агентов в разработке.
13. Импортозамещение в сфере ИИ: отечественные аппаратные и программные платформы для агентных систем.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

5. Образовательные технологии

В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении лекций используется мультимедийная техника для демонстрации архитектур генеративных моделей, принципов работы ИИ-агентов и примеров взаимодействия с генеративными моделями в реальном времени. При проведении практических занятий применяется имитационный подход: обучающиеся работают с доступными генеративными моделями и средами разработки, моделируя реальные задачи — генерацию кода, рефакторинг, тестирование, документирование. Преподаватель демонстрирует типовые приёмы работы с ИИ-инструментами, после чего студенты самостоятельно выполняют аналогичные задания с последующим обсуждением результатов и разбором ошибок.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Пиляй А.И. Основы методов искусственного интеллекта [Электронный ресурс]: учебно-методическое пособие/ Пиляй А.И., Адамцевич Л.А.— Электрон. текстовые данные.— М.: МИСИ-МГСУ, ЭБС АСВ, 2023.— 60 с.— Режим доступа: https://ipr-smart.ru/142181. - https://www.iprbookshop.ru/142181.html

2. Джонс, М. Т. Программирование искусственного интеллекта в приложениях / М. Т. Джонс ; перевод А. И. Осипов. — 3-е изд. — Саратов : Профобразование, 2024. — 312 с. — ISBN 978-5-4488-0116-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/145923.html (дата обращения: 18.05.2026). - https://www.iprbookshop.ru/145923.html

3. Истратова, Е. Е. Системы искусственного интеллекта и машинное обучение : учебное пособие / Е. Е. Истратова, Е. Н. Антонянц. — Новосибирск : Новосибирский государственный технический университет, 2025. — 64 с. — ISBN 978-5-7782-5504-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/158747.html - https://www.iprbookshop.ru/158747.html

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Доверенный искусственный интеллект [Электронный ресурс]: монография/ Ю.А. Анисимов [и др.].— Электрон. текстовые данные.— Саров: Российский федеральный ядерный центр – ВНИИЭФ, 2025.— 191 c.— Режим доступа: https://ipr-smart.ru/153825. - https://www.iprbookshop.ru/153825.html

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

Информационно-аналитический ресурс по машинному обучению и интеллектуальному анализу данных — http://www.machinelearning.ru

Образовательный ресурс по современным технологиям программирования — https://metanit.com

Яндекс Образование (Хендбук по машинному обучению) — https://education.yandex.ru/handbook/ml

Программное обеспечение:

Не предусмотрено.

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru
machinelearning.ru
mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Компьютерный класс

Персональный компьютер - 12 шт., колонки Genius -1 шт., коммутатор TRENDnet TEG-S24G; видеопроектор SANYO PLC-XU355; экран Lumien Master Picture LMP-100109; IP-камера ORIENT на телескопическим кронштейне. Доступ к сети Интернет

Лекционная аудитория

Комплект учебно-методических материалов; видеопроектор ViewSonic PG603X DLP; персональный компьютер -1 шт., экран DRAPPER Apex STAR. Доступ к сети Интернет

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя,

каким дополнительным пособиям и интернет-ресурсам следует отдать предпочтение; ведёт конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями; при изучении материала обращает внимание на практическую применимость рассматриваемых методов и инструментов генеративного ИИ в задачах разработки программного обеспечения; регулярно выполняет самостоятельное практическое закрепление лекционного материала путём экспериментов с доступными генеративными моделями; отслеживает актуальные изменения и новые возможности в сфере ИИ-инструментов для разработки ПО, поскольку данная область стремительно развивается.

На практических занятиях пройденный теоретический материал подкрепляется выполнением заданий по основным темам дисциплины. Занятия проводятся в компьютерном классе с использованием доступных генеративных моделей и сред разработки. Каждому обучающемуся или подгруппе преподаватель выдаёт задание, связанное с применением ИИ-инструментов для генерации, рефакторинга, тестирования или документирования программного кода. В ходе занятия обучающиеся самостоятельно формулируют запросы к генеративным моделям, анализируют полученные результаты, выполняют верификацию сгенерированного кода и при необходимости осуществляют корректировку запросов. В конце занятия обучающиеся демонстрируют полученные результаты преподавателю, обсуждают типовые ошибки и способы их устранения.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу по изучению разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий. При освоении дисциплины особое внимание рекомендуется уделить практическому знакомству с доступными генеративными моделями и ИИ-агентами, самостоятельному экспериментированию с промпт-инжинирингом и анализу актуальных публикаций.

Форма заключительного контроля при промежуточной аттестации – экзамен. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *01.03.02 Прикладная математика и информатика* и профилю подготовки *Интеллектуальный анализ данных*
Рабочую программу составил *старший преподаватель Абрамова Е.С.*_____

Программа рассмотрена и одобрена на заседании кафедры *ФПМ*

протокол № 19 от 26.04.2023 года.
Заведующий кафедрой *ФПМ* _____*Орлов А.А.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 19.05.2023 года.
Председатель комиссии ФИТР _____ *Рыжкова М.Н.*
(Подпись) (Ф.И.О.)

Приложение

Фонд оценочных материалов (средств) по дисциплине

Технологии, методы и средства использования генеративных моделей искусственного интеллекта для разработки программного обеспечения

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

Тесты

Раздел 1. Теоретические основы и методы применения генеративных моделей ИИ в разработке программного обеспечения

Генеративные модели искусственного интеллекта — это модели, способные:
только классифицировать данные
создавать новый контент на основе обученных закономерностей
только распознавать изображения
только обрабатывать числовые данные

Архитектура трансформеров была впервые представлена в статье:
«ImageNet Classification with Deep Convolutional Networks»
«Attention Is All You Need»
«Generative Adversarial Networks»
«Deep Residual Learning for Image Recognition»

Основными компонентами архитектуры трансформера являются:
энкодер и декодер
свёрточные и пулинговые слои
решающие деревья и случайный лес
фильтры Калмана

Механизм внимания (attention) в трансформерах предназначен для:
сжатия входных данных
определения значимости различных частей входной последовательности
шифрования токенов
уменьшения размерности эмбеддингов

Self-attention (самовнимание) позволяет модели:
игнорировать длинные последовательности
учитывать взаимосвязи между всеми элементами входной последовательности
работать только с короткими текстами
обходиться без обучающих данных

Токенизация в контексте языковых моделей — это:
процесс шифрования текста
процесс разбиения текста на минимальные единицы (токены)
процесс перевода текста на другой язык
процесс сжатия текста

Эмбеддинг (embedding) — это:
вид компьютерного вируса
векторное представление слова или токена в многомерном пространстве
тип базы данных
язык программирования

Предобучение (pre-training) большой языковой модели выполняется:
на небольшом специализированном наборе данных

на масштабном корпусе текстовых данных
исключительно на программном коде
только на англоязычных текстах

Файнтюнинг (fine-tuning) модели — это:
полное переобучение модели с нуля
дообучение предобученной модели на специализированном наборе данных
удаление модели
перевод модели на другой язык программирования

Промпт-инжиниринг — это:
процесс обучения модели на новых данных
процесс проектирования и формулирования запросов к генеративной модели
процесс компиляции программного кода
процесс сжатия весов модели

Техника Zero-shot в промпт-инжиниринге подразумевает:
запрос с множеством примеров
запрос вообще без примеров
запрос с одним примером
запрос с дообучением модели

Техника Few-shot в промпт-инжиниринге означает:
запрос без примеров
запрос с одним примером
запрос с несколькими примерами
запрос с полным обучением модели

Техника Chain-of-Thought (цепочка рассуждений) применяется для:
ускорения работы модели
решения сложных задач путём пошаговых рассуждений
уменьшения размера модели
шифрования запросов

Техника Tree-of-Thought (дерево рассуждений) отличается от Chain-of-Thought тем, что:
работает быстрее
исследует несколько параллельных путей рассуждений
не требует примеров
применяется только для генерации кода

Галлюцинацией генеративной модели называют:
задержку при генерации ответа
правдоподобный, но фактически неверный ответ модели
отказ модели отвечать на запрос
повторение одного и того же ответа

Температура (temperature) в параметрах генеративной модели отвечает за:
скорость работы модели
степень случайности и креативности генерируемого текста
температуру процессора
размер модели

Какая из перечисленных моделей является отечественной разработкой?
GPT-4

Claude
GigaChat
Gemini

YandexGPT разработана компанией:
Сбер
Яндекс
VK
МТС

GigaChat разработана компанией:
Яндекс
Сбер
Тинькофф
Ростелеком

Открытая модель LLaMA разработана компанией:
Google
OpenAI
Meta
Microsoft

Раздел 2. Инструментальные средства генеративного ИИ и их интеграция в процессы разработки ПО

ИИ-агент в контексте разработки ПО — это:
среда программирования
система, способная автономно выполнять последовательность действий для достижения цели
база данных с исходным кодом
компилятор программ

Ключевое отличие ИИ-агента от обычной генеративной модели заключается в:
большем размере
способности самостоятельно планировать и выполнять многошаговые действия
использовании другого языка программирования
работе без интернета

Фреймворк LangChain предназначен для:
разработки компьютерных игр
построения цепочек вызовов языковых моделей и создания ИИ-агентов
создания баз данных
разработки мобильных приложений

Retrieval-Augmented Generation (RAG) — это технология, которая:
сжимает модель перед использованием
дополняет генерацию модели информацией из внешней базы знаний
обучает модель в реальном времени
переводит запросы на другие языки

Основное преимущество RAG-систем при работе с кодовой базой:
ускорение компиляции
возможность получать ответы, основанные на актуальном коде проекта
уменьшение размера кода
автоматическое исправление всех ошибок

При верификации сгенерированного кода в первую очередь необходимо проверить:
стиль форматирования
корректность логики и отсутствие уязвимостей
названия переменных
цветовую схему редактора

OWASP Top 10 for LLM — это:
список лучших языковых моделей
перечень основных уязвимостей, связанных с использованием больших языковых моделей
рейтинг языков программирования
справочник по архитектуре трансформеров

Prompt injection (инъекция промпта) — это вид атаки, при котором:
модель обучается на вредоносных данных
злоумышленник внедряет вредоносные инструкции в запрос к модели
перехватывается сетевой трафик
заражается операционная система

При использовании облачных генеративных моделей в коммерческой разработке основным риском для конфиденциальности является:
медленная скорость работы
передача чувствительного кода и данных на внешние серверы
высокая стоимость подписки
отсутствие технической поддержки

Мультиагентная система в программной инженерии — это:
несколько независимых программ
система из нескольких взаимодействующих ИИ-агентов, совместно решающих комплексную задачу
антивирусный комплекс
распределённая база данных

Локальное развёртывание генеративной модели предпочтительнее облачного в случае:
необходимости обеспечения конфиденциальности данных
необходимости максимальной скорости генерации
отсутствия доступа к интернету
необходимости экономии дискового пространства

GitHub Copilot относится к классу инструментов:
систем контроля версий
ИИ-ассистентов для написания кода
средств визуализации данных
инструментов управления проектами

Непрерывная интеграция (CI/CD) с использованием ИИ позволяет:
полностью исключить человека из процесса разработки
автоматически генерировать и оптимизировать сценарии сборки и развёртывания
обходиться без систем контроля версий
заменить все этапы тестирования

При генерации модульных тестов с помощью ИИ наиболее важно обеспечить:
максимальную длину тестов

покрытие всех ветвей алгоритма и граничных условий
использование только англоязычных названий
минимальное время выполнения тестов

Docstring в программном коде — это:
название переменной
строка документации, описывающая назначение функции, класса или модуля
тип данных
оператор языка программирования

Этические принципы применения ИИ в разработке ПО включают:
максимизацию прибыли любой ценой
прозрачность, подотчётность и контроль со стороны человека
полную автоматизацию всех процессов
игнорирование авторских прав

В Российской Федерации основным документом, регулирующим развитие ИИ, является:
Федеральный закон «О персональных данных»
Национальная стратегия развития искусственного интеллекта
Гражданский кодекс РФ
Трудовой кодекс РФ

При использовании ИИ-инструментов в командной разработке рекомендуется:
полностью доверять сгенерированному коду
обязательно проводить код-ревью результатов генерации
скрывать факт использования ИИ от команды
использовать ИИ только для написания комментариев

Метрика DORA в контексте ИИ-ассистированной разработки позволяет оценить:
качество пользовательского интерфейса
эффективность процессов доставки программного обеспечения
стоимость лицензий на ПО
количество строк кода

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	тестирование	до 15 баллов
Рейтинг-контроль 2	тестирование	до 15 баллов
Рейтинг-контроль 3	тестирование	до 15 баллов
Посещение занятий студентом		0
Дополнительные баллы (бонусы)		0
Выполнение семестрового плана самостоятельной работы		до 15 баллов

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой.
Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

ОПК-2
Блок 1 (знать)
Архитектура трансформеров лежит в основе:
реляционных баз данных
больших языковых моделей
систем контроля версий
компиляторов программ

Механизм самовнимания (self-attention) позволяет модели:
игнорировать длинные последовательности
учитывать взаимосвязи между всеми элементами входной последовательности
работать только с англоязычными текстами
обходиться без обучающих данных

Токенизация в языковых моделях — это:
процесс шифрования текста
процесс разбиения текста на минимальные смысловые единицы
процесс перевода на другой язык
процесс сжатия изображений

Эмбеддинг представляет собой:
вид вредоносного программного обеспечения
векторное представление токена в многомерном пространстве
тип операционной системы
формат файла исходного кода

Файнтюнинг генеративной модели — это:
удаление модели
дообучение предобученной модели на специализированном наборе данных
полное обучение модели с нуля
перевод модели на другой язык

Техника Few-shot в промпт-инжиниринге подразумевает:
запрос вообще без примеров
запрос с одним примером
запрос с несколькими примерами
запрос с полным переобучением модели

Техника Chain-of-Thought применяется для:
сжатия ответа модели
решения задач путём пошаговых рассуждений
ускорения обучения модели
шифрования запросов

Галлюцинацией генеративной модели называют:
отказ модели отвечать
правдоподобный, но фактически неверный ответ
задержку генерации
повторение одного и того же ответа

ИИ-агентом называют:
среду программирования
систему, способную автономно планировать и выполнять действия для достижения цели
систему контроля версий
базу данных

Retrieval-Augmented Generation (RAG) — это:
метод сжатия модели
технология дополнения генерации информацией из внешней базы знаний
способ шифрования запросов
алгоритм сортировки данных

Блок 2 (уметь)
Дан фрагмент кода с дублированием логики. Какой тип запроса следует использовать для его рефакторинга?
запрос на генерацию нового проекта
запрос с просьбой упростить и устранить дублирование в предоставленном коде
запрос на перевод кода на другой язык
запрос на удаление всех комментариев

Требуется сгенерировать модульные тесты для функции расчёта скидки. Какой промпт наиболее корректен?
«Напиши код»
«Сгенерируй модульные тесты на Python для функции calculate_discount, проверяющие граничные значения: нулевая сумма, отрицательная сумма, максимальная скидка»
«Сделай тесты»
«Проверь функцию»

При верификации сгенерированного моделью кода в первую очередь необходимо:
проверить цветовую схему редактора
проверить корректность логики, граничные условия и отсутствие уязвимостей
проверить название файла
измерить длину строк

Для генерации документации к функции в формате Docstring следует составить запрос, содержащий:
только название функции
код функции и указание формата документации
только список параметров
название языка программирования

При локальном развёртывании открытой генеративной модели основной выигрыш достигается в:
скорости генерации
обеспечении конфиденциальности обрабатываемых данных
размере занимаемого дискового пространства
количестве поддерживаемых языков

Блок 3 (владеть)
Дан фрагмент кода, сгенерированный моделью. Обнаружено, что при пустом входном массиве функция завершается с ошибкой. Данная ситуация является примером:
корректной работы модели

галлюцинации модели
отсутствия проверки граничных условий в сгенерированном решении
оптимального алгоритма

При анализе сгенерированного кода выявлена передача конфиденциальных данных в открытом виде. Данная проблема классифицируется как:
синтаксическая ошибка
уязвимость безопасности
особенность языка программирования
оптимизация производительности

Модель сгенерировала код, содержащий вызов несуществующей функции. Данная ситуация является примером:
корректной генерации
галлюцинации модели
оптимального решения
стандартной практики программирования

При формировании запроса к ИИ-агенту для автоматизации задач в проекте необходимо указать:
только название задачи
последовательность шагов, ожидаемый результат и ограничения
только язык программирования
только время выполнения

В проект внедряется ИИ-ассистент для командной разработки. Какое обязательное правило следует установить?
доверять сгенерированному коду без проверки
проводить обязательное код-ревью результатов генерации
запретить использование ИИ
использовать ИИ только для форматирования кода

Методические материалы, характеризующие процедуры оценивания

На основе типовых заданий программным комплексом информационно-образовательного портала МИ ВлГУ формируются в автоматическом режиме тестовые задания для студентов. Программный комплекс формирует индивидуальные задания для каждого зарегистрированного в системе студента и устанавливает время прохождения тестирования. Результатом тестирования является процент правильных ответов, с учётом индивидуального семестрового рейтинга студента формируется экзаменационная оценка.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	***Уровень сформированности компетенций***
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их	***Высокий уровень***

		выполнения оценено числом баллов, близким к максимальному	
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	***Продвинутый уровень***
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	***Пороговый уровень***
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	***Компетенции не сформированы***

3. Задания в тестовой форме по дисциплине

Примеры заданий:

Трансформерная архитектура впервые была предложена для решения задач:
компьютерного зрения
обработки естественного языка
робототехники
криптографии

Параметр «температура» при генерации текста моделью влияет на:
скорость обучения
степень вариативности и креативности ответа
размер модели
количество токенов

Мультиагентная система в программной инженерии предполагает:
использование одной модели для всех задач
совместную работу нескольких ИИ-агентов над комплексной задачей
полный отказ от использования ИИ
применение ИИ только для тестирования

Ключевым ограничением облачных генеративных моделей при работе с коммерческим кодом является:
низкая скорость генерации

риск передачи конфиденциальных данных на внешние серверы
отсутствие документации
малый объём памяти

Код-ревью результатов генерации ИИ в командной разработке:
не требуется
обязательно, поскольку модель может допускать логические ошибки и создавать уязвимости
выполняется только для проектов с открытым исходным кодом
требуется только для кода на Python

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/question/edit.php?courseid=4579&cat=76538%2C473287&qpage=0&deleteall=1&category=76525%2C473287&qbshowtext=0&recurse=0&showhidden=0

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.