

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:

Проректор по учебной работе,  
качеству образования – первый  
проректор

\_\_\_\_\_ Хагуров Т.А.

*подпись*

« 29 » августа 2025 г.

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

Б1. В.ДВ.06.01 Системы искусственного интеллекта

Направление подготовки 02.03.02 Фундаментальная информатика и  
информационные технологии

Профиль Современные методы машинного обучения и компьютерного зрения

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Системы искусственного интеллекта» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии.

Программу составил(и):

С. Г. Сеница, доцент КИТ, к.т.н.

И.О. Фамилия, должность, ученая степень, ученое звание

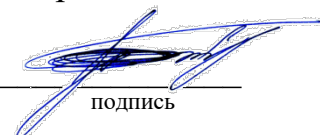


подпись

Рабочая программа дисциплины «Системы искусственного интеллекта» утверждена на заседании центра искусственного интеллекта протокол № 1 «28» августа 2025 г.

Руководитель центра ИИ Коваленко А.В.

фамилия, инициалы




подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 1 «28» августа 2025 г.

Председатель УМК факультета

А. В. Коваленко



подпись

Рецензенты:

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг»,  
e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат технических наук, профессор кафедры компьютерных технологий и систем Федерального государственного бюджетное образовательное учреждение высшего образования «Кубанский государственный аграрный университет имени И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

# 1 Цели и задачи изучения дисциплины (модуля)

## 1.1 Цель освоения дисциплины

Цель дисциплины – изучение технологий искусственного интеллекта.

## 1.2 Задачи дисциплины

- Знакомство с историей развития интеллектуальных информационных систем, современным состоянием дисциплины и перспективами развития AGI.
  - Изучение моделей представления структурированных знаний и возможностей их использования совместно с LLM.
  - Изучение архитектуры экспертных систем, систем управления знаниями организации и других прикладных систем ИИ, возможностей их использования с LLM.
  - Изучение основ инженерии онтологий и semantic web, технологии построения RAG-систем с использованием LLM и онтологий.
  - Изучение моделей интеллектуальных агентов и мультиагентных систем, применение обучения с подкреплением для реализации интеллектуальных агентов.
- Получение практического опыта реализации экспертных систем и программирования интеллектуальных агентов с использованием классических методов представления структурированных знаний и LLM.

## 1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Системы искусственного интеллекта» относится к дисциплинам по выбору, код Б1.В.ДВ.06.01.

Дисциплина в значительной степени **взаимодействует для формирования компетенций** с дисциплинами:

- Обработка естественного языка;
- Промпт-инжиниринг в профессиональной деятельности.

Требованием к «входным» знаниям является понимание основ программирования на Java и Python, базовых структур данных и алгоритмов, основ трансляции программ, основ математической логики.

## 1.4 Профессиональные роли в структуре образовательной программы

Роль 1: **Data Engineer (Инженер по данным)**

Задачи:

1. Проектирование и построение ETL-процессов
2. Создание и оптимизация хранилищ данных
3. Обеспечение качества и доступности данных
4. Настройка инфраструктуры для обработки больших данных
5. Интеграция разрозненных источников данных
6. Работа с данными в области природопользования, медицины, связи и телекоммуникаций

Роль 2: **ML Engineer (Инженер МО)**

Задачи:

1. Реализация ML-моделей в продуктивных системах
2. Оптимизация производительности и масштабирование моделей
3. Разработка ML-пайплайнов и автоматизация процессов
4. Мониторинг качества моделей в продуктиве
5. Интеграция ML-решений с бизнес-приложениями

### Роль 3: MLOps (Специалист по эксплуатации ИИ)

Задачи:

1. Автоматизация процессов обучения и развертывания моделей
2. Мониторинг производительности ML-систем
3. Управление версиями моделей и данных
4. Обеспечение CI/CD для ML-проектов
5. Оптимизация вычислительных ресурсов

#### **1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы**

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

**ПК-1** *Способен активно участвовать в исследовании новых математических моделей в естественных науках; выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем*

ПК-1.1 Разрабатывает концепцию и архитектуру программной системы, ее функциональные возможности и логику работы, делает выбор средств проектирования и реализации на основе требований с учетом существующих ограничений

Умеет разрабатывать концепцию и архитектуру программной системы, основанной на формализованных знаниях.

Знает модели представления формализованных знаний.

Владеет инструментами программирования систем обработки знаний.

ПК-1.2 Способен использовать знания о базовых принципах организации и основных этапах проектирования ИС

Знает этапы разработки онтологий.

Умеет использовать существующие и создавать новые онтологии.

Владеет инструментами разработки онтологий.

ПК-1.3 Использует методологии разработки программного обеспечения и технологии программирования, методологии и технологии проектирования и использования баз данных, методы и средства проектирования программных интерфейсов, принципы построения архитектуры программного обеспечения

Знает методологию инженерии онтологий.

Умеет применять специализированные инструменты работы с формализованными знаниями.

Владеет методами логического программирования.

**ПК-2** *Способен ориентироваться в современных алгоритмах компьютерной математики; обладать способностями к эффективному применению и реализации математически сложных алгоритмов*

ПК-2.1 Использует современные решения и технологии проектирования при разработке программного обеспечения

Знает и умеет использовать современные решения и технологии проектирования при разработке систем, основанных на формализованных знаниях.

ПК-2.2 Использует современные языки и системы программирования, технологии проектирования программного обеспечения

Использует язык программирования Java.

Владеет инструментами Java-разработки.

- ПК-2.3 Применяет критерии и методики оценки эффективности проектного решения при разработке отдельных программно-аппаратных компонентов информационных систем  
Знает и умеет применять критерии и методики оценки эффективности проектного решения при разработке систем с использованием формализованных знаний.
- ПК-2.4 Использует типовые методы контроля, оценки и обеспечения качества программного обеспечения при решении задач в различных предметных областях  
Знает типовые методы контроля и обеспечения качества систем, основанных на формализованных знаниях.  
Умеет использовать инструменты контроля баз знаний.
- LLM-4** ***Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей***
- LLM-4.1 Умеет применять и разрабатывать интеллектуальных агентов  
Использует простейших агентов в пайплайнах  
Настраивает агентов и управляет их контекстом и задачами
- LLM-4.2 Интегрирует агентов с внешними сервисами  
Подключает внешние функции и данные через wrapper Организует взаимодействие между агентом и внешними источниками
- LLM-4.3 Разрабатывает агентные паттерны  
Использует паттерны "задай-выполни"  
Реализует рассуждение на основе цепочек (ReAct, Plan&Solve)
- LLM-4.4 Управляет состоянием и памятью агентов  
Использует кратковременную память и system prompts  
Настраивает и переключает долгосрочную/контекстную память
- LLM-4.5 Оценивает и оптимизирует эффективность агентов  
Тестирует работу агента на стандартных сценариях  
Оценивает отклонения, настраивает поведение и порог доверия
- ML-1** ***Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ***
- ML-1.1 Позиционирует собственную задачу в заданной области знания с точки зрения трендов современного искусственного интеллекта  
Формулирует задачу в заданной предметной области, соотносит её с базовыми направлениями ИИ, указывает общие актуальные технологии
- ML-1.2 Определяет тенденции развития, оценивает новизну и практическую значимость своих решений с точки зрения современного искусственного интеллекта  
Определяет основные тенденции развития ИИ, оценивает новизну решения на уровне известных практик.  
Объясняет минусы традиционных подходов в сравнении с современными принципами, заложенными в технологии БД
- ML-1.3 Оценивает конкурирующие решения и разработки с точки зрения трендов современного искусственного интеллекта  
Сравнивает конкурирующие решения по базовым характеристикам (точность, скорость, применимость), ориентируясь на общепринятые подходы (например, сравнение различных ML-алгоритмов)
- MF-6** Способен применять логический аппарат для формализации задач представления знаний, проектирования логических моделей и использования систем автоматического доказательства теорем.

- MF-6.1 Применяет логические структуры для принятия решений в автоматизированных системах ИИ. Использует методы дерева решений и логистической регрессии для построения моделей.
- MF-6.2 Разрабатывает логические модели и алгоритмы для использования в ИИ. Применяет методы булевой алгебры и теории множеств для решения задач логики ИИ.
- O-1 Способен осуществлять управление знаниями, в том числе с применением алгоритмов интеллектуального поиска решений и формирования стратегий**
- O-1.1 Способен создавать базы знаний для решения задач управления бизнес-процессами предприятия  
Способен преобразовать формализованные модели бизнес-процессов в структуры баз знаний
- PL-2** Способен применять JVM-совместимые языки программирования для решения задач в области ИИ
- PL-2.1 Разрабатывает и отлаживает прикладные решения разного уровня сложности и для широкого круга конечных пользователей с использованием JVM-совместимых языков программирования, тестирует, испытывает и оценивает качество таких решений  
Применяет основные библиотеки для решения рутинных задач в серверном программировании: ввод-вывод, применение простейших примитивов многопоточного программирования, интеграция с базами данных.
- PL-2.2 Разрабатывает и поддерживает системы обработки больших данных различной степени сложности  
Разрабатывает и поддерживает простые ETL алгоритмы в пайплайнах обработки данных
- FC-3** Способен проводить фронтальные исследования в области управления, решения, агентных и мультиагентных систем
- FC-3.1 Разрабатывает алгоритмы обучения с подкреплением  
  
Владеет базовыми принципами предобучения RL-агентов на множестве сред (multi-task, multi-environment). Понимает концепции трансферного обучения (transfer learning), умеет применять готовые решения (например, R3L, Progen, OpenAI Gym Retro) для адаптации моделей к новым задачам. Знает основные метрики оценки обобщающей способности RL-агентов.
- FC-3.2 Исследует и создает агентные системы  
  
Применяет стандартные алгоритмы RL и эволюционные методы для обучения агентов в простых средах. Использует готовые фреймворки (OpenAI Gym, Stable Baselines) для быстрого прототипирования. Анализирует базовые метрики эффективности исследования среды.

## 2. Структура и содержание дисциплины

### 2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 4 зач. ед. (144 часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)					
		7					
<b>Контактная работа, в том числе:</b>	<b>72,3</b>	<b>72,3</b>					
<b>Аудиторные занятия (всего):</b>	<b>68</b>	<b>68</b>					
Занятия лекционного типа	34	34					
Лабораторные занятия	34	34					
Занятия семинарского типа (семинары, практические занятия)							
<b>Иная контактная работа:</b>	<b>4,3</b>	<b>4,3</b>					
Контроль самостоятельной работы (КСР)	4	4					
Промежуточная аттестация (ИКР)	0,3	0,3					
<b>Самостоятельная работа, в том числе:</b>	<b>36</b>	<b>36</b>					
Курсовая работа							
Проработка учебного (теоретического) материала	15	15					
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)	15	15					
Реферат							
Подготовка к текущему контролю	6	6					
<b>Контроль:</b>	<b>35,7</b>	<b>35,7</b>					
Подготовка к экзамену	35,7	35,7					
<b>Общая трудоемкость</b>	<b>час.</b>	<b>144</b>	<b>144</b>				
	<b>в том числе контактная работа</b>	<b>72,3</b>	<b>72,3</b>				
	<b>зач. ед</b>	<b>4</b>	<b>4</b>				

## 2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины. Разделы (темы) дисциплины, изучаемые в 7 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Введение в ИИ	4	4			
2.	Представление знаний	20	8		6	6
3.	Интеллектуальные информационные системы	24	4		10	10
4.	Инженерия онтологий и Semantic Web	16	6		4	6
5.	Интеллектуальные агенты	36	12		12	12
6.	Защита экзаменационных проектов	4			2	2
<b>ИТОГО по разделам дисциплины</b>		<b>104</b>	<b>34</b>		<b>34</b>	<b>36</b>
Контроль самостоятельной работы (КСР)		4				
Промежуточная аттестация (ИКР)		0,3				
Подготовка к текущему контролю		35,7				
<b>Общая трудоемкость по дисциплине</b>		<b>144</b>				

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

## 2.3 Содержание разделов (тем) дисциплины

### 2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Введение в ИИ	<p>Фундамент Искусственного Интеллекта: История, Определения и Философские Вопросы</p> <p>От мечты к науке: генезис ИИ, его суть и фундаментальные дилеммы.</p> <p>1. История Развития ИИ: Истоки (1940-1956): Кибернетика, работы Тьюринга ("Вычислительные машины и разум", Тест Тьюринга - 1950), Дартмутская конференция (1956 - рождение термина "ИИ").</p> <p>Эйфория и Первая "Зима" (1956-1974): Логические теоретики, ELIZA, блоковые миры, перцептрон. Нереалистичные ожидания и ограничения -&gt; спад финансирования.</p> <p>Экспертные системы и Вторая "Зима" (1980-1987): Расцвет символического ИИ, коммерческие ЭС (MYCIN, XCON), LISP-машины. Ограничения масштабируемости, "проблема курицы и яйца" знаний -&gt; второй спад.</p> <p>Статистический поворот и Возрождение (1980-2010): Машинное обучение (деревья решений, SVM, Байес), нейронные сети (обратное распространение), Data Mining. Deep Blue vs Каспаров (1997), победа Watson в Jeopardy! (2011).</p> <p>Эра Глубокого Обучения и Генеративного ИИ (2012-н.в.): Прорывы в компьютерном зрении (AlexNet), NLP (трансформеры), RL. AlphaGo (2016), GPT и LLM, DALL-E, Stable Diffusion.</p> <p>Текущие тренды.</p> <p>2. Определения ИИ:</p> <p>Действия: Системы, действующие разумно (Rational Action - Норвиг, Рассел).</p> <p>Мышление: Системы, мыслящие как человек (когнитивное моделирование) или мыслящие рационально (законы мышления).</p>	Р

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Тест Тьюринга (операциональное определение): Может ли машина вести беседу так, чтобы человек не отличил ее от человека?</p> <p>Современный взгляд: Создание систем, способных выполнять задачи, требующие человеческого интеллекта (восприятие, понимание языка, обучение, рассуждение, принятие решений, творчество).</p> <p>3. Философские Аспекты ИИ:          Проблема "Сильного" vs "Слабого" ИИ: Может ли машина действительно мыслить/осознавать (сильный ИИ) или лишь имитировать мышление (слабый ИИ)? (Сёрл, Китайская комната).          Проблема сознания: Что такое сознание? Может ли оно возникнуть в машине? (Qualia, трудная проблема сознания - Дэвид Чалмерс).          Этические вопросы (начало):          Ответственность, предвзятость (bias), влияние на занятость, автономное оружие.          Пределы вычислимости: Теоремы Гёделя о неполноте, проблема остановки Тьюринга - ограничивают ли они ИИ?          Визуализация/Акцент: Лента времени ИИ. Схема "Действия vs Мышление".          Китайская комната. Эксперимент "Мозг в банке".</p>	
2.		<p>Мир Интеллектуальных Систем:          Классификация, Приложения, Архитектуры и Будущее          Картография ИИ: как устроены интеллектуальные системы, где они работают и куда движутся.          1. Обзор Интеллектуальных Систем и Решаемых Задач:          Восприятие: Компьютерное зрение (CV), Распознавание речи (ASR), Обработка естественного языка (NLP).          Понимание и Генерация: Машинный перевод, Анализ тональности, Генерация текста/кода/изображений/видео/музыки (LLM, GenAI).          Рассуждение и Принятие Решений:          Экспертные системы, Логический вывод,</p>	

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Планирование, Оптимизация, Игровые ИИ.</p> <p>Обучение: Распознавание образов, Прогнозирование, Классификация, Кластеризация, Обучение с подкреплением (RL).</p> <p>2. Классификация Интеллектуальных Информационных Систем (ИИС):  По функции: Экспертные системы, Интеллектуальные обучающие системы (ИОС), Интеллектуальные информационно-поисковые системы (ИПС), Робототехнические системы, Системы поддержки принятия решений (СППР), Генеративные системы.  По методу: Символьные (правила, логика), Связонистские (нейронные сети), Эволюционные (генетические алгоритмы), Статистические (байесовские сети).  По "интеллектуальности": Жестко запрограммированные -&gt; Системы, основанные на знаниях -&gt; Самообучающиеся системы -&gt; (Гибридные).</p> <p>3. Приложения ИИ (Краткий обзор ключевых областей):  Здравоохранение (диагностика, открытие лекарств, персонализированная медицина).  Финансы (кредитный скоринг, торговые алгоритмы, обнаружение мошенничества).  Транспорт (автономные ТС, оптимизация логистики).  Розничная торговля (рекомендательные системы, управление запасами).  Промышленность (умное производство, предиктивное обслуживание).  Развлечения (гейминг, персонализация контента, генеративный арт).  Образование (адаптивное обучение, ИИ-тьюторы).  Наука (ускорение открытий, анализ больших данных).</p> <p>4. Символьный ИИ (ИИ, основанный на знаниях):  Суть: Представление знаний и манипулирование символами (логика</p>	

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>предикатов, правила продукции, семантические сети, фреймы, онтологии).</p> <p>Сильные стороны: Прозрачность, объяснимость, возможность строгого вывода.</p> <p>Слабые стороны: Сложность приобретения знаний ("бутылочное горлышко"), хрупкость, плохая работа с неопределенностью и нечеткими данными.</p> <p>Связь с Машинным Обучением (ML) и LLM:</p> <p>ML как альтернатива/дополнение: Обучение паттернам из данных, а не ручному кодированию правил.</p> <p>Гибридные системы: Символьный ИИ для представления знаний/рассуждений + ML (особенно DL) для восприятия/обучения из данных.</p> <p>Пример: Нейро-символьный ИИ (NeSy).</p> <p>LLM как "Статистические Машины Символов": Генерация символьных выводов (текста, кода) на основе статистических паттернов в данных.</p> <p>Контраст с детерминированным символьным выводом. Проблемы LLM (галлюцинации, отсутствие истинного понимания).</p> <p>5. Искусственный Общий Интеллект (AGI) - Перспективы и Дебаты:</p> <p>Определение AGI: Гипотетический ИИ, способный понимать, учиться и применять интеллект широко и гибко, как человек, а не в узкой области.</p> <p>Ключевые характеристики (гипотетические): Перенос обучения, Обобщение, Рассуждение по аналогии, Самосознание (необязательно), Понимание.</p> <p>Пути к AGI: Углубление нейросетей? Архитектуры агентов? Нейро-символьная интеграция? Принципиально новые парадигмы?</p> <p>Временные горизонты и дебаты: Оптимисты (ближайшие десятилетия) vs Скептики (столетия/никогда).</p> <p>Технологическая сингулярность.</p>	

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Риски и Безопасность AGI (AI Alignment): Проблема контроля, катастрофические сценарии, этические императивы.</p> <p>Визуализация/Акцент: Матрица классификации ИИС. Примеры гибридных систем. Сравнительная таблица Символьный ИИ / ML / LLM.</p> <p>Шкала "Узкий ИИ -&gt; AGI -&gt; Суперинтеллект".</p>	
3.	Представление знаний	<p>Продукционные системы как основа экспертных систем и систем принятия решений.</p> <p>Компоненты продукционной системы: База правил (продукций), Рабочая память (факты), Механизм логического вывода (интерпретатор).</p> <p>Стратегии управления решением: Прямая цепочка рассуждений (Data-Driven), Обратная цепочка рассуждений (Goal-Driven). Конфликтный набор и стратегии разрешения конфликтов.</p> <p>Организация поиска в пространстве состояний: Формулировка задачи (состояния, операторы, начальное и целевое состояние). Примеры задач.</p> <p>Эвристический поиск: Принципы использования эвристик. Алгоритм A*: оценка <math>f(n)=g(n)+h(n)</math>, допустимость и монотонность эвристик.</p> <p>АВ-отсечение: Принцип минимаксного поиска. Оптимизация с помощью альфа-бета отсечений. Применимость в играх и планировании.</p> <p>Примеры построения простой продукционной системы и применения A* для задачи навигации.</p>	ЛР
4.		<p>Методы решения задач и основы представления знаний</p> <p>Метод ключевых состояний и ключевых операторов: Идея выделения критических точек (ключевых состояний) и действий (ключевых операторов) для структурирования пространства поиска. Примеры задач.</p> <p>Метод анализа средств и целей (Means-Ends Analysis - MEA): Принцип работы (выявление различий между текущим и целевым состоянием, выбор оператора</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>для уменьшения различия, рекурсия). Связь с планированием. Пример (задача Ханойских башен). Обучение с подкреплением (Reinforcement Learning - RL) - основы: Агент, среда, состояние, действие, вознаграждение, политика. Идея Q-обучения. Связь с поиском оптимальных стратегий в пространстве состояний. Введение в представление знаний: Зачем нужно представление знаний? Требования к системам представления знаний (выразительность, эффективность вывода, понятность). Основные подходы. Разбор решения задачи с помощью МЕА и обсуждение применения RL в простых средах (например, антагонистические игры).</p>	
5.		<p>Формализмы представления знаний и базы знаний Классические и структурированные методы представления знаний в ИИ. Представление на правилах: Продукционные системы (углубление), достоинства и недостатки. Логические представления: Пропозициональная логика. Логика предикатов первого порядка (FOPL): синтаксис, семантика, использование для представления знаний. Резолюция как метод вывода. Ограничения. Семантические сети: Узлы (понятия, объекты, события), дуги (отношения). Наследование свойств. Достоинства и проблемы (неоднозначность, формальная семантика). Фреймы: Структура (слоты, значения, демоны). Иерархии фреймов. Наследование. Сходства и различия с семантическими сетями. Использование для представления стереотипных ситуаций. Сценарии (скрипты): Представление типизированных последовательностей событий. Компоненты сценария (роли, реквизиты, эпизоды). Базы знаний (БЗ): Архитектура БЗ. Отличие от баз данных. Принципы проектирования и наполнения БЗ.</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Системы, основанные на знаниях (Knowledge-Based Systems - KBS). Сравнительный анализ выразительности логики предикатов, семантических сетей и фреймов на примере конкретной предметной области. Построение простой иерархии фреймов.</p>	
6.		<p>Планирование, неклассические представления и современные подходы. Автоматическое планирование действий, альтернативные модели представления и роль больших языковых моделей. Планирование в интеллектуальных системах: Задача планирования (операторы, состояния, цели). STRIPS-подобные языки описания действий (предусловия, эффекты). Методы планирования: Планирование с частичным упорядочением. Планирование как поиск в пространстве планов/состояний. Эвристики в планировании. Составление расписаний (Scheduling): Отличие от классического планирования (ресурсы, временные ограничения). Основные подходы (эвристики, ограниченный перебор). Примеры автоматического построения планов: Демонстрация работы планировщиков на классических задачах (мир блоков, логистика). Сети Петри: Основы (позиции, переходы, фишки). Использование для моделирования параллельных процессов, потоков данных и управления ресурсами в ИС. Анализ свойств (достижимость, тупики). Нечеткая логика (Fuzzy Logic): Принципы (нечеткие множества, лингвистические переменные, функции принадлежности). Нечеткий вывод (фазификация, база нечетких правил, дефазификация). Применение для представления нечетких знаний и управления. LLM для систем планирования: Роль больших языковых моделей (LLM) в планировании: генерация планов по описанию на естественном языке,</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>извлечение знаний для планирования из текста, интерпретация и объяснение планов, взаимодействие "человек-планировщик". Текущие возможности и ограничения.</p> <p>Анализ простой сети Петри.</p> <p>Демонстрация применения нечеткой логики для простой задачи управления.</p> <p>Обсуждение примеров использования LLM в задачах планирования (например, в робототехнике, игровых ИИ).</p>	
7.	Интеллектуальные информационные системы	<p>Экспертные системы и адаптивные ИИ-системы: принципы, взаимодействие, эволюция</p> <p>Основы экспертных систем, механизмы принятия решений и переход к адаптивным, динамическим и персонализированным ИИ-системам.</p> <p>Экспертные системы (ЭС): Сущность и архитектура:</p> <p>Назначение ЭС: копирование экспертных знаний и компетенций в узкой предметной области.</p> <p>Ключевые компоненты: База знаний (правила, фреймы, онтологии), Механизм логического вывода, Рабочая память, Интерфейс пользователя, Подсистема объяснений.</p> <p>Взаимодействие пользователя с ЭС:</p> <p>Диалоговые режимы: консультация, объяснение, приобретение знаний.</p> <p>Роль интерфейса: естественно-языковые интерфейсы (простые формы), графические интерфейсы. Вариативность входных данных.</p> <p>Принятие решений в ЭС:</p> <p>Стратегии вывода: прямая/обратная цепочка рассуждений.</p> <p>Обработка неопределенности: коэффициенты уверенности, нечеткая логика, вероятностные методы (Байес).</p> <p>Роль подсистемы объяснений: "Как?" и "Почему?" - ключ к доверию и обучению пользователя.</p> <p>Эволюция ИИ-систем: От статических к адаптивным:</p> <p>Адаптируемые системы (Configurable): Настройка под задачи/пользователя</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>вручную (администратором/пользователем).  Адаптивные системы (Adaptive):  Автоматическая адаптация поведения во время работы на основе:  Контекста (время, место, устройство).  Обратной связи пользователя (явной/неявной).  Изменений среды (динамические данные).  Динамические системы: Фокус на обработку потоковых данных в реальном времени и быструю реакцию на изменения среды (часто основаны на обучении с подкреплением или сложных моделях событий).  Персонализируемые системы: Подвид адаптивных, где адаптация направлена на индивидуальные предпочтения, цели, историю и характеристики конкретного пользователя (рекомендательные системы - яркий пример).  LLM для экспертных систем (Перспективы и вызовы):  Генерация объяснений на естественном языке (усиление подсистемы объяснений).  Извлечение знаний из текста для пополнения базы знаний.  Улучшение интерфейсов: более естественный диалог.  Ограничения и риски: Галлюцинации, отсутствие истинного понимания, "чёрный ящик", сложность верификации выводов, зависимость от данных обучения.  Разбор архитектуры классической ЭС (например, MYCIN или Dendral). Анализ, как современный рекомендательный сервис (например, Netflix/Spotify) сочетает адаптивность, динамичность и персонализацию. Обсуждение гипотетического использования LLM как "говорящего" интерфейса для legacy экспертной системы.</p>	
8.		<p>Интеллектуальные обучающие системы (ИОС) и революция LLM в образовании.  Принципы построения интеллектуальных обучающих систем,</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>их адаптивность и трансформация под влиянием больших языковых моделей.</p> <p>Интеллектуальные обучающие системы (ИОС):</p> <p>Цели: Персонализация обучения, адаптация к уровню и стилю ученика, предоставление интеллектуальной обратной связи.</p> <p>Ключевые компоненты:</p> <p>Доменная модель: Представление знаний предметной области.</p> <p>Модель ученика: Оценка знаний, навыков, мiskonцепций, стиля обучения ученика (часто основана на Байесовских сетях или правилах).</p> <p>Педагогическая модель: Стратегии и методы обучения (что, когда и как преподавать?).</p> <p>Интерфейс пользователя:</p> <p>Взаимодействие с учеником.</p> <p>Адаптивность в ИОС: Динамическая подстройка сложности материала, темпа обучения, типа заданий и обратной связи на основе Модели ученика.</p> <p>Типы и подходы в ИОС:</p> <p>Системы-тьюторы (Intelligent Tutoring Systems - ITS).</p> <p>Системы для обучения на основе примеров и кейсов.</p> <p>Автоматизированные системы оценки знаний (с элементами ИИ).</p> <p>Обучающие среды микромиры (Intelligent Learning Environments - ILE).</p> <p>LLM для интеллектуальных обучающих систем:</p> <p>Генерация контента: Создание персонализированных задач, объяснений, примеров, учебных материалов "на лету".</p> <p>Диалоговые тьюторы: Естественно-языковое взаимодействие для объяснений, ответов на вопросы, ведения дискуссий (Socratic tutoring).</p> <p>Обратная связь: Генерация развернутых комментариев к эссе, коду, решениям задач.</p> <p>Адаптация и персонализация: Потенциал для создания более глубоких и гибких моделей ученика через анализ диалога.</p>	

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Динамическая подстройка стиля и сложности.</p> <p>Доступность: Снижение барьера для создания контента и инструментов обучения.</p> <p>Критические аспекты: Достоверность генерируемого контента, педагогическая обоснованность, отсутствие глубокого понимания предмета LLM, этические вопросы (сбор данных, влияние на преподавателей).</p> <p>LLM для экспертных систем (Углубление):</p> <p>RAG (Retrieval-Augmented Generation) для ЭС: Комбинирование точных знаний из базы экспертной системы с генеративными возможностями LLM для более точных и обоснованных ответов.</p> <p>Автоматизация поддержки знаний: Помощь экспертам в документировании и структурировании знаний для пополнения БЗ.</p> <p>Создание гибридных систем: LLM как интеллектуальный интерфейс + традиционная ЭС как надежный механизм вывода в ядре.</p> <p>Анализ компонентов классической ИОС (например, Cognitive Tutor по математике). Демонстрация современных диалоговых ИИ-тьюторов (на базе LLM типа Khanmigo, Duolingo Max).</p> <p>Обсуждение примера использования LLM + RAG в корпоративной экспертной системе для техподдержки или диагностики. Критическое сравнение возможностей и ограничений "чистых" LLM-тьюторов и традиционных ИОС.</p>	
9.	Инженерия онтологий и Semantic Web	<p>Основы онтологий и инженерии знаний. Онтологии как основа представления знаний: от философии к информатике. Принципы и методологии построения. Определение онтологии в ИИ и информатике:</p> <p>Эволюция понятия (от философии к компьютерным наукам).</p> <p>Формальное определение: Онтология как явная, формальная спецификация разделяемой концептуализации (Gruber).</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Ключевые аспекты: концептуализация, явность, формальность, разделяемость.</p> <p>Компоненты онтологии: Классы (понятия), Экземпляры (индивиды), Атрибуты (свойства, слота), Отношения, Аксиомы (ограничения), События (опционально).</p> <p>Цели и назначение онтологий в ИС: Обеспечение семантической интероперабельности, разделение знаний между людьми и системами, поддержка логического вывода, повторное использование знаний, улучшение поиска, основа для ИИ-систем (экспертные системы, ИОС, семантические сервисы).</p> <p>Методики построения онтологий: Обзор жизненного цикла разработки онтологии.</p> <p>Ключевые методологии: METHONTOLOGY: Детальный процесс-ориентированный подход.</p> <p>ОТК (Ontology Development 101 - Natalya F. Noy &amp; Deborah L. McGuinness): Практическое пошаговое руководство. Основные этапы (на примере ОТК):</p> <ul style="list-style-type: none"> <li>Определение области и масштаба.</li> <li>Выбор существующих онтологий для повторного использования.</li> <li>Перечисление ключевых терминов (глоссарий).</li> <li>Определение таксономии (иерархии классов).</li> <li>Определение свойств (слотов) классов.</li> <li>Определение фацетов свойств (домены, диапазоны, кардинальность).</li> <li>Создание экземпляров.</li> <li>Добавление аксиом и ограничений.</li> </ul> <p>Инструменты: Краткий обзор (Protégé как основной).</p> <p>Введение в Semantic Web (Семантическую Паутину): Проблема современного Веба: данные для людей, а не для машин. Понятие "Web of Documents" vs "Web of Data".</p> <p>Цель Semantic Web: Создание среды, где информация имеет четко определенный смысл, понятный компьютерам, для</p>	

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>автоматической обработки, интеграции и вывода.</p> <p>Роль онтологий в Semantic Web:  Предоставление словаря и правил для придания смысла данным в Сети.  Разбор простой предметной области (например, "Университет") и начало построения онтологии в Protégé:  определение классов (Факультет, Студент, Курс), свойств (учитсяНа, преподает).</p>	
10.		<p>Технологический стек Semantic Web: RDF, RDFS, OWL.</p> <p>Тема: Языки и стандарты представления знаний и построения онтологий для Semantic Web.</p> <p>Стек Semantic Web (Layer Cake): Обзор архитектуры (Unicode/URI, XML, RDF, RDFS, OWL, SPARQL, RIF, Trust). Фокус на слоях знаний.</p> <p>RDF (Resource Description Framework):  Базовая модель данных: Триплеты (Субъект - Предикат - Объект).  Представление знаний как графа.  Синтаксисы: RDF/XML, Turtle (N-Triples), JSON-LD (кратко).  Практическое чтение/запись триплетов.  Примеры: Представление фактов.</p> <p>RDFS (RDF Schema):  Назначение: Определение словарей (таксономий классов и свойств) для RDF.  Ключевые элементы:  Классы: rdfs:Class, rdfs:subClassOf.  Свойства: rdf:Property, rdfs:domain, rdfs:range, rdfs:subPropertyOf.  Экземпляры: rdf:type.</p> <p>Простейший логический вывод на основе иерархий (наследование типа, домена, диапазона).</p> <p>Ограничения RDFS: Невозможность выразить сложные ограничения.</p> <p>OWL (Web Ontology Language):  Назначение: Создание выразительных онтологий, поддерживающих сложный логический вывод.  Связь с логикой (Description Logics - DL):  OWL DL как основа.  Основные конструкторы (на примере OWL 2 DL):</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Классы: Более сложные классы (owl:intersectionOf, owl:unionOf, owl:complementOf), Кванторы (owl:someValuesFrom, owl:allValuesFrom), Ограничения кардинальности (owl:minCardinality, owl:exactCardinality), Эквивалентность (owl:equivalentClass).</p> <p>Свойства: Типы свойств (owl:ObjectProperty, owl:DatatypeProperty), Характеристики свойств (owl:FunctionalProperty, owl:InverseFunctionalProperty, owl:TransitiveProperty, owl:SymmetricProperty), Эквивалентность свойств (owl:equivalentProperty), Обратные свойства (owl:inverseOf).</p> <p>Индивиды: Тождество/различие (owl:sameAs, owl:differentFrom).</p> <p>Профили OWL 2: OWL 2 EL (для больших онтологий в биомедицине), OWL 2 QL (для интеграции с БД), OWL 2 RL (для эффективного вывода на основе правил).</p> <p>Логический вывод: Принцип работы reasoner'ов (HermiT, Pellet, FaCT++).</p> <p>Примеры вывода (автоматическая классификация, обнаружение противоречий).</p> <p>Практика/Пример: Построение более сложной онтологии в Protégé (например, расширение "Университета"): использование OWL-конструкторов (ОчныйСтудент как подкласс Студент, ограничение учитсяНа только на Факультет, свойство имеетСтепень с диапазоном Степень). Демонстрация работы reasoner'a.</p>	
11.		<p>Практическое применение онтологий и интеграция с современным ИИ</p> <p>Реальные приложения онтологий и Semantic Web, вызовы и синергия с большими языковыми моделями (LLM).</p> <p>Запросы к онтологиям и данным: SPARQL:</p> <p>Назначение: Язык запросов к RDF-графам (аналог SQL для реляционных БД).</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Основные паттерны запросов: SELECT, CONSTRUCT, ASK, DESCRIBE.</p> <p>Ключевые конструкции: WHERE (шаблоны триплетов), FILTER, OPTIONAL, агрегация (COUNT, SUM), сортировка (ORDER BY).</p> <p>Примеры запросов к онтологии "Университет".</p> <p>Применение онтологий в интеллектуальных системах:</p> <p>Экспертные системы и системы принятия решений: Представление предметной области и правил.</p> <p>Информационный поиск и интеграция данных (Semantic Search, Data Integration): Улучшение релевантности, преодоление неоднородности источников.</p> <p>Биомедицинские ИС (Bioinformatics): Генетика, фармацевтика (GO, SNOMED CT).</p> <p>Управление корпоративными знаниями (Enterprise Knowledge Graphs - EKG): Построение семантических сетей знаний организации.</p> <p>Интернет вещей (IoT): Семантическое описание устройств и их данных.</p> <p>Интеллектуальные обучающие системы (ИОС): Представление предметной области и модели ученика.</p> <p>Использование онтологий с LLM: Синергия и преодоление ограничений: Проблемы "чистых" LLM: Галлюцинации, отсутствие точных знаний, неконтролируемость вывода, сложность верификации, слабая связь с актуальными данными.</p> <p>Роль онтологий в работе с LLM: Структурирование знаний: Предоставление формального каркаса для организации информации, извлекаемой или используемой LLM. Контроль вывода и снижение галлюцинаций: Использование онтологии как источника достоверных фактов и ограничений для "направления" генерации LLM.</p> <p>Обоснование ответов (Explainability): Связывание ответов LLM с элементами</p>	

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>онтологии для объяснения происхождения информации.</p> <p>Интеграция с корпоративными данными (RAG - Retrieval-Augmented Generation):  Онтология как основа семантического индекса для поиска релевантных фактов из корпоративных источников перед генерацией ответа LLM.</p> <p>Автоматизация построения/обогащения онтологий: Использование LLM для:  Извлечения терминов и отношений из текста (Ontology Learning).</p> <p>Генерации определений классов и свойств.</p> <p>Нахождения экземпляров.</p> <p>Проверки согласованности (частично).</p> <p>Архитектурные паттерны: LLM как интерфейс/генератор + Онтология/БЗ как источник истины и контроллер. RAG + Онтология.</p> <p>Вызовы: Качество генерируемых онтологических элементов LLM, необходимость экспертной валидации, сложность интеграции, производительность.</p> <p>Демонстрация SPARQL-запросов к публичному датасету (например, DBPedia).</p> <p>Разбор кейса: Использование онтологии в медицинской экспертной системе для диагностики.</p> <p>Демо/Обсуждение LLM + Онтология:  Пример использования LLM API (например, OpenAI) в паре с небольшой онтологией (например, в формате RDF/OWL) для:  Ответа на вопросы, требующие фактов из онтологии (RAG-подход).</p> <p>Попытки генерации OWL-аксиом по текстовому описанию (с анализом качества результата).</p> <p>Обсуждение реальных гибридных систем (например, в фармацевтических исследованиях или техподдержке).</p>	
12.	Интеллектуальные агенты	<p>Введение в интеллектуальные агенты:  Парадигма и базовые модели</p> <p>Агентный подход в ИИ. Ключевые концепции, свойства и простейшие архитектуры.</p>	ЛР, КП

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Определение интеллектуального агента: Автономность, Реактивность, Активность (Pro-activeness), Социальность. Понятие PEAS (Performance, Environment, Actuators, Sensors).</p> <p>Классификация сред: Доступность, Детерминированность, Эпизодичность, Статичность, Дискретность.</p> <p>Модели (архитектуры) агентов: Рефлективные (простые рефлексы): `if percept then action`.</p> <p>Рефлективные с внутренним состоянием: Учет истории перцептов.</p> <p>Основанные на модели (Model-based): Внутреннее представление среды.</p> <p>Основанные на целях (Goal-based): Выбор действий для достижения цели.</p> <p>Основанные на полезности (Utility-based): Максимизация ожидаемой "полезности".</p> <p>Введение в среду Robocode: Краткий обзор (танки, сенсоры, действия, бои). Почему Robocode - отличный полигон для изучения агентов?</p> <p>Анализ PEAS для разных агентов (робот-пылесос, веб-краулер, агент в Robocode). Написание простейшего рефлективного агента для Robocode (например, "есть враг - стреляй").</p>	
13.		<p>Обучение с подкреплением (RL) для агентов: Основы и применение.</p> <p>Теоретические основы RL как ключевого подхода к созданию адаптивных агентов.</p> <p>Формальная постановка задачи RL: Агент, Среда, Состояние (S), Действие (A), Награда (R), Политика (<math>\pi</math>), Функция ценности (V, Q).</p> <p>Марковские процессы принятия решений (MDP): Определение, свойства.</p> <p>Алгоритмы RL: Q-Learning: Обновление.. Стратегии исследования-эксплуатации (<math>\epsilon</math>-жадный, softmax).</p> <p>SARSA: Обновление с учетом следующего действия.</p> <p>Основы Policy Gradients (кратко).</p> <p>Применение RL в Robocode:</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Определение состояния агента (координаты, энергия, угол до врага и т.д.).</p> <p>Определение действий (движение, поворот пушки, поворот радара, огонь).</p> <p>Определение функции награды (попадание по врагу, урон от врага, столкновение со стеной).</p> <p>Разбор кода простого Q-Learning агента для Robocode. Эксперименты с разными функциями награды и стратегиями исследования.</p>	
14.		<p>Программирование RL-агентов в Robocode: Углубление и оптимизация</p> <p>Практические аспекты реализации, отладки и улучшения RL-агентов в сложной среде Robocode.</p> <p>Проблемы RL в Robocode:</p> <p>Большое/непрерывное пространство состояний, задержки награды, частичная наблюдаемость, стохастичность.</p> <p>Методы решения:</p> <p>Аппроксимация функции ценности:</p> <p>Линейные аппроксиматоры, Нейронные сети (DQN - кратко).</p> <p>Представление состояний:</p> <p>Дискретизация, Feature Engineering.</p> <p>Обработка частичной наблюдаемости:</p> <p>Использование истории, RNN/LSTM (концептуально).</p> <p>Ускорение обучения: Experience Replay, Target Networks (DQN).</p> <p>Оптимизация гиперпараметров.</p> <p>Соревновательная среда: Обучение против разных оппонентов.</p> <p>Реализация агента с аппроксимацией Q-функции (например, линейной). Анализ логов обучения. Проведение "турнира" между простыми RL-агентами.</p>	ЛР
15.		<p>Многоагентные системы и кооперация в Robocode.</p> <p>Переход от одиночных агентов к взаимодействию в системах с множеством агентов.</p> <p>Многоагентные системы (MAS):</p> <p>Кооперация, Конкуренция, Координация.</p> <p>Распределенное представление знаний.</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>Проблемы MAS: Нестационарность среды (другие обучающиеся агенты), коммуникация.</p> <p>Подходы к обучению в MAS:</p> <p>Independent Learners (каждый учится сам по себе).</p> <p>Joint Action Learners (учет действий других).</p> <p>Team Q-Learning / Кооперативное RL (общая награда).</p> <p>Теория игр (кратко): Равновесие Нэша.</p> <p>Применение в Robocode:</p> <p>Кооперативные сценарии (2 vs 2): Согласование целей, разделение ролей (танк, снайпер), избегание friendly fire.</p> <p>Определение наград для команды.</p> <p>Практика: Разработка пары простых кооперативных RL-агентов для боя 2 vs 2 в Robocode. Анализ проблем координации.</p>	
16.		<p>LLM-агенты: Архитектура и базовые возможности.</p> <p>Тема: Использование больших языковых моделей как "мозга" интеллектуальных агентов.</p> <p>Содержание:</p> <p>Определение LLM-агента: Агент, использующий LLM как ядро для планирования, рассуждений и генерации действий.</p> <p>Ключевые компоненты архитектуры LLM-агента:</p> <p>Ядро рассуждений (LLM): Выполнение цепочек рассуждений (CoT, ReAct).</p> <p>Память: Краткосрочная (контекст окна), Долгосрочная (векторные БД, извлечение релевантности).</p> <p>Планирование: Разбивка целей на подзадачи, декомпозиция (LLM как планировщик).</p> <p>Исполнительные модули (Tools): Доступ к API, базам данных, поиску, коду, симуляторам (включая Robocode!).</p> <p>Обратная связь и рефлексия: Анализ результатов действий, коррекция планов.</p> <p>Базовые паттерны:</p> <p>ReAct (Reasoning + Acting): Цикл "Мысль-Действие-Наблюдение".</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		<p>AutoGPT / BabyAGI: Автономные циклы планирования-исполнения-рефлексии. Примеры применения: Персональные ассистенты, исследовательские агенты, агенты поддержки.</p> <p>Практика: Разбор кода простого LLM-агента (например, на LangChain) для решения текстовой задачи с использованием поиска. Обсуждение ограничений (стоимость, задержки, "галлюцинации").</p>	
17.		<p>Продвинутые LLM-агенты, гибридные подходы и будущее</p> <p>Углубление в LLM-агентов, интеграция с классическими методами (включая RL) и этические аспекты.</p> <p>Улучшение надежности LLM-агентов: Проверка планов и действий (Self-Consistency, Self-Correction).</p> <p>Критик-агенты (Critic Agents): Проверка выводов и действий основного агента.</p> <p>Снижение галлюцинаций: Grounding через внешние источники, RAG (Retrieval-Augmented Generation).</p> <p>Гибридные агенты (LLM + Классические методы):</p> <p>LLM + RL: LLM как планировщик высокого уровня / генератор целей для RL-агента низкого уровня (например, в Robocode: LLM решает стратегию "быть снайпером", RL учит тактики точной стрельбы и уклонения). LLM для генерации начальной политики RL.</p> <p>LLM + Символическое ИИ / Онтологии: Использование формальных знаний для ограничения и направления LLM.</p> <p>LLM-агенты в симуляциях и играх: Применение в Robocode (LLM для стратегии, интерпретации сенсоров, генерации кода RL-компонент).</p> <p>Проблемы реального времени.</p> <p>Многоагентные LLM-системы: Взаимодействие и коммуникация между LLM-агентами (SAML, Generative Agents). Примеры (AutoGen, GPTEam).</p> <p>Ключевые вызовы и будущее: Безопасность, управляемость, выравнивание (Alignment), ресурсоемкость, необходимость в "мире"</p>	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		(симуляторе). Роль симуляций типа Robocode в тестировании. Обсуждение архитектуры гибридного агента для Robocode (LLM + RL). Демонстрация платформы для многоагентных LLM (AutoGen). Эссе/дискуссия на тему "Этические риски автономных LLM-агентов".	

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

### 2.3.2 Занятия семинарского типа

Не предусмотрены

### 2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Представление знаний	Фреймы.	Т, ЛР
2.		Семантические сети.	Т, ЛР
3.		Нечеткая логика.	Т, ЛР
4.	Интеллектуальные информационные системы	Экспертные системы.	Т, ЛР
5.		Поиск в пространстве состояний.	Т, ЛР
6.		Планирование.	Т, ЛР
7.		Обработка ЕЯ.	Т, ЛР
8.		Составление расписаний.	Т, ЛР
9.	Инженерия онтологий и Semantic Web	Protege	Т, ЛР
10.		OWL + LLM	
11.	Интеллектуальные агенты	Robocode 1	Т, ЛР
12.		Robocode 2	Т, ЛР
13.		Robocode 3	Т, ЛР
14.		Robocode 4	Т, ЛР
15.		Robocode 5	Т, ЛР
16.		Robocode 6	Т, ЛР
17.		Турнир по Robocode	КП

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

### 2.3.4 Примерная тематика курсовых работ (проектов)

Не предусмотрены

## 2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

### **3. Образовательные технологии**

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

- Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

- Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

- Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

- Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

- Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

- Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

- Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

- Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

- Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

- Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

- Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

- работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

- проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

- анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

- развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
7	Л, ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	16
<b>Итого</b>			<b>16</b>

*Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента*

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

## 4. Оценочные и методические материалы

### 4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «название дисциплины».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме тестовых заданий, кейсов и **промежуточной аттестации** в форме вопросов и заданий к **экзамену**.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

### Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Введение в ИИ	LLM-4, ML-1, O-1, FC-3	<i>Тест</i>	<i>Вопросы к экзамену 1-3</i>
2	Представление знаний	LLM-4, ML-1, O-1, FC-3	<i>Лабораторные работы №1-4</i>	<i>Вопросы к экзамену 4-8</i>
3	Интеллектуальные информационные системы	LLM-4, ML-1, O-1, FC-3	<i>Лабораторные работы №5-8</i>	<i>Вопросы к экзамену 9-13</i>
4	Инженерия онтологий и Semantic Web	LLM-4, ML-1, O-1, FC-3	<i>Лабораторные работы №9-12</i>	<i>Вопросы к экзамену 14-18</i>
5	Интеллектуальные агенты	LLM-4, ML-1, O-1, FC-3	<i>Лабораторные работы №13-16</i>	<i>Вопросы к экзамену 19-23</i>

### Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **базовому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **удовлетворительно /зачтено**):

**ПК-1** *Способен активно участвовать в исследовании новых математических моделей в естественных науках; выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем*

ПК-1.1 Разрабатывает концепцию и архитектуру программной системы, ее функциональные возможности и логику работы, делает выбор средств проектирования и реализации на основе требований с учетом существующих ограничений

Умеет разрабатывать концепцию и архитектуру программной системы, основанной на формализованных знаниях.

Знает модели представления формализованных знаний.

Владеет инструментами программирования систем обработки знаний.

- ПК-1.2 Способен использовать знания о базовых принципах организации и основных этапах проектирования ИС  
Знает этапы разработки онтологий.  
Умеет использовать существующие и создавать новые онтологии.  
Владеет инструментами разработки онтологий.
- ПК-1.3 Использует методологии разработки программного обеспечения и технологии программирования, методологии и технологии проектирования и использования баз данных, методы и средства проектирования программных интерфейсов, принципы построения архитектуры программного обеспечения  
Знает методологию инженерии онтологий.  
Умеет применять специализированные инструменты работы с формализованными знаниями.  
Владеет методами логического программирования.
- ПК-2** Способен ориентироваться в современных алгоритмах компьютерной математики; обладать способностями к эффективному применению и реализации математически сложных алгоритмов
- ПК-2.1 Использует современные решения и технологии проектирования при разработке программного обеспечения  
Знает и умеет использовать современные решения и технологии проектирования при разработке систем, основанных на формализованных знаниях.
- ПК-2.2 Использует современные языки и системы программирования, технологии проектирования программного обеспечения  
Использует язык программирования Java.  
Владеет инструментами Java-разработки.
- ПК-2.3 Применяет критерии и методики оценки эффективности проектного решения при разработке отдельных программно-аппаратных компонентов информационных систем  
Знает и умеет применять критерии и методики оценки эффективности проектного решения при разработке систем с использованием формализованных знаний.
- ПК-2.4 Использует типовые методы контроля, оценки и обеспечения качества программного обеспечения при решении задач в различных предметных областях  
Знает типовые методы контроля и обеспечения качества систем, основанных на формализованных знаниях.  
Умеет использовать инструменты контроля баз знаний.
- LLM-4** *Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей*
- LLM-4.1 Умеет применять и разрабатывать интеллектуальных агентов  
Использует простейших агентов в пайплайнах  
Настраивает агентов и управляет их контекстом и задачами
- LLM-4.2 Интегрирует агентов с внешними сервисами  
Подключает внешние функции и данные через wrapper Организует взаимодействие между агентом и внешними источниками
- LLM-4.3 Разрабатывает агентные паттерны  
Использует паттерны "задай-выполни"  
Реализует рассуждение на основе цепочек (ReAct, Plan&Solve)
- LLM-4.4 Управляет состоянием и памятью агентов  
Использует кратковременную память и system prompts  
Настраивает и переключает долгосрочную/контекстную память
- LLM-4.5 Оценивает и оптимизирует эффективность агентов

- Тестирует работу агента на стандартных сценариях  
Оценивает отклонения, настраивает поведение и порог доверия
- ML-1** **Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ**
- ML-1.1 Позиционирует собственную задачу в заданной области знания с точки зрения трендов современного искусственного интеллекта  
Формулирует задачу в заданной предметной области, соотносит её с базовыми направлениями ИИ, указывает общие актуальные технологии
- ML-1.2 Определяет тенденции развития, оценивает новизну и практическую значимость своих решений с точки зрения современного искусственного интеллекта  
Определяет основные тенденции развития ИИ, оценивает новизну решения на уровне известных практик.  
Объясняет минусы традиционных подходов в сравнении с современными принципами, заложенными в технологии БД
- ML-1.3 Оценивает конкурирующие решения и разработки с точки зрения трендов современного искусственного интеллекта  
Сравнивает конкурирующие решения по базовым характеристикам (точность, скорость, применимость), ориентируясь на общепринятые подходы (например, сравнение различных ML-алгоритмов)
- MF-6** **Способен применять логический аппарат для формализации задач представления знаний, проектирования логических моделей и использования систем автоматического доказательства теорем.**
- MF-6.1 Применяет логические структуры для принятия решений в автоматизированных системах ИИ.  
Использует методы дерева решений и логистической регрессии для построения моделей.
- MF-6.2 Разрабатывает логические модели и алгоритмы для использования в ИИ.  
Применяет методы булевой алгебры и теории множеств для решения задач логики ИИ.
- O-1** **Способен осуществлять управление знаниями, в том числе с применением алгоритмов интеллектуального поиска решений и формирования стратегий**
- O-1.1 Способен создавать базы знаний для решения задач управления бизнес-процессами предприятия  
Способен преобразовать формализованные модели бизнес-процессов в структуры баз знаний
- PL-2** **Способен применять JVM-совместимые языки программирования для решения задач в области ИИ**
- PL-2.1 Разрабатывает и отлаживает прикладные решения разного уровня сложности и для широкого круга конечных пользователей с использованием JVM-совместимых языков программирования, тестирует, испытывает и оценивает качество таких решений  
Применяет основные библиотеки для решения рутинных задач в серверном программировании: ввод-вывод, применение простейших примитивов многопоточного программирования, интеграция с базами данных.
- PL-2.2 Разрабатывает и поддерживает системы обработки больших данных различной степени сложности  
Разрабатывает и поддерживает простые ETL алгоритмы в пайплайнах обработки данных

- FC-3** Способен проводить фронтальные исследования в области управления, решения, агентных и мультиагентных систем
- FC-3.1 Разрабатывает алгоритмы обучения с подкреплением
- Владеет базовыми принципами предобучения RL-агентов на множестве сред (multi-task, multi-environment). Понимает концепции трансферного обучения (transfer learning), умеет применять готовые решения (например, R3L, Progen, OpenAI Gym Retro) для адаптации моделей к новым задачам. Знает основные метрики оценки обобщающей способности RL-агентов.
- FC-3.2 Исследует и создает агентные системы
- Применяет стандартные алгоритмы RL и эволюционные методы для обучения агентов в простых средах. Использует готовые фреймворки (OpenAI Gym, Stable Baselines) для быстрого прототипирования. Анализирует базовые метрики эффективности исследования среды.

Соответствие **продвинутому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **хорошо /зачтено**):

- ПК-1** *Способен активно участвовать в исследовании новых математических моделей в естественных науках; выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем*
- ПК-1.1 Разрабатывает концепцию и архитектуру программной системы, ее функциональные возможности и логику работы, делает выбор средств проектирования и реализации на основе требований с учетом существующих ограничений
- Умеет разрабатывать концепцию и архитектуру программной системы, основанной на формализованных знаниях.
- Знает модели представления формализованных знаний.
- Владеет инструментами программирования систем обработки знаний.
- ПК-1.2 Способен использовать знания о базовых принципах организации и основных этапах проектирования ИС
- Знает этапы разработки онтологий.
- Умеет использовать существующие и создавать новые онтологии.
- Владеет инструментами разработки онтологий.
- ПК-1.3 Использует методологии разработки программного обеспечения и технологии программирования, методологии и технологии проектирования и использования баз данных, методы и средства проектирования программных интерфейсов, принципы построения архитектуры программного обеспечения
- Знает методологию инженерии онтологий.
- Умеет применять специализированные инструменты работы с формализованными знаниями.
- Владеет методами логического программирования.
- ПК-2** Способен ориентироваться в современных алгоритмах компьютерной математики; обладать способностями к эффективному применению и реализации математически сложных алгоритмов

- ПК-2.1 Использует современные решения и технологии проектирования при разработке программного обеспечения  
Знает и умеет использовать современные решения и технологии проектирования при разработке систем, основанных на формализованных знаниях.
- ПК-2.2 Использует современные языки и системы программирования, технологии проектирования программного обеспечения  
Использует язык программирования Java.  
Владеет инструментами Java-разработки.
- ПК-2.3 Применяет критерии и методики оценки эффективности проектного решения при разработке отдельных программно-аппаратных компонентов информационных систем  
Знает и умеет применять критерии и методики оценки эффективности проектного решения при разработке систем с использованием формализованных знаний.
- ПК-2.4 Использует типовые методы контроля, оценки и обеспечения качества программного обеспечения при решении задач в различных предметных областях  
Знает типовые методы контроля и обеспечения качества систем, основанных на формализованных знаниях.  
Умеет использовать инструменты контроля баз знаний.
- LLM-4** ***Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей***
- LLM-4.1 Умеет применять и разрабатывать интеллектуальных агентов  
Использует простейших агентов в пайплайнах  
Настраивает агентов и управляет их контекстом и задачами
- LLM-4.2 Интегрирует агентов с внешними сервисами  
Подключает внешние функции и данные через wrapper Организует взаимодействие между агентом и внешними источниками
- LLM-4.3 Разрабатывает агентные паттерны  
Использует паттерны "задай-выполни"  
Реализует рассуждение на основе цепочек (ReAct, Plan&Solve)
- LLM-4.4 Управляет состоянием и памятью агентов  
Использует кратковременную память и system prompts  
Настраивает и переключает долгосрочную/контекстную память
- LLM-4.5 Оценивает и оптимизирует эффективность агентов  
Тестирует работу агента на стандартных сценариях  
Оценивает отклонения, настраивает поведение и порог доверия
- ML-1** ***Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ***
- ML-1.1 Позиционирует собственную задачу в заданной области знания с точки зрения трендов современного искусственного интеллекта  
Формулирует задачу в заданной предметной области, соотносит её с базовыми направлениями ИИ, указывает общие актуальные технологии
- ML-1.2 Определяет тенденции развития, оценивает новизну и практическую значимость своих решений с точки зрения современного искусственного интеллекта  
Определяет основные тенденции развития ИИ, оценивает новизну решения на уровне известных практик.  
Объясняет минусы традиционных подходов в сравнении с современными принципами, заложенными в технологии БД

- ML-1.3 Оценивает конкурирующие решения и разработки с точки зрения трендов современного искусственного интеллекта  
Сравнивает конкурирующие решения по базовым характеристикам (точность, скорость, применимость), ориентируясь на общепринятые подходы (например, сравнение различных ML-алгоритмов)
- MF-6 Способен применять логический аппарат для формализации задач представления знаний, проектирования логических моделей и использования систем автоматического доказательства теорем.
- MF-6.1 Применяет логические структуры для принятия решений в автоматизированных системах ИИ.  
Использует методы дерева решений и логистической регрессии для построения моделей.
- MF-6.2 Разрабатывает логические модели и алгоритмы для использования в ИИ.  
Применяет методы булевой алгебры и теории множеств для решения задач логики ИИ.
- O-1 Способен осуществлять управление знаниями, в том числе с применением алгоритмов интеллектуального поиска решений и формирования стратегий**
- O-1.1 Способен создавать базы знаний для решения задач управления бизнес-процессами предприятия  
Способен преобразовать формализованные модели бизнес-процессов в структуры баз знаний
- PL-2** Способен применять JVM-совместимые языки программирования для решения задач в области ИИ
- PL-2.1 Разрабатывает и отлаживает прикладные решения разного уровня сложности и для широкого круга конечных пользователей с использованием JVM-совместимых языков программирования, тестирует, испытывает и оценивает качество таких решений  
Применяет основные библиотеки для решения рутинных задач в серверном программировании: ввод-вывод, применение простейших примитивов многопоточного программирования, интеграция с базами данных.
- PL-2.2 Разрабатывает и поддерживает системы обработки больших данных различной степени сложности  
Разрабатывает и поддерживает простые ETL алгоритмы в пайплайнах обработки данных
- FC-3** Способен проводить фронтальные исследования в области управления, решения, агентных и мультиагентных систем
- FC-3.1 Разрабатывает алгоритмы обучения с подкреплением  
  
Владеет базовыми принципами предобучения RL-агентов на множестве сред (multi-task, multi-environment). Понимает концепции трансферного обучения (transfer learning), умеет применять готовые решения (например, R3L, Progen, OpenAI Gym Retro) для адаптации моделей к новым задачам. Знает основные метрики оценки обобщающей способности RL-агентов.
- FC-3.2 Исследует и создает агентные системы  
  
Применяет стандартные алгоритмы RL и эволюционные методы для обучения агентов в простых средах. Использует готовые фреймворки (OpenAI Gym, Stable Baselines) для быстрого прототипирования. Анализирует базовые метрики эффективности исследования среды.

Соответствие экспертному уровню освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **отлично /зачтено**):

**ПК-1** *Способен активно участвовать в исследовании новых математических моделей в естественных науках; выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем*

ПК-1.1 Разрабатывает концепцию и архитектуру программной системы, ее функциональные возможности и логику работы, делает выбор средств проектирования и реализации на основе требований с учетом существующих ограничений

Умеет разрабатывать концепцию и архитектуру программной системы, основанной на формализованных знаниях.

Знает модели представления формализованных знаний.

Владеет инструментами программирования систем обработки знаний.

ПК-1.2 Способен использовать знания о базовых принципах организации и основных этапах проектирования ИС

Знает этапы разработки онтологий.

Умеет использовать существующие и создавать новые онтологии.

Владеет инструментами разработки онтологий.

ПК-1.3 Использует методологии разработки программного обеспечения и технологии программирования, методологии и технологии проектирования и использования баз данных, методы и средства проектирования программных интерфейсов, принципы построения архитектуры программного обеспечения

Знает методологию инженерии онтологий.

Умеет применять специализированные инструменты работы с формализованными знаниями.

Владеет методами логического программирования.

**ПК-2** *Способен ориентироваться в современных алгоритмах компьютерной математики; обладать способностями к эффективному применению и реализации математически сложных алгоритмов*

ПК-2.1 Использует современные решения и технологии проектирования при разработке программного обеспечения

Знает и умеет использовать современные решения и технологии проектирования при разработке систем, основанных на формализованных знаниях.

ПК-2.2 Использует современные языки и системы программирования, технологии проектирования программного обеспечения

Использует язык программирования Java.

Владеет инструментами Java-разработки.

ПК-2.3 Применяет критерии и методики оценки эффективности проектного решения при разработке отдельных программно-аппаратных компонентов информационных систем

Знает и умеет применять критерии и методики оценки эффективности проектного решения при разработке систем с использованием формализованных знаний.

- ПК-2.4 Использует типовые методы контроля, оценки и обеспечения качества программного обеспечения при решении задач в различных предметных областях  
Знает типовые методы контроля и обеспечения качества систем, основанных на формализованных знаниях.  
Умеет использовать инструменты контроля баз знаний.
- LLM-4** ***Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей***
- LLM-4.1 Умеет применять и разрабатывать интеллектуальных агентов  
Использует простейших агентов в пайплайнах  
Настраивает агентов и управляет их контекстом и задачами
- LLM-4.2 Интегрирует агентов с внешними сервисами  
Подключает внешние функции и данные через wrapper Организует взаимодействие между агентом и внешними источниками
- LLM-4.3 Разрабатывает агентные паттерны  
Использует паттерны "задай-выполни"  
Реализует рассуждение на основе цепочек (ReAct, Plan&Solve)
- LLM-4.4 Управляет состоянием и памятью агентов  
Использует кратковременную память и system prompts  
Настраивает и переключает долгосрочную/контекстную память
- LLM-4.5 Оценивает и оптимизирует эффективность агентов  
Тестирует работу агента на стандартных сценариях  
Оценивает отклонения, настраивает поведение и порог доверия
- ML-1** ***Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ***
- ML-1.1 Позиционирует собственную задачу в заданной области знания с точки зрения трендов современного искусственного интеллекта  
Формулирует задачу в заданной предметной области, соотносит её с базовыми направлениями ИИ, указывает общие актуальные технологии
- ML-1.2 Определяет тенденции развития, оценивает новизну и практическую значимость своих решений с точки зрения современного искусственного интеллекта  
Определяет основные тенденции развития ИИ, оценивает новизну решения на уровне известных практик.  
Объясняет минусы традиционных подходов в сравнении с современными принципами, заложенными в технологии БД
- ML-1.3 Оценивает конкурирующие решения и разработки с точки зрения трендов современного искусственного интеллекта  
Сравнивает конкурирующие решения по базовым характеристикам (точность, скорость, применимость), ориентируясь на общепринятые подходы (например, сравнение различных ML-алгоритмов)
- MF-6** ***Способен применять логический аппарат для формализации задач представления знаний, проектирования логических моделей и использования систем автоматического доказательства теорем.***
- MF-6.1 Применяет логические структуры для принятия решений в автоматизированных системах ИИ.  
Использует методы дерева решений и логистической регрессии для построения моделей.
- MF-6.2 Разрабатывает логические модели и алгоритмы для использования в ИИ.  
Применяет методы булевой алгебры и теории множеств для решения задач логики ИИ.

- О-1**            **Способен осуществлять управление знаниями, в том числе с применением алгоритмов интеллектуального поиска решений и формирования стратегий**
- О-1.1            Способен создавать базы знаний для решения задач управления бизнес-процессами предприятия  
Способен преобразовать формализованные модели бизнес-процессов в структуры баз знаний
- PL-2**            **Способен применять JVM-совместимые языки программирования для решения задач в области ИИ**
- PL-2.1            Разрабатывает и отлаживает прикладные решения разного уровня сложности и для широкого круга конечных пользователей с использованием JVM-совместимых языков программирования, тестирует, испытывает и оценивает качество таких решений  
Применяет основные библиотеки для решения рутинных задач в серверном программировании: ввод-вывод, применение простейших примитивов многопоточного программирования, интеграция с базами данных.
- PL-2.2            Разрабатывает и поддерживает системы обработки больших данных различной степени сложности  
Разрабатывает и поддерживает простые ETL алгоритмы в пайплайнах обработки данных
- FC-3**            **Способен проводить фронтальные исследования в области управления, решения, агентных и мультиагентных систем**
- FC-3.1            Разрабатывает алгоритмы обучения с подкреплением  
  
Владеет базовыми принципами предобучения RL-агентов на множестве сред (multi-task, multi-environment). Понимает концепции трансферного обучения (transfer learning), умеет применять готовые решения (например, R3L, Procsen, OpenAI Gym Retro) для адаптации моделей к новым задачам. Знает основные метрики оценки обобщающей способности RL-агентов.
- FC-3.2            Исследует и создает агентные системы  
  
Применяет стандартные алгоритмы RL и эволюционные методы для обучения агентов в простых средах. Использует готовые фреймворки (OpenAI Gym, Stable Baselines) для быстрого прототипирования. Анализирует базовые метрики эффективности исследования среды.

**Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

***Практические задания по тематике лабораторных работ***

**1. Фреймы**

Представить в виде системы фреймов и метафреймов, записать в EAV-таблицу, записать в виде программы Prolog или Python:

а) Катя, София и Мария -- женщины, Сергей, Петр, Иван и Даниил -- мужчины. Иван родитель Кати, Сергей родитель Софии и Даниила, Катя родитель Даниила, Даниил родитель Марии и Петра.

Описать предикаты:

Пример:           Сын,  
woman(katrin).    Дочь,  
woman(sofia).    Мама,  
woman(mary).     Папа,  
man(sergey).      Бабушка,  
man(daniel).      Дедушка,  
man(petr).        Дядя,  
man(ivan).        Тетья,  
parent(ivan,katrin). Предок ancestor(A, B),  
parent(sergey,sofia). Родственник relative(A, B)  
parent(sergey,daniel).  
parent(katrin,daniel).  
parent(daniel,mary).  
parent(daniel,petr).

Записать последовательность унификаций, подстановок, резолюций и откатов при вычислении цели: relative(petr, sofia).

б) Краснодар - столица краснодарского края. Население Краснодара 1 млн. Краснодарский край граничит с Ростовской областью и Ставропольским краем. Столица ростовской области - город Ростов-на-Дону с населением 1.5 млн. В ростовской области находится город Таганрог с населением 250 тыс. Река Кубань протекает по Краснодарскому краю и Ростовской области.

в) Студент университета обучается в группе. На каждом курсе несколько групп. Бакалавр пишет диссертацию у научного руководителя на 4 курсе. Некоторые бакалавры поступают в магистратуру. Магистр пишет диссертацию у научного руководителя на 2м году обучения.

## 2. Семантические сети

а) Нарисуйте семантическую сеть для задачи 1 а).

б) Нарисуйте семантическую сеть для задачи 1 б).

в) Нарисуйте семантическую сеть для задачи 1 в).

г) Нарисуйте семантическую сеть для базы знаний:

cond(1, 'кормит детенышей молоком').

cond(2, 'имеет перья').

cond(3, 'плавает').

cond(4, 'ест мясо').

cond(5, 'имеет копыта').

cond(6, 'летает').

cond(7, 'откладывает яйца').

cond(8, 'имеет шерсть').

cond(9, 'имеет полосы').

cond(10, 'имеет пятна').

cond(11, 'имеет черно-белую окраску').

```
rule('гепард', [1,4,8,10]).
rule('тигр', [1,4,8,9]).
rule('зебра', [1,5,8,9,11]).
rule('пингвин', [2,3,11]).
rule('орел', [2,6]).
rule('кит', [1,3,11]).
```

д) Представьте следующие предложения в виде семантической сети и фактов в прологе:

Иван дает своим студентам много трудных заданий. Мария дает студентам Ивана огромное количество сложных заданий.

Предположим, нам нужно написать систему, которая определит кто дает студентам Ивана больше всех заданий. Как можно это сделать?

### 3. Экспертные системы

Реализовать экспертную систему на Prolog, должна задавать вопросы пользователю и находить подходящий ответ:

- от 10 фактов,
- от 5 правил,
- возможность кроме да и нет отвечать неопределенно,
- возможность более чем одного варианта ответа,
- обучение (при отсутствии подходящего ответа или не правильном ответе должна спрашивать правильный ответ и пополнять базу знаний дописывая факты в файл).

### 4. Нечеткая логика

а) Почитать спецификацию FCL (IEC 1331 part 7, файл [iec\\_1131\\_7\\_cd1.pdf](#)).

б) Почитать описание, примеры и API библиотеки jFuzzyLogic (каталог Docs и сайт <http://jfuzzylogic.sourceforge.net/>). Разобрать и запустить TestTipper.java и документацию библиотеки simplful

в) Придумать нечеткий контроллер какого-либо механизма, например климат-контроль, насос для откачки воды, автономный робот-марсоход, беспилотный летательный аппарат. Контроллер должен содержать минимум четыре входные переменные, две выходные, по три правила на каждую выходную переменную. Задать контроллер в виде FCL-файла. Продемонстрировать его работу на разных значениях входных переменных с помощью программы на Java или Python. С помощью LLM реализовать интерфейс к нечеткому контроллеру на естественном языке.

5. Поиск в пространстве состояний. По шаблону поиска в пространстве состояний написать программу (реализовать 'start', 'goal', 'next\_state' и 'safe\_state' на Prolog) для поиска минимального решения одной из задач:

Фермер-волк-коза-капуста  
(классика)

Миссионеры и каннибалы

3 миссионера и 3 каннибала на одной стороне, перевести на другую живыми, в лодке от одного до 2-х человек. Если каннибалы оказываются в большинстве, то они съедают миссионера.

Ревнивые мужья

Три ревнивых мужа переправляются с женами через реку в

двухместной лодке. Как переправить, чтобы ни одна жена с чужим мужем не оставалась в лодке или на берегу.

#### 6. Поиск A\*

Реализовать "пятнашки" алгоритмом A\*, реализовать проверку на заикливание, составить эвристическую функцию для цели:

1 2 3 4  
5 6 7 8  
9 10 11 12  
13 14 15

#### 7. Проект ИИ (АВ-поиск, обучение с подкреплением)

Реализовать антагонистическую игру (человек против компьютера) с использованием поиска в пространстве состояний на глубину два или более ходов, использовать алгоритм АВ-отсечения и обучение с подкреплением для функции игры. Одна игра на трех человек, игры не должны повторяться в группе. Примеры хороших игр: рендзю (сложно), шашки (сложно), уголки, реверси, 3D крестики-нолики, крестики нолики по 4 с гравитацией (просто). Компьютер должен выигрывать у преподавателя!

#### 8. Планирование в мире блоков

Блоки лежат на столе или друг на друге. Блоки не могут висеть в воздухе. На один блок можно поставить не более одного блока. Например:

on(a, table).

on(c, a).

on(b, table).

За одно действие можно переставить только один блок. Написать планировщик действий по достижению определенного расположения блоков из заданного начального, используя подход анализа целей и средств.

Реализовать защиту целей, см. пример

plan([on(b, c), on(a, b), on(c, table)]).

Сформулировать в предикатах понятие правильного списка целей в зависимости от того, может ли данный список целей быть полностью выполнен для мира физических блоков.

Например, [on(a, b), on(b,a)] — некорректно. Написать предикат, который бы анализировал корректность перемещений блоков. Подсказка:

- стол не может быть сверху чего-либо
- нельзя двигать несуществующие блоки
- блок не может быть сверху себя
- нельзя поставить два блока на один
- нельзя поставить один блок на два блока
- все блоки должны на чем-то лежать
- ...

Переписать на списках без assert и retract, явно хранить список удаления и добавления (не обязательное задание).

#### 9. Обработка ЕЯ

а) С помощью DCG построить контекстно-зависимую грамматику на русском языке.

б) Идиоматический интерфейс в мире блоков

Для мира блоков сформулировать идиомы «what is block X sitting on?», «which blocks are on the table?», «put all blocks in single pile.», «put the block on top of X on top of block Y (or on the table).» С помощью LLM классифицировать произвольную команду на ЕЯ как одну из идиом с необходимыми параметрами и вызвать планировщик.

в) Идиоматический интерфейс экспертной системы

Для экспертной системы 3 сформулировать не менее 5 идиом вида «Какие животные умеют летать?», «Кто кормит детенышей молоком?» и т.д. Строки из базы знаний по возможности не должны дублироваться в грамматике. С помощью LLM классифицировать произвольную команду на ЕЯ как одну из идиом экспертной системы и вызвать соответствующую идиому.

10. Составление расписаний, программирование в ограничениях

Реализовать задачу 8 ферзей через CLP-FD. Разобрать программу составления расписаний школьных занятий Маркуса Триски. Прочитать его диссертацию про CLP-FD в SWI-Prolog. С помощью LLM преобразовать пожелания преподавателя на естественном языке в CLP-FD ограничение в системе расписаний.

11. Онтология

С помощью Protege построить онтологию для задач 1 а) и б). Построить онтологию по индивидуальному заданию, предметные области не должны совпадать в группе, продемонстрировать на примере все свойства отношений (рефлексивное, симметричное, транзитивное, функциональное, инверсно-функциональное и т.д.).

12. Агенты (RoboCode или робототехника)

Поучаствовать в турнире факультета по RoboCode.

Программа Robocode должна содержать один или более следующих алгоритмов:

- а) Захват и отслеживание цели, переключение на наиболее близкую и/или опасную цель.
- б) Стрельба с упреждением по противнику с учетом предсказания его перемещений по результатам предыдущих наблюдений.
- в) Уход с линии огня, избегание столкновений по результатам анализа истории наблюдений с помощью реализации обучения с подкреплением (Reinforcement Learning), допускается использование библиотек.
- г) Выбор стратегии поведения в зависимости от результатов предыдущих раундов с помощью реализации обучения с подкреплением (Reinforcement Learning), допускается использование библиотек.

Для студентов, участвующих в соревнованиях по робототехнике, данная задача может быть заменена на защиту проекта по управлению роботом с применением машинного зрения (распознавание объектов) или обучения с подкреплением.

**Пример лабораторной работы**

*Лабораторная работа №2: Представление знаний*

Тема: Построение семантической сети и фреймовой модели

*Цель работы:*

Научиться представлять знания в виде семантической сети и фреймовой модели , а также применять эти методы для моделирования предметной области.

*Задачи:*

Построить семантическую сеть для предметной области (например, "Смартфоны").

Реализовать фреймовую модель на основе этой сети.

Описать отношения IS-A и Part-Of.

Создать EV-таблицу для одного из фреймов.

Проанализировать эффективность и удобство обоих подходов.

*Ожидаемые результаты:*

Семантическая сеть в графическом виде.

Фреймовая модель в табличном или текстовом виде.

EV-таблица для одного из объектов.

Отчет с анализом применения методов представления знаний.

*Ход работы*

1. Выбор предметной области

Выбрать область, например:

- "Смартфоны"
- "Автомобили"
- "Птицы"
- "Игровые ноутбуки"

2. Построение семантической сети

Пример:

- Узлы: iPhone 15, смартфон, Apple, камера, экран, процессор
- Связи: iPhone 15 IS-A смартфон, iPhone 15 HAS камера, камера HAS разрешение, процессор MANUFACTURED\_BY Apple.

Рекомендуется использовать инструменты:

- draw.io
- yEd
- Graphviz
- рукописная схема (в отчете можно отсканировать)

3. Построение фреймовой модели

Пример фрейма:

```
(Frame: iPhone 15
  (IS-A: смартфон)
  (HAS: камера)
  (HAS: экран)
  (HAS: процессор)
  (MANUFACTURED_BY: Apple)
)
```

Каждый объект должен быть представлен как отдельный фрейм. Описать слоты и их значения.

4. Создание EV-таблицы

Для одного из объектов (например, камера iPhone 15) создать EV-таблицу:

СЛОТ	ЗНАЧЕНИЕ	ЕДИНИЦА ИЗМЕРЕНИЯ
Разрешение	12 мегапикселей	Мп
Апертура	f/1.8	-
Оптическая стабилизация	Да	-

#### 5. Анализ эффективности

- Какие отношения легче выражаются в семантической сети?
- Какие преимущества дают фреймы?
- Какие сложности возникли при моделировании?
- В каких ИИ-системах может использоваться эта модель?

#### *Требования к отчету*

##### 1. Титульный лист

Название работы, ФИО студента, группа, дата.

##### 2. Введение

Цель и задачи работы.

Описание датасета (объем, типы ошибок).

##### 3. Теоретическая часть

Кратко описать:

- Что такое семантические сети
- Что такое фреймы
- Что такое EV-таблицы
- Отношения IS-A и Part-Of

##### 4. Реализация

Схема семантической сети (можно вставить как изображение)

Текстовое представление фреймов

EV-таблица

Комментарии к выбору слотов и связей.

##### 5. Результаты

Выводы по применимости методов.

Сравнение наглядности и удобства семантической сети и фреймов.

##### 6. Выводы

Проблемы, возникшие при обработке.

Предложения по улучшению.

##### 7. Приложения

Исходные файлы (схемы, фреймы, таблицы)

Словарь терминов (если использовались специфичные).

#### *Критерии оценки*

Отлично (5): Полная реализация, грамотная модель, четкие выводы, высокая наглядность.

Хорошо (4): Хорошая реализация, но есть мелкие недочеты в модели или выводах.

Удовлетворительно (3): Выполнены основные этапы, но модель не полностью соответствует предметной области.

Неудовлетворительно (2): Не выполнены ключевые этапы, модель не отражает смысл.

#### *Рекомендуемые инструменты*

- draw.io / yEd / Graphviz – для построения семантической сети
- Текстовый редактор (Notepad++, Word) – для оформления фреймов и EV-таблиц
- Python или Prolog – для представления фреймов в памяти программы

#### ***Тестовые вопросы по дисциплине***

##### 1. Введение в системы ИИ

Какое утверждение наиболее точно описывает ИИ?

- a) Программа, которая может играть в шахматы
- b) Система, способная воспринимать среду, учиться и действовать рационально
- c) Автомат, выполняющий жёстко заданные инструкции
- d) Система, способная выполнять любые задачи быстрее человека

##### 2. Экспертные системы

Что не является компонентом экспертной системы?

- a) Машина вывода
- b) База знаний
- c) Генератор отчетов
- d) Интерфейс пользователя

##### 3. Продукционные системы

Какой метод вывода начинается с целей и ищет пути их достижения?

- a) Прямой
- b) Обратный
- c) Смешанный
- d) Гибридный

##### 4. Prolog

Какой механизм в Prolog позволяет вернуться к предыдущему выбору при неудаче?

- a) Отсечение
- b) Возврат (backtracking)
- c) Унификация
- d) Рекурсия

##### 5. Фреймы

Как называется элемент фрейма, описывающий его характеристику?

- a) Атрибут
- b) Слот
- c) Значение
- d) Дескриптор

##### 6. Семантические сети

Какой из подходов к машинному переводу не является семантическим?

- a) Статистический
- b) Синтаксический
- c) Логический
- d) Глубокий семантический

7. Обработка естественного языка

Какой инструмент в Prolog используется для описания грамматики?

- a) DCG
- b) CFG
- c) NLP
- d) DTD

8. Поиск в пространстве состояний

Какой алгоритм использует эвристическую функцию для поиска кратчайшего пути?

- a) BFS
- b) DFS
- c) A\*
- d) Dijkstra

9. Нейронные сети

Какой тип нейронной сети лучше всего подходит для распознавания изображений?

- a) Рекуррентная
- b) Полносвязная
- c) Сверточная
- d) Радиально-базисная

10. Планирование действий

Какой метод планирования использует декомпозицию целей?

- a) A\*
- b) BFS
- c) GPS
- d) STRIPS

11. Адаптивные системы

Какой метод используется для персонализации контента в обучающих системах?

- a) Кластеризация
- b) Рекомендательные системы
- c) Линейная регрессия
- d) PCA

12. RDF и графовые модели

Какой формат данных используется для описания информации в виде триплетов?

- a) JSON
- b) XML
- c) RDF
- d) CSV

13. Semantic Web

Какой язык используется для описания онтологий в Semantic Web?

- a) RDF
- b) RDFS

- c) OWL
- d) XML

14. Нечеткая логика

Какое множество описывается с помощью функции принадлежности?

- a) Четкое
- b) Нечеткое
- c) Случайное
- d) Дискретное

15. Интеллектуальные агенты

Какой тип агента наиболее близок к человеку?

- a) Реактивный
- b) Адаптивный
- c) Рациональный
- d) Простой

**Проверяемые компетенции комплексом практических заданий:** LLM-4, ML-1, O-1, FC-3.

**Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)**

**Вопросы для подготовки к экзамену**

1. Введение в системы искусственного интеллекта. Обзор интеллектуальных систем и решаемых задач. Классификация ИИС.
2. Экспертные системы.
3. Продукционные системы. Прямой и обратный вывод. Базы знаний.
4. Схема работы Prolog. Построение дерева вывода, возврат и отсечение.
5. Фреймы. Понятие слота. EV-таблицы. Отношения IS-A и PartOf.
6. Модель языка. Семантические сети. Задача автоматизированного перевода. Синтаксический, семантический и статистический подход к переводу. Семантизация утверждений и запросов.
7. Задача парсинга при обработке естественного языка, DCG в Prolog, пример.
8. Поиск в пространстве состояний, алгоритм A\*, алгоритм A-B отсечения.
9. Цели и задачи систем управления знаниями организаций. Первое поколение систем. Инженеры знаний. Проблемы внедрения систем первого поколения. Второе поколения систем управления знаниями организаций.
10. Методы машинного обучения. Генетические алгоритмы и генетическое программирование.
11. Нейронные сети, перцептрон. Системы классификации и распознавания на нейронных сетях. Обучение обратным распространением ошибки.
12. Системы распознавания образов, задача распознавания структуры документов и текста, машинное зрение, задача распознавания лиц (2D и 3D случаи). Сверточные нейронные сети.
13. Задача планирования действий, алгоритм анализа целей и средств.
14. Адаптивные и персонализируемые системы. Методы адаптации навигации и содержимого. Интеллектуальные обучающие системы. Адаптация в интеллектуальных обучающих системах.
15. Интеграция информационных систем. Понятие интероперабельности. XML. XSD (XML-Schema). Элементарные типы данных в XML-Schema.
16. Понятие пространства имен в XML. Schema.org. Dublin Core.
17. Информационный ресурс, атрибуты ресурсов и графовая модель. Обзор RDF и RDFS.

18. Словарь, тезаурус, таксономия, онтология. Этапы построения онтологии.
19. Обзор Semantic Web, стек протоколов, OWL.
20. Нечеткая логика. Определение нечеткого множества и операций над ними, определение нечеткой и лингвистической переменной.
21. Интеллектуальные Агенты. Модели агентов, определение рационального агента и функции агентов. Примеры применения моделей агентов в ИИ.

***Перечень компетенций (части компетенции), проверяемых оценочным средством LLM-4, ML-1, O-1, FC-3.***

#### ***Практические задания к экзамену***

1. Напишите базу знаний на языке Prolog или Python для модели родственных отношений. Реализуйте предикаты родитель/2, муж/1, жен/1, и определите на их основе предикаты отец/2, мать/2, брат/2, сестра/2.
2. Реализуйте простую продукционную систему с механизмом прямого вывода. Создайте 5 правил и рабочую память. Продемонстрируйте, как система выбирает и применяет правила.
3. Постройте семантическую сеть для предметной области "Автомобили". Укажите узлы и связи (IS-A, HAS, PART-OF). Опишите графически или в текстовом виде.
4. Создайте фреймовую модель на основе семантической сети из задания 3. Представьте фреймы в виде структур с использованием слотов и значений.
5. Рассчитайте вероятность наступления события с помощью наивного Байесовского классификатора. Используйте простой набор данных (например: прогноз погоды — будет ли дождь?).
6. Напишите код для поиска кратчайшего пути с помощью алгоритма A\*. \* Используйте граф с 6–8 узлами и заданными весами рёбер и эвристикой.
7. Создайте онтологию для предметной области "Птицы" с использованием RDF/OWL (в текстовом виде). Определите не менее 3 классов, 2 свойств и 2 подклассов.
8. Примените нечеткую логику для оценки "качества обслуживания" в ресторане. Определите лингвистические переменные ("низкое", "среднее", "высокое"), функции принадлежности и правило вывода.
9. Опишите простого рационального агента для задачи "поиск выхода из лабиринта". Опишите его перцепты, возможные действия и функцию рациональности.

***Перечень компетенций (части компетенции), проверяемых оценочным средством LLM-4, ML-1, O-1, FC-3.***

#### ***Вопросы на анализ и сравнение***

1. Сравните продукционные и фреймовые системы: особенности, сильные и слабые стороны, сферы применения.
2. Почему онтологии предпочтительнее словарей и таксономий при построении интеллектуальных систем?
3. Какие преимущества и недостатки у Prolog по сравнению с императивными языками при реализации ИИ-систем?
4. Чем отличается представление знаний в семантических сетях и в логических системах? Какой подход более удобен для машинного вывода?

***Перечень компетенций (части компетенции), проверяемых оценочным средством LLM-4, ML-1, O-1, FC-3.***

## **4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

### **Методические рекомендации, определяющие процедуры оценивания на экзамене:**

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является экзамен. Студенты обязаны сдать экзамен в соответствии с расписанием и учебным планом.

ФОС промежуточной аттестации состоит из вопросов к экзамену и результатов текущего контроля.

Экзамен по дисциплине преследует цель оценить работу студента за курс, получение теоретических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Форма проведения экзамена: устно.

Экзаменатору предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи экзамена заносится преподавателем в экзаменационную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и ответов на вопросы экзамена.

### **Критерии оценки:**

#### **1. Оценка ответов на экзаменационные вопросы (30% итоговой оценки)**

##### **Отлично (5)**

Полные, развернутые ответы с демонстрацией глубокого понимания темы.

Использование примеров, формул, корректных терминов.

Умение анализировать и сравнивать методы (например, производственные системы vs. фреймы).

**% выполнения:** 90–100% (допускаются незначительные неточности).

##### **Хорошо (4)**

Ответы содержат основные идеи, но без углубленного анализа.

Возможны небольшие ошибки в деталях или формулировках.

**% выполнения:** 75–89%.

##### **Удовлетворительно (3)**

Ответы поверхностные, с существенными пробелами.

Отсутствие примеров или некорректное применение терминов.

**% выполнения:** 60–74%.

##### **Неудовлетворительно (2)**

Отсутствие понимания ключевых концепций.

Грубые ошибки или неспособность ответить на большую часть вопросов.

**% выполнения:** <60%.

#### **2. Оценка выполнения проектов и лабораторных работ (60% итоговой оценки)**

##### **Отлично (5)**

Полное выполнение всех этапов проектов и лабораторных работ, включая интеграцию LLM. Создание антагонистической игры с элементами обучения с подкреплением. Создание агента Robocode с обучением с подкреплением.

Четкая документация кода и анализ результатов.

**% выполнения:** 90–100%.

**Хорошо (4)**

Выполнены основные задачи лабораторных работ, но в проектах имеются существенные недочеты. Интеграция LLM не выполнена или работает в большинстве случаев некорректно.

**% выполнения:** 75–89%.

**Удовлетворительно (3)**

Решены базовые задачи лабораторных работ, проекты выполнены с критическими ошибками.

Низкое качество кода или отсутствие анализа.

**% выполнения:** 60–74%.

**Неудовлетворительно (2)**

Невыполнение ключевых этапов.

Код нерабочий или отсутствует.

**% выполнения:** <60%.

**3. Оценка тестовых вопросов (10% итоговой оценки)**

**Отлично (5)**

13–15 правильных ответов (87–100%).

Демонстрация уверенного владения терминологией и методами.

**Хорошо (4)**

10–12 правильных ответов (67–80%).

Незначительные ошибки.

**Удовлетворительно (3)**

7–9 правильных ответов (47–60%).

Путаница в базовых концепциях.

**Неудовлетворительно (2)**

Менее 7 правильных ответов (<47%).

Неспособность отличить ключевые модели представления знаний.

*Итоговая оценка (суммарно)*

Оценка	Экзамен (30%)	Практика (60%)	Тест (10%)	Общий %
Отлично (5)	90–100%	90–100%	87–100%	≥85%
Хорошо (4)	75–89%	75–89%	67–80%	70–84%
Удовлетворительно (3)	60–74%	60–74%	47–60%	55–69%
Неудовлетворительно (2)	<60%	<60%	<47%	<55%

Для допуска к экзамену необходимо выполнить **все лабораторные работы на минимум "удовлетворительно"**.

"Отлично" требует высоких результатов во всех компонентах (особенно в практических заданиях).

**Практические кейсы** оцениваются по:

- Корректности кода.
- Достижению метрик.
- Качеству отчета (анализ ошибок, визуализация).

### **Методические рекомендации, определяющие процедуры оценивания лабораторных работ:**

Процедура оценивания лабораторных работ проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

По каждой лабораторной работе оформляется отчет. Отчеты сдаются на проверку руководителю в течение курса по мере их выполнения, и защищаются студентами в установленном порядке.

При защите отчета студенту могут быть заданы вопросы и дополнительные задания по сути лабораторной работы, в том числе из списка контрольных вопросов к данной лабораторной работе. При неудовлетворительной оценке знаний студента по теме данного отчета, студент возвращается к повторному изучению соответствующих материалов, после чего допускается к повторной защите. Неудовлетворительно выполненный отчет также возвращается на доработку.

Отчет должен содержать заголовок, тему лабораторной работы, цель, задание, индивидуальную тему, описание хода выполнения работы, необходимые прикладные материалы (схемы, макеты документов и т.п.), в соответствии с требованиями к содержанию, и выводы по работе.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

### **4.3. Методические указания по организации вычислительной инфраструктуры**

#### **Условия применения:**

- Курс рассчитан на студентов 4-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab, Google Colab или Yandex DataSphere, JupyterHub, Hugging Face).
- Разработаны лабораторные работы;
- Инфраструктура для приёма задач (gitlab, CI/CD) согласована с лабораторными работами.

#### **Цели, задачи и ожидаемые результаты**

Цели организации вычислительной инфраструктуры:

- дать начальное представление о работе в IT инфраструктуре (приучить пользоваться гитом, jupyter-ноутбуками).

Задачи преподавателя:

- Организация регистрации студентов в Google Colab и Yandex DataSphere
- Создание учетных записей студентов в gitlab вуза;
- Настройка GitLab Runner для автоматического тестирования кода.
- Разработка шаблонного репозитория для лабораторных работ с предустановленными зависимостями (PyTorch, Keras, TensorFlow, Hugging Face Transformers).
- Написание автотестов для проверки корректности выполнения заданий (например, fine-tuning моделей).
- Визуализация результатов тестирования через HTML-отчеты.
- Подготовка инструкций по работе с Git и облачными ресурсами.

Ожидаемые результаты студентов:

- представление о работе в IT инфраструктуре (гит, нейминг).
- Навыки запуска и тестирования глубоких нейросетей в облачных средах.
- Понимание CI/CD-процессов в контексте разработки глубоких нейросетей.

### Порядок реализации

**Задача №1:** Организация регистрации студентов в Google Colab и Yandex DataSphere

**Задача №2:** Создание учетных записей студентов в gitlab вуза

**Задача №3:** Настройка GitLab Runner:

Для автоматического тестирования кода используется Docker-образ с предустановленными библиотеками (PyTorch, Keras, TensorFlow, Hugging Face Transformers).

Для выполнения CI/CD пайплайна был настроен GitLab Runner на удалённой виртуальной машине с ОС Ubuntu 24.04.

Последовательность настройки включала следующие шаги:

- Настройка системы – установка необходимых компонентов, таких, как Docker.
- Установка GitLab Runner по официальной инструкции.
- Регистрация Runner для частного сервера GitLab.

**Задача №4: Шаблонный репозиторий:**

Включает:

.gitlab-ci.yml для CI/CD.

Скрипты для предобработки текста и обучения моделей.

Примеры кода для работы с BERT, GPT и другими архитектурами.

**Задача №5: Автотесты:**

Проверяют корректность fine-tuning моделей (например, ассигасу на тестовом датасете).

**Задача №6: Визуализация результатов:**

Генерация HTML-отчетов с результатами тестирования, включая метрики качества моделей.

**Порядок проверки корректности:**

- Наличие Git-репозитория у всех студентов.
  - Шаблонный репозиторий с подключенными автотестами.
  - Инструкция по работе с Git и CI/CD в формате README.md.

Вся структура максимально адаптирована для копирования студентами и минимизации порога входа при выполнении лабораторных

## 4.4. Методические указания по организации лабораторных работ

**Условия применения:**

- Курс рассчитан на студентов 4-го года обучения.

- Наличие доступа к вычислительным ресурсам (GitLab) и к GPU/CPU (Kaggle, локальные серверы).
- Разработана инфраструктура для приёма задач (Gitlab, CI/CD) и согласована с лабораторными работами и настроена на всех студентов образовательной программы;
- Использование открытых датасетов и библиотек.

### **Цели, задачи и ожидаемые результаты**

Цели организации лабораторных работ:

- Закрепление теоретических знаний на практике.
- Развитие навыков разработки систем искусственного интеллекта.
- Подготовка к решению реальных задач в индустрии.

Задачи преподавателя:

- Обеспечить студентов структурированными лабораторными работами.
- Предоставить доступ к необходимым вычислительным ресурсам.
- Организовать проверку и обратную связь по выполненным работам.

Ожидаемые результаты студентов:

- Умение применять систем искусственного интеллекта на практике.
- Владение библиотеками и фреймворками (PyTorch, TensorFlow, Keras, Hugging Face Transformers).

## **4.5. Методические указания по организации проектной деятельности студентов**

### **Условия применения:**

Курс рассчитан на студентов 4-го года обучения,

Общее время на проект – не более 16 часов на каждого студента.

Имеется доступ к кейсам индустриальных партнеров.

### **Цели, задачи и ожидаемые результаты**

#### **Цели организации вычислительной инфраструктуры:**

дать начальное представление о реальных задачах, решаемых с помощью систем искусственного интеллекта и возникающих проблемах.

#### **Задачи преподавателя:**

- сбор кейсов индустриальных партнеров;
- сбор кейсов преподавателей практиков и лабораторий в вузе;
- формирование ТЗ на зачетный проект на основе кейсов;
- разработка системы учёта результатов проекта в итоговой оценке зачета

#### **Ожидаемые результаты студентов:**

начальное представление о реальных задачах, решаемых с помощью с помощью систем искусственного интеллекта и возникающих проблемах.

#### **Задача №1: Сбор кейсов индустриальных партнеров**

##### **1. Кредитный скоринг для микрофинансовых организаций**

**Описание:** Разработать систему автоматической оценки кредитоспособности заемщиков на основе альтернативных данных (история мобильных платежей, активность в соцсетях, поведенческие паттерны). Традиционные кредитные истории часто отсутствуют у целевой аудитории МФО.

**Цель:** Снизить уровень просрочек по микрокредитам за счет более точной оценки рисков и расширения клиентской базы.

#### **Ожидаемые результаты:**

- Модель скоринга с AUC-ROC не менее 0.85
- Автоматизированная система принятия решений по кредитным заявкам
- Снижение уровня дефолтов на 15-20%

##### **2. Алгоритмический трейдинг для хедж-фонда**

**Описание:** Создать систему автоматической торговли на финансовых рынках, использующую ensemble методов машинного обучения для прогнозирования движения

цен. Система должна учитывать множественные факторы: технические индикаторы, новостной фон, макроэкономические данные.

**Цель:** Максимизировать доходность торговых операций при заданном уровне риска.

**Ожидаемые результаты:**

- Торговая стратегия с Sharpe Ratio > 2.0
- Система управления рисками в реальном времени
- Панель мониторинга эффективности стратегии

**Задача №2: Кейсы преподавателей-практиков и лабораторий в вузе**

### **1. Прогнозирование оттока клиентов банка**

**Постановка задачи:** Банк хочет снизить отток клиентов премиум-сегмента. Необходимо разработать модель, предсказывающую вероятность ухода клиента в ближайшие 3 месяца.

**Требования:**

- Собрать и очистить данные о транзакциях, продуктах и взаимодействиях клиентов
- Достичь precision не менее 0.75 для класса "уходящие клиенты"
- Разработать систему рекомендаций по удержанию клиентов

### **2. Обнаружение мошенничества с кредитными заявками**

**Постановка задачи:** Создать систему выявления поддельных документов и мошеннических схем при оформлении кредитов.

**Требования:**

- Реализовать пайплайн обработки структурированных и неструктурированных данных
- Обучить модель аномалий на исторических данных
- Интегрировать решение с CRM-системой банка
- Обеспечить F2-score не менее 0.8

## **5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)**

### **5.1 Основная литература:**

1. Стюарт Рассел, Питер Норвиг. Искусственный интеллект: современный подход, 4-е изд. Издательский дом «Вильямс», 2021. - 704 с. ISBN 978-5-907365-25-4 (рус.)
1. Бессмертный, И. А. Системы искусственного интеллекта : учебник для вузов / И. А. Бессмертный. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 164 с. — (Высшее образование). — ISBN 978-5-534-18416-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561602> (дата обращения: 21.07.2025).
2. Воронов, М. В. Системы искусственного интеллекта : учебник и практикум для вузов / М. В. Воронов, В. И. Пименов, И. А. Небаев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 268 с. — (Высшее образование). — ISBN 978-5-534-17032-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567794> (дата обращения: 21.07.2025)..

### **5.2 Дополнительная литература:**

1. Программирование на JAVA [Текст] : учебное пособие / С. Г. Сеница, А. В. Уварова ; М-во образования и науки Рос. Федерации, Кубанский гос. ун-т. - Краснодар : [Кубанский государственный университет], 2016. - 117 с. : ил. - Библиогр.: с. 116. - ISBN 978-5-8209-1215-3

2. Стандарты оформления исходного кода программ и современные интегрированные среды разработки программного обеспечения: учеб.-метод.пособие. Ю.В. Кольцов, А.В.Уварова, С.Г.Синица [и др.] – Краснодар: Кубанский гос.ун-т, 2017
3. Смолин Д. В. Введение в искусственный интеллект: конспект лекций. Учебное пособие. М.: Физматлит, 2007. - 291 с. ISBN: 978-5-9221-0862-1. Электронная библиотечная система «Университетская библиотека ONLINE». URL: <http://www.biblioclub.ru/book/127785/>
4. Харахан О. Г. Системы искусственного интеллекта. Практикум для проведения лабораторных работ. Ч. 1. Учебное пособие для вузов М.: Московский государственный горный университет, 2006. - 80 с. ISBN: 5-7418-0425-X. Электронная библиотечная система «Университетская библиотека ONLINE». URL: <http://www.biblioclub.ru/book/83722/>
5. Джонс М. Т. Программирование искусственного интеллекта в приложениях. М.: ДМК Пресс, 2011. - 313 с. ISBN: 978-5-94074-746-8. Электронная библиотечная система «Университетская библиотека ONLINE». URL: [http://www.biblioclub.ru/131005\\_Programmirovanie\\_iskusstvennogo\\_intellekta\\_v\\_prilozheniyakh.html](http://www.biblioclub.ru/131005_Programmirovanie_iskusstvennogo_intellekta_v_prilozheniyakh.html)

### 5.3. Периодические издания:

1. Базы данных компании «Ист Вью» <http://dlib.eastview.com>
2. Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

### 5.4. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

*Электронно-библиотечные системы (ЭБС):*

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» <http://www.biblioclub.ru/>
3. ЭБС «BOOK.ru» <https://www.book.ru>
4. ЭБС «ZNANIUM.COM» [www.znanium.com](http://www.znanium.com)
5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

*Профессиональные базы данных*

1. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>
3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prlib.ru/>
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <http://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>
10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>
12. Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>
13. Springer Materials: <http://materials.springer.com/>
14. Nano Database: <https://nano.nature.com/>
15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
16. "Лекториум ТВ" <http://www.lektorium.tv/>
17. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

### *Бесплатные образовательные ресурсы*

1. Jupyter Notebook – интерактивные вычисления
2. Visual Studio Code – редактор кода с поддержкой Python
3. Google Scholar/arXiv – доступ к научным публикациям

### *Ресурсы свободного доступа*

1. КиберЛенинка <http://cyberleninka.ru/>;
2. Американская патентная база данных <http://www.uspto.gov/patft/>
3. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
4. Федеральный портал "Российское образование" <http://www.edu.ru/>;
5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
6. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
7. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
8. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
9. Служба тематических толковых словарей <http://www.glossary.ru/>;
10. Словари и энциклопедии <http://dic.academic.ru/>;
11. Образовательный портал "Учеба" <http://www.ucheba.com/>;
12. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы [http://xn--273--84d1f.xn--plai/voprosy\\_i\\_otvety](http://xn--273--84d1f.xn--plai/voprosy_i_otvety)

### *Собственные электронные образовательные и информационные ресурсы КубГУ*

1. Электронный каталог Научной библиотеки КубГУ <http://megapro.kubsu.ru/MegaPro/Web>
2. Электронная библиотека трудов ученых КубГУ <http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>
3. Среда модульного динамического обучения <http://moodle.kubsu.ru>
4. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://infoneeds.kubsu.ru/>
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru>;
6. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" <http://icdau.kubsu.ru/>

## **6. Методические указания для обучающихся по освоению дисциплины (модуля)**

По курсу предусмотрено проведение лекционных занятий, на которых дается систематизированный материал по основным направлениям искусственного интеллекта. В ходе лекций рассматриваются ключевые концепции: от представления знаний (семантические сети, фреймы, онтологии) до современных подходов, таких как большие языковые модели, интеллектуальные агенты и системы управления знаниями. Особое внимание уделяется практическим аспектам: разбираются примеры применения ИИ в реальных задачах (экспертные системы, автоматическое планирование), обсуждаются типичные проблемы и ограничения при внедрении ИИ-решений. После каждой лекции рекомендуется выполнение практических заданий для закрепления ключевых понятий и методов.

Лабораторные занятия курса посвящены практическому освоению методов построения систем искусственного интеллекта. На занятиях студенты реализуют и анализируют модели представления знаний, работают с продукционными системами, фреймами и семантическими сетями, а также исследуют работу алгоритмов поиска (A\*, BFS, DFS), работают с нечеткой логикой. Используются реальные сценарии, такие как построение онтологий, моделирование агентов, реализация простых экспертных систем. Студенты применяют такие языки, как Prolog, RDF/OWL, Java, Python. Используют знания, полученные в других курсах, для интеграции больших языковых моделей с ИИ на базе структурированных знаний. После каждого лабораторного занятия предлагаются задания для самостоятельного выполнения — от построения семантических сетей до применения алгоритмов вывода.

При самостоятельной работе студентам необходимо изучать рекомендованную литературу (учебники, научные статьи, документацию библиотек) для глубокого понимания теоретических основ и современных подходов в области искусственного интеллекта. Выполняя проектные задания, студент должен уметь: формулировать задачу (например, моделирование знаний, поиск решения, построение агента); подбирать и адаптировать подходящие методы (от классических логических систем до вероятностных и нейросетевых подходов); реализовывать и оценивать эффективность выбранных решений. Особое внимание уделяется навыкам анализа и сравнения методов, а также критическому осмыслению их применимости в конкретной предметной области.

Важнейшим компонентом курса является самостоятельная проектная работа, в ходе которой студент разрабатывает законченное решение на основе ИИ-технологий — например, антагонистическую игру, интеллектуального агента, систему на основе нечеткой логики. Такой проект позволяет закрепить навыки проектирования, реализации и оценки систем искусственного интеллекта, а также научиться интегрировать разные методы и подходы в единое решение.

Для студентов с ограниченными возможностями здоровья предусмотрены дополнительные индивидуальные консультации, на которых преподаватель подробно разъясняет сложные аспекты дисциплины, помогает адаптировать практические задания и обеспечивает специальные условия для освоения методов работы с системами искусственного интеллекта. Индивидуальный подход позволяет таким студентам полноценно участвовать в учебном процессе и достигать требуемых результатов обучения.

## **7. Материально-техническое обеспечение по дисциплине (модулю)**

### **7.1 Перечень информационно-коммуникационных технологий**

#### 1. Облачные платформы и сервисы

AWS/GCP/Azure/YandexCloud/cloud.ru – облачные вычисления для использования

LLM

#### 2. Системы управления версиями и коллаборации

Git/GitHub/GitLab – контроль версий кода и совместная разработка

#### 4. Система управления обучением

Moodle – сдача работ

### **7.2 Перечень лицензионного и свободно распространяемого программного обеспечения**

#### 1. Свободное ПО (Open Source)

Инструменты для разработки ИИ-систем:

**SWI Prolog** – реализация логических и экспертных систем

**Java** – язык программирования общего назначения с мощной экосистемой, активно используемый в разработке ИИ-систем, особенно в корпоративной среде и образовании  
**Python** – основной язык программирования для ИИ-приложений

Библиотеки для работы с знаниями и логикой:

networkx – построение и анализ семантических сетей

rdflib – работа с RDF и онтологиями

OWLready2 – загрузка и модификация онтологий в формате OWL

Системы управления знаниями:

**Protégé** – инструмент для создания онтологий

Apache Jena – платформа для работы с RDF, RDFS и OWL

Neo4j – графовая СУБД для представления знаний в виде графов

Инструменты для нечеткой логики и агентов:

jFuzzyLogic – библиотека нечеткой логики на Java

**Robocode** – платформа симулятор боя роботов для разработки и тестирования интеллектуальных агентов

simpful – реализация нечетких систем на Python

APLpy / Jason – разработка интеллектуальных агентов

Визуализация и анализ:

Graphviz – построение графов и диаграмм (семантические сети, деревья вывода)

Matplotlib / Plotly – визуализация результатов работы ИИ-моделей

Streamlit / Gradio – создание веб-интерфейсов для экспертных и интеллектуальных систем

СУБД и хранение знаний:

SQLite / PostgreSQL – хранение структурированных данных и знаний

MongoDB – хранение неструктурированных данных и графов знаний

Neo4J – обработка RDF-данных

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
6.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.