

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ
по дисциплине «Интеллектуальные агенты и многоагентные системы»
для студентов направления подготовки
09.03.01 Информатика и вычислительная техника
Направленность (профиль)
«Программное обеспечение средств вычислительной техники и автоматизированных систем»

Содержание

Введение

Лабораторная работа №1 Знакомство со средой NetLogo и базовыми элементами агентного моделирования.

Лабораторная работа №2 Реализация агентной модели в NetLogo: создание агентов, среды и базовых правил поведения

Лабораторная работа №3 Анализ динамики агентной модели и проведение вычислительных экспериментов в NetLogo

Лабораторная работа №4 Создание LLM-агента с использованием LangChain: цепочки обработки запросов, инструменты и память

Лабораторная работа №5 Реализация многошаговых задач LLM-агента с использованием LangChain

Лабораторная работа №6 Проектирование stateful-агента с использованием LangGraph и графовой модели выполнения

Лабораторная работа №7 Реализация управления состояниями и потоками выполнения в LangGraph

Лабораторная работа №8 Разработка системы взаимодействующих LLM-агентов и организация распределения задач

Лабораторная работа №9. Реализация механизмов координации и обмена сообщениями между агентами

Лабораторная работа №10. Создание прикладного AI-агента с использованием платформы Agno

Лабораторная работа №11. Интеграция инструментов и управление рабочими процессами AI-агента в Agno

Лабораторная работа №12. Разработка прототипа многоагентной системы на основе изученных инструментов

Заключение

Учебно-методическое и информационное обеспечение дисциплины

ВВЕДЕНИЕ

1. Цель и задачи освоения дисциплины (модуля)

Цель дисциплины – формирование, согласно образовательной программы, компетенций ОПК-1, ОПК-2, ПК-2 обучающегося.

Для достижения этой цели в процессе изучения дисциплины решаются следующие задачи: ознакомление с основными понятиями, архитектурами и направлениями развития интеллектуальных агентов и многоагентных систем; изучение моделей интеллектуальных агентов, их структуры, механизмов восприятия, принятия решений и взаимодействия; освоение принципов построения и анализа многоагентных систем, включая кооперацию, конкуренцию и распределённое принятие решений; изучение методов агентного и имитационного моделирования сложных систем; получение практических навыков построения и анализа агентных моделей в среде NetLogo; освоение современных подходов к разработке программных AI-агентов с использованием библиотек LangChain, LangGraph и Agno; знакомство с возможностями использования локальных больших языковых моделей и LLM-агентов, включая применение платформы Ollama; развитие навыков анализа поведения многоагентных систем и интерпретации результатов моделирования; формирование умений применять агентные и многоагентные подходы при решении задач интеллектуального управления и координации в сложных распределённых системах; развитие навыков проектирования и представления решений в области интеллектуальных и многоагентных систем.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина относится к факультативным дисциплинам.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ОПК-1. Способен применять естественнонаучные и инженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ИД-2 _{ОПК-1} решает стандартные профессиональные задачи с применением естественнонаучных и инженерных знаний, методов математического анализа и моделирования	Применение естественнонаучных и инженерных знаний при решении задач, связанных с интеллектуальными агентами и многоагентными системами; применение методов математического анализа при исследовании поведения агентов и их взаимодействия; применение методов моделирования при описании, анализе и решении стандартных профессиональных задач в

		области интеллектуальных и распределённых систем.
	ИД-3 _{ОПК-1} осуществляет теоретические и экспериментальные исследования объектов профессиональной деятельности.	Понимание методов теоретического исследования интеллектуальных агентов и многоагентных систем; понимание принципов экспериментального исследования и имитационного моделирования сложных систем; понимание методов анализа и интерпретации результатов моделирования и вычислительных экспериментов.
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ИД-2 _{ОПК-2} использует современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	Понимание принципов работы современных информационных технологий и программных средств, применяемых при разработке интеллектуальных агентов и многоагентных систем; понимание архитектур программных средств для моделирования и разработки интеллектуальных систем; понимание принципов использования современных программных средств и инструментальных сред при решении профессиональных задач в области интеллектуальных и распределённых систем.
ПК-2. Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	ИД-1 _{ПК-2} реализует методы планирования проектных работ; применяет методы классического системного анализа; использует теорию управления бизнес-процессами, методы концептуального проектирования, методы оценки качества программных систем; осуществляет решение задач	Понимание методов классического системного анализа; понимание теории управления бизнес-процессами; понимание методов концептуального проектирования; понимание методов оценки качества

	оптимизации при разработке бизнес-требований к системе.	программных систем.
	ИД-3 _{ПК-2} составляет графики контрольных мероприятий; разрабатывает бизнес-требования к системе; способен определять ключевые свойства и ограничения системы; выделяет подсистемы системы	Понимание принципов формулирования требований к интеллектуальным и распределённым системам; понимание методов выделения ключевых свойств и ограничений сложных программных систем; понимание принципов декомпозиции системы на подсистемы при проектировании интеллектуальных и многоагентных систем; понимание методов планирования и контроля этапов разработки программных систем.
	ИД-11 _{ПК-2} Реализует конструкции распределенного и параллельного программирования	Понимание принципов распределённого и параллельного программирования; понимание моделей взаимодействия компонентов в распределённых программных системах; понимание методов организации параллельного выполнения вычислений и координации взаимодействия программных компонентов в интеллектуальных и многоагентных системах.

4. Объем учебной дисциплины (модуля) и формы контроля *

Объем занятий: всего: 3 з.е. 108 ак.ч.	ОФО, В акад. часах
Контактная работа:	54.00/36.00
Лекции/из них практическая подготовка	18.00/0
Лабораторных работ/из них практическая подготовка	36.00/0
Практических занятий/из них практическая подготовка	
Самостоятельная работа	54.00
Формы контроля	

Экзамен	нет
Зачет	нет
Зачет с оценкой	да
Курсовая работа	нет

5. Важность лабораторной работы студентов

Лабораторная работа является важнейшим элементом освоения дисциплины «Интеллектуальные агенты и многоагентные системы», поскольку позволяет студентам перейти от теоретического понимания принципов построения агентных систем к их практической реализации. Выполнение лабораторных заданий обеспечивает закрепление знаний о моделях интеллектуальных агентов, механизмах взаимодействия агентов и архитектуре многоагентных систем.

В ходе лабораторных работ студенты получают практический опыт разработки и исследования агентных моделей, используют специализированные инструменты и программные средства для моделирования и реализации интеллектуальных агентов. Особое внимание уделяется работе с современными программными библиотеками и фреймворками для создания AI-агентов, включая LangChain, LangGraph и Agno, а также использованию средств агентного моделирования для анализа поведения сложных систем.

Практическая деятельность в рамках лабораторных работ способствует развитию навыков программирования, проектирования распределённых систем и экспериментального исследования поведения агентов. Студенты учатся формулировать задачи моделирования, реализовывать алгоритмы взаимодействия агентов, анализировать результаты экспериментов и делать обоснованные выводы о функционировании многоагентных систем.

Таким образом, лабораторные работы обеспечивают интеграцию теоретических знаний и практических навыков, формируют у студентов способность применять современные методы и инструменты разработки интеллектуальных систем и способствуют подготовке специалистов, способных проектировать и исследовать многоагентные программные системы в профессиональной деятельности.

ЛАБОРАТОРНАЯ РАБОТА №1

Тема: Знакомство со средой NetLogo и базовыми элементами агентного моделирования

Цель работы

Овладение базовыми навыками работы со средой агентного моделирования NetLogo. Изучение основных элементов агентной модели, принципов управления агентами и построения простых моделей поведения агентов в среде.

Теоретическое обоснование

Агентное моделирование является одним из подходов к исследованию сложных систем, в которых поведение всей системы формируется на основе взаимодействия большого количества относительно простых элементов — агентов. Каждый агент обладает собственными характеристиками и правилами поведения, а глобальные свойства системы возникают как результат их взаимодействия.

Агентные модели широко применяются в различных областях науки и техники: экономике, социологии, биологии, экологии, транспортных системах, а также при исследовании

распределённых вычислительных систем. Такой подход позволяет исследовать процессы самоорганизации, распространения информации, коллективного поведения и других сложных явлений.

Одним из наиболее известных инструментов агентного моделирования является среда NetLogo. Она была разработана в Northwestern University и предназначена для создания и анализа агентных моделей различной сложности. NetLogo предоставляет удобный язык программирования и графический интерфейс, позволяющий быстро создавать модели и наблюдать результаты их работы.

В NetLogo используется несколько типов объектов. Основными из них являются агенты, представленные подвижными объектами (turtles), элементы среды (patches), а также связи между агентами (links). Модель выполняется в дискретном времени, а поведение агентов задаётся с помощью программных процедур.

Использование среды NetLogo позволяет на практике изучить принципы агентного моделирования, научиться создавать простые модели и исследовать поведение систем, состоящих из множества взаимодействующих агентов.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Установленная среда агентного моделирования NetLogo

Методические указания по лабораторной работе

Методические указания по лабораторной работе

Перед выполнением работы необходимо установить и запустить среду NetLogo. После запуска программы следует ознакомиться со структурой интерфейса, включающего окно модели, панель управления, редактор кода и элементы управления моделированием.

Необходимо изучить основные типы объектов среды NetLogo:

- patches — элементы среды, представляющие отдельные клетки пространства
- turtles — агенты, способные перемещаться и выполнять действия
- links — связи между агентами

Далее необходимо изучить базовые команды языка NetLogo, используемые для создания агентов, управления их движением и изменения параметров среды. Особое внимание следует уделить процедурам `setup` и `go`, которые обычно используются для инициализации модели и запуска моделирования.

После изучения интерфейса и основных команд необходимо реализовать простую модель поведения агентов и провести наблюдение за результатами её работы.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями

Запрещается установка стороннего программного обеспечения без разрешения преподавателя

Не допускается использование личных или конфиденциальных данных при выполнении лабораторных работ

Типовые задачи

1. Установить и запустить среду NetLogo. Ознакомиться с интерфейсом программы, определить назначение основных элементов интерфейса.
2. Создать простую модель, в которой несколько агентов (turtles) случайным образом перемещаются по среде.
3. Добавить в модель элементы среды (patches) с различными цветами и наблюдать влияние среды на поведение агентов.
4. Реализовать модель, в которой агенты изменяют направление движения при столкновении с границами среды.
5. Добавить в модель параметр, регулирующий количество агентов, и исследовать влияние этого параметра на поведение системы.

Контрольные вопросы

1. Что понимается под агентным моделированием?
 2. Какие задачи могут решаться с помощью агентных моделей?
 3. Что представляет собой среда NetLogo?
 4. Какие основные типы объектов используются в NetLogo?
 5. Чем отличаются объекты turtles и patches?
 6. Как создаются агенты в NetLogo?
 7. Что представляет собой процедура setup?
 8. Для чего используется процедура go?
 9. Какие преимущества имеет агентное моделирование при исследовании сложных систем?
 10. В каких областях науки и техники применяется агентное моделирование?
- 1.

ЛАБОРАТОРНАЯ РАБОТА №2

Тема: Реализация агентной модели в NetLogo: создание агентов, среды и базовых правил поведения

Цель работы

Овладение навыками создания агентных моделей в среде NetLogo. Изучение способов программного задания среды, создания агентов и реализации базовых правил их поведения. Формирование практических навыков разработки простых агентных моделей и анализа поведения системы.

Теоретическое обоснование

Агентное моделирование позволяет исследовать поведение сложных систем через моделирование взаимодействия множества автономных агентов. Каждый агент действует со-

гласно набору правил и взаимодействует с другими агентами и окружающей средой. Коллективное поведение системы формируется как результат локальных взаимодействий.

В агентных моделях важную роль играет среда, в которой существуют агенты. Среда может представлять собой дискретное пространство, разделённое на клетки, каждая из которых обладает определёнными свойствами. В NetLogo такие элементы среды представлены объектами типа `patches`.

Агенты в NetLogo представлены объектами `turtles`. Они обладают координатами, направлением движения и набором переменных, которые могут изменяться в процессе работы модели. Агенты могут перемещаться по среде, взаимодействовать друг с другом, изменять параметры среды и реагировать на её состояние.

Поведение агентов определяется набором правил, которые реализуются в виде процедур на языке NetLogo. Эти правила могут описывать движение агентов, их взаимодействие, изменение состояния среды или реакцию на события.

Разработка агентной модели обычно включает несколько этапов: создание среды, генерацию агентов, определение параметров модели, реализацию правил поведения агентов и запуск процесса моделирования. В процессе выполнения модели исследователь может наблюдать за поведением системы, изменять параметры и анализировать полученные результаты.

Понимание принципов построения агентных моделей и умение реализовать их в среде NetLogo является важной основой для изучения многоагентных систем и анализа сложных распределённых процессов.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Установленная среда агентного моделирования NetLogo

Методические указания по лабораторной работе

Методические указания по лабораторной работе

Перед выполнением лабораторной работы необходимо создать новый проект в среде NetLogo и перейти в редактор кода.

Следует реализовать базовую структуру модели, включающую процедуры `setup` и `go`. Процедура `setup` должна выполнять инициализацию среды и создание агентов, а процедура `go` — описывать поведение агентов во времени.

При создании модели необходимо выполнить следующие действия:

- задать параметры среды
- создать необходимое количество агентов
- определить начальные свойства агентов
- реализовать правила их движения и взаимодействия

Особое внимание следует уделить использованию команд языка NetLogo для управления агентами, включая создание агентов, изменение их параметров и управление их движением.

После реализации модели необходимо провести серию запусков и наблюдать за изменением поведения системы при различных параметрах.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями
Запрещается установка стороннего программного обеспечения без разрешения преподавателя

Не допускается использование личных или конфиденциальных данных при выполнении лабораторных работ

Типовые задачи

1. Создать агентную модель, в которой агенты случайным образом перемещаются по среде. Задать начальное количество агентов и реализовать их движение.
2. Добавить в модель параметр, регулирующий количество агентов, и реализовать интерфейсный элемент управления этим параметром.
3. Реализовать взаимодействие агентов со средой: агенты должны изменять цвет клетки среды, на которую они перемещаются.
4. Реализовать правило изменения направления движения агента при достижении границы пространства.
5. Провести серию экспериментов с различным количеством агентов и проанализировать, как изменяется динамика системы.

Контрольные вопросы

1. Какие этапы включает разработка агентной модели?
2. Какие объекты используются для представления среды в NetLogo?
3. Как создаются агенты в NetLogo?
4. Какие параметры могут быть заданы для агента?
5. Как реализуется поведение агента в модели?
6. Какие команды используются для перемещения агентов?
7. Как можно изменять свойства среды в процессе моделирования?
8. Для чего используются процедуры `setup` и `go`?
9. Какие параметры модели могут влиять на поведение системы?
10. Как можно проводить экспериментальные исследования с использованием агентной модели?

ЛАБОРАТОРНАЯ РАБОТА №3

Тема: Анализ динамики агентной модели и проведение вычислительных экспериментов в NetLogo

Цель работы

Освоение методов анализа поведения агентных моделей и проведения вычислительных экспериментов в среде NetLogo. Формирование навыков изменения параметров модели, наблюдения за динамикой системы и интерпретации результатов моделирования.

Теоретическое обоснование

Агентное моделирование используется не только для построения моделей сложных систем, но и для проведения вычислительных экспериментов, позволяющих исследовать динамику поведения системы при различных условиях. Вычислительный эксперимент представляет собой серию запусков модели с различными значениями параметров, что позволяет изучить влияние этих параметров на поведение системы.

В агентных моделях динамика системы формируется как результат взаимодействия большого количества агентов. Даже простые правила поведения агентов могут приводить к сложным коллективным эффектам, таким как самоорганизация, кластеризация или устойчивые паттерны движения.

Среда NetLogo предоставляет удобные инструменты для анализа динамики моделей. В частности, можно использовать графики, мониторы и элементы управления интерфейса для отслеживания изменений параметров системы во времени. Это позволяет визуально наблюдать за эволюцией модели и анализировать поведение агентов.

Одним из важных этапов исследования является проведение серии экспериментов с различными параметрами модели. Например, можно изменять количество агентов, скорость их движения, параметры взаимодействия или свойства среды. Сравнение результатов таких экспериментов позволяет выявить закономерности поведения системы.

Проведение вычислительных экспериментов и анализ результатов являются важной частью исследования многоагентных систем. Эти методы широко применяются в экономике, биологии, экологии, транспортных системах и других областях, где необходимо изучать поведение сложных распределённых систем.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Установленная среда агентного моделирования NetLogo

Модель, разработанная в предыдущих лабораторных работах

Методические указания по лабораторной работе

Перед началом работы необходимо открыть модель, созданную в предыдущих лабораторных работах, и подготовить её к проведению экспериментов. Для этого следует добавить элементы интерфейса, позволяющие изменять параметры модели, например количество агентов или скорость их движения.

Для анализа динамики модели необходимо использовать графики (plots), которые отображают изменение выбранных параметров во времени. В графике можно отображать количество агентов, среднюю скорость движения или другие характеристики системы.

Необходимо провести серию вычислительных экспериментов, изменяя параметры модели и наблюдая за изменением поведения системы. Результаты экспериментов следует фиксировать и анализировать, обращая внимание на устойчивость поведения системы и возможные закономерности.

В процессе работы рекомендуется провести несколько запусков модели с одинаковыми параметрами, поскольку поведение агентных моделей может зависеть от случайных факторов.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями
Запрещается установка стороннего программного обеспечения без разрешения преподавателя

Не допускается использование личных или конфиденциальных данных при выполнении лабораторных работ

Типовые задачи

1. Добавить в модель элементы интерфейса (ползунки, мониторы, графики), позволяющие изменять параметры модели и наблюдать за динамикой системы.
2. Реализовать график, отображающий изменение количества агентов или другого параметра системы во времени.
3. Провести серию экспериментов, изменяя количество агентов в модели. Проанализировать влияние этого параметра на поведение системы.
4. Изменить параметры движения агентов и исследовать, как это влияет на формирование пространственных структур или динамику системы.
5. Провести несколько запусков модели с одинаковыми параметрами и сравнить полученные результаты. Сделать вывод о роли случайности в агентных моделях.

Контрольные вопросы

1. Что понимается под вычислительным экспериментом?
2. Какие задачи решаются при анализе динамики агентной модели?
3. Какие инструменты NetLogo используются для анализа результатов моделирования?
4. Для чего используются графики в агентных моделях?
5. Какие параметры модели могут быть изменены в ходе эксперимента?
6. Почему результаты работы агентной модели могут отличаться при разных запусках?
7. Что понимается под динамикой системы в агентном моделировании?
8. Какие методы анализа результатов моделирования можно использовать?
9. Почему важно проводить серию экспериментов при исследовании модели?
10. Какие выводы можно сделать на основе анализа результатов агентного моделирования?

Тема: Создание LLM-агента с использованием LangChain: цепочки обработки запросов, инструменты и память

Цель работы

Овладение навыками разработки интеллектуальных программных агентов на основе больших языковых моделей с использованием библиотеки LangChain. Изучение механизмов построения цепочек обработки запросов, использования инструментов и реализации памяти агента.

Теоретическое обоснование

Современные интеллектуальные системы всё чаще используют большие языковые модели (Large Language Models, LLM) в качестве основного компонента принятия решений и обработки информации. Такие модели способны интерпретировать текстовые запросы, генерировать ответы, выполнять логические рассуждения и взаимодействовать с внешними источниками данных.

Однако сами по себе языковые модели не являются полноценными агентами. Для построения интеллектуального агента необходимо дополнить модель механизмами управления задачами, доступом к внешним инструментам и хранением контекста взаимодействия. Для решения этих задач используются специальные программные библиотеки и фреймворки.

Одним из наиболее распространённых инструментов разработки LLM-агентов является библиотека LangChain. Она предоставляет инфраструктуру для построения систем, использующих большие языковые модели, включая механизмы создания цепочек обработки запросов (chains), подключения внешних инструментов (tools) и реализации памяти (memory).

Цепочки обработки запросов позволяют объединять несколько этапов обработки информации. Например, агент может сначала интерпретировать пользовательский запрос, затем извлечь данные из внешнего источника и после этого сформировать итоговый ответ.

Инструменты позволяют агенту взаимодействовать с внешними системами, такими как поисковые сервисы, базы данных, вычислительные модули или API. Благодаря этому агент способен выполнять действия, выходящие за пределы возможностей самой языковой модели.

Память агента используется для сохранения контекста взаимодействия. Это позволяет агенту учитывать предыдущие сообщения пользователя и поддерживать осмысленный диалог.

Использование таких механизмов делает возможным создание интеллектуальных программных агентов, способных выполнять сложные задачи, взаимодействовать с пользователями и интегрироваться с различными информационными системами.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Интерпретатор Python версии 3.9 и выше

Среда разработки: Jupyter Notebook, VS Code или аналогичная

Установленные библиотеки: langchain, langchain-community, openai или локальная модель (например, Ollama)

Доступ к API языковой модели или установленная локальная модель

Методические указания по лабораторной работе

Перед выполнением лабораторной работы необходимо подготовить программную среду. Для этого следует установить библиотеку LangChain и необходимые зависимости.

В рамках лабораторной работы необходимо реализовать простого программного агента, использующего большую языковую модель для обработки пользовательских запросов.

Работа выполняется в несколько этапов:

1. Подключение языковой модели и создание базовой цепочки обработки запросов.
2. Создание цепочки обработки (chain), которая принимает пользовательский запрос и формирует ответ.
3. Добавление инструментов, позволяющих агенту выполнять дополнительные действия (например, математические вычисления или поиск информации).
4. Реализация памяти агента для сохранения истории взаимодействия.
5. Тестирование работы агента с различными пользовательскими запросами.

В процессе работы необходимо проанализировать поведение агента и оценить влияние различных компонентов системы на качество ответов.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями
Запрещается установка стороннего программного обеспечения без разрешения преподавателя

При работе с API необходимо соблюдать правила безопасности и не публиковать ключи доступа

Типовые задачи

1. Установить библиотеку LangChain и необходимые зависимости. Проверить корректность установки.
2. Создать простую цепочку обработки запроса (chain), которая принимает текстовый вопрос пользователя и формирует ответ с использованием языковой модели.
3. Реализовать агентную цепочку, включающую несколько этапов обработки запроса. Например: анализ запроса, извлечение данных и генерация ответа.
4. Подключить к агенту инструмент для выполнения математических вычислений и протестировать его использование.
5. Реализовать память агента, позволяющую сохранять историю диалога и использовать её при формировании ответов.

Контрольные вопросы

1. Что понимается под большим языковым моделем (LLM)?

2. Что представляет собой библиотека LangChain?
3. Что такое цепочка обработки запросов (chain)?
4. Какие задачи решают инструменты (tools) в архитектуре LLM-агента?
5. Для чего используется память агента?
6. Какие преимущества даёт использование цепочек обработки?
7. Какие типы памяти могут использоваться в LLM-агентах?
8. Как агент может взаимодействовать с внешними сервисами?
9. Какие ограничения имеют большие языковые модели?
10. В каких областях могут применяться LLM-агенты?

ЛАБОРАТОРНАЯ РАБОТА №5

Тема: Реализация многошаговых задач LLM-агента с использованием LangChain

Цель работы

Освоение методов реализации многошаговых задач интеллектуального агента на основе больших языковых моделей с использованием библиотеки LangChain. Формирование навыков проектирования цепочек рассуждений, последовательной обработки информации и организации выполнения сложных задач агентом.

Теоретическое обоснование

Современные интеллектуальные агенты на основе больших языковых моделей способны решать не только простые вопросы пользователя, но и выполнять сложные задачи, состоящие из нескольких последовательных этапов. Такие задачи требуют разбиения проблемы на подзадачи, последовательной обработки информации и объединения промежуточных результатов.

Подобные задачи называют многошаговыми задачами (multi-step tasks). В рамках таких задач агент может выполнять различные действия: анализировать запрос пользователя, извлекать дополнительную информацию, обращаться к внешним инструментам, выполнять вычисления и формировать итоговый ответ.

Библиотека LangChain предоставляет средства для построения систем, выполняющих подобные задачи. Одним из ключевых механизмов является организация цепочек обработки запросов (chains), которые позволяют последовательно объединять несколько этапов обработки информации.

Каждый этап цепочки может выполнять отдельную функцию, например: интерпретацию запроса, извлечение данных из внешнего источника, выполнение вычислений или генерацию ответа. Результат каждого этапа передается следующему этапу обработки.

Использование многошаговых цепочек позволяет повысить качество решений, так как агент может структурировать процесс рассуждения и использовать дополнительные источники информации. Такой подход применяется при разработке интеллектуальных помощников, аналитических систем, систем поддержки принятия решений и других интеллектуальных программных систем.

Проектирование многошаговых задач требует понимания структуры задачи, определения последовательности действий агента и выбора соответствующих инструментов обработки информации.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Интерпретатор Python версии 3.9 и выше

Среда разработки: Jupyter Notebook, VS Code или аналогичная

Установленные библиотеки: langchain, langchain-community

Доступ к API языковой модели или установленная локальная модель (например, через Ollama)

Методические указания по лабораторной работе

Перед началом работы необходимо подготовить программную среду и убедиться в корректной установке библиотеки LangChain и выбранной языковой модели.

В рамках лабораторной работы необходимо реализовать интеллектуального агента, способного выполнять задачу, состоящую из нескольких этапов обработки информации.

Работа выполняется в следующей последовательности:

1. Создание базовой цепочки обработки запроса пользователя.
2. Реализация нескольких этапов обработки запроса, выполняющих различные функции.
3. Организация передачи результатов между этапами обработки.
4. Добавление инструментов, которые агент может использовать для выполнения подзадач.
5. Тестирование работы агента на задачах, требующих последовательного рассуждения.

При выполнении лабораторной работы необходимо проанализировать, как изменение структуры цепочки влияет на поведение агента и качество получаемых результатов.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями

Запрещается установка стороннего программного обеспечения без разрешения преподавателя

При использовании API языковых моделей необходимо соблюдать правила безопасности и не распространять ключи доступа

Типовые задачи

1. Реализовать цепочку обработки запроса, состоящую из двух этапов: анализа вопроса пользователя и генерации ответа.

2. Создать многошаговую цепочку обработки, в которой первый этап выполняет извлечение ключевых данных из запроса, а второй этап формирует ответ на основе этих данных.
3. Реализовать цепочку из нескольких этапов, включающую использование инструмента для выполнения вычислений.
4. Создать цепочку, в которой агент последовательно анализирует запрос, ищет дополнительную информацию и формирует итоговый ответ.
5. Провести тестирование разработанного агента на задачах, требующих последовательного рассуждения и выполнения нескольких действий.

Контрольные вопросы

1. Что понимается под многошаговой задачей в интеллектуальных системах?
2. Какие преимущества дает использование многошаговых цепочек обработки запросов?
3. Что представляет собой цепочка обработки (chain) в LangChain?
4. Как осуществляется передача данных между этапами цепочки?
5. Какие задачи может выполнять интеллектуальный агент при решении сложной проблемы?
6. Как можно подключить инструменты к LLM-агенту?
7. Какие этапы может включать процесс обработки пользовательского запроса?
8. Какие преимущества имеют многошаговые системы по сравнению с одношаговыми?
9. Какие ограничения существуют при использовании больших языковых моделей?
10. В каких прикладных задачах используются многошаговые интеллектуальные агенты?

ЛАБОРАТОРНАЯ РАБОТА №6

Тема: Проектирование stateful-агента с использованием LangGraph и графовой модели выполнения

Цель работы

Освоение методов проектирования интеллектуальных агентов с сохранением состояния (stateful-агентов) с использованием библиотеки LangGraph. Формирование навыков построения графовой модели выполнения задач и управления переходами между этапами обработки информации.

Теоретическое обоснование

Современные интеллектуальные системы часто требуют выполнения сложных задач, состоящих из нескольких взаимосвязанных этапов. Для управления такими процессами используется графовая модель выполнения, в которой этапы обработки представляются в виде узлов графа, а переходы между ними — в виде направленных рёбер.

Графовая модель позволяет гибко описывать логику выполнения задачи. В отличие от линейных цепочек обработки, граф может содержать ветвления, циклы и альтернативные сценарии выполнения. Это делает такой подход особенно полезным при разработке ин-

теллектуальных агентов, которые должны принимать решения в зависимости от состояния системы.

Библиотека LangGraph предназначена для построения систем, в которых выполнение задач организовано в виде графа состояний. Каждый узел графа представляет собой отдельный этап обработки данных, а переходы между узлами определяют последовательность выполнения действий.

Важной особенностью таких систем является наличие состояния (state). Состояние представляет собой набор данных, которые сохраняются между этапами выполнения задачи. Это позволяет агенту учитывать результаты предыдущих действий и принимать решения на их основе.

Stateful-агенты используются в различных интеллектуальных системах, включая диалоговые системы, системы поддержки принятия решений, автоматизированные аналитические системы и интеллектуальные помощники. Использование графовой модели выполнения позволяет создавать более устойчивые и управляемые архитектуры интеллектуальных агентов.

Понимание принципов построения графовых моделей и управления состояниями является важным этапом в изучении современных архитектур интеллектуальных и многоагентных систем.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Интерпретатор Python версии 3.9 и выше

Среда разработки: Jupyter Notebook, VS Code или аналогичная

Установленные библиотеки: langgraph, langchain, langchain-community

Доступ к API языковой модели или локальной модели (например, через Ollama)

Методические указания по лабораторной работе

Перед началом работы необходимо установить библиотеку LangGraph и подготовить программную среду для разработки.

В рамках лабораторной работы необходимо реализовать интеллектуального агента, использующего графовую модель выполнения задач. Такой агент должен сохранять состояние между этапами выполнения и принимать решения на основе текущего состояния системы.

Работа выполняется в следующей последовательности:

1. Создание структуры состояния агента, которая будет хранить данные между этапами выполнения.
2. Определение узлов графа, представляющих отдельные этапы обработки информации.
3. Определение переходов между узлами графа.
4. Реализация функций обработки для каждого узла графа.
5. Запуск графовой модели выполнения и анализ поведения агента.

При выполнении лабораторной работы необходимо обратить внимание на структуру состояния агента и корректность передачи данных между этапами выполнения.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями

Запрещается установка стороннего программного обеспечения без разрешения преподавателя

При работе с API языковых моделей необходимо соблюдать правила безопасности и не публиковать ключи доступа

Типовые задачи

1. Установить библиотеку LangGraph и проверить корректность её подключения в среде Python.
2. Реализовать простую графовую модель выполнения, состоящую из нескольких узлов, выполняющих последовательную обработку запроса.
3. Создать структуру состояния агента и реализовать передачу данных между узлами графа.
4. Реализовать граф с ветвлением, в котором выбор следующего узла зависит от результата обработки данных.
5. Реализовать stateful-агента, который сохраняет историю взаимодействия с пользователем и использует её при формировании ответа.

Контрольные вопросы

1. Что понимается под stateful-агентом?
2. Что представляет собой графовая модель выполнения задач?
3. Какие преимущества имеет использование графовой модели по сравнению с линейной цепочкой обработки?
4. Что представляет собой состояние агента?
5. Какие элементы включает граф выполнения в LangGraph?
6. Как осуществляется переход между узлами графа?
7. Какие задачи решает библиотека LangGraph?
8. В каких системах используются stateful-агенты?
9. Какие преимущества имеет сохранение состояния в интеллектуальных системах?
10. Какие ограничения могут возникать при проектировании графовых моделей выполнения?

ЛАБОРАТОРНАЯ РАБОТА №7

Тема: Реализация управления состояниями и потоками выполнения в LangGraph

Цель работы

Освоение методов управления состояниями и потоками выполнения в интеллектуальных агентах на основе графовой модели с использованием библиотеки LangGraph. Формиро-

вание навыков проектирования сложной логики выполнения задач с использованием переходов, условий и обработки состояния системы.

Теоретическое обоснование

При разработке интеллектуальных программных агентов часто возникает необходимость управления сложными процессами обработки информации. Такие процессы могут включать несколько этапов вычислений, принятие решений на основе промежуточных результатов, а также возможность перехода между различными сценариями выполнения.

Для решения подобных задач используется управление потоками выполнения. Поток выполнения определяет последовательность операций, которые выполняет система при обработке задачи. В интеллектуальных системах поток выполнения может изменяться в зависимости от состояния системы или результатов предыдущих вычислений.

В библиотеке LangGraph управление выполнением задач реализуется с использованием графовой модели. В такой модели каждый этап обработки представляется в виде узла графа, а переходы между узлами задают возможные пути выполнения программы.

Особую роль в таких системах играет состояние (state). Состояние представляет собой структуру данных, в которой сохраняется информация, необходимая для продолжения выполнения задачи. Состояние может включать входные данные пользователя, промежуточные результаты вычислений, историю взаимодействия и другие параметры.

Управление состоянием позволяет агенту корректно продолжать выполнение задачи после каждого этапа обработки. Кроме того, состояние может использоваться для выбора дальнейшего пути выполнения в графе.

Использование механизмов управления состояниями и потоками выполнения делает архитектуру интеллектуальных агентов более гибкой и устойчивой. Такой подход широко применяется при разработке диалоговых систем, интеллектуальных помощников, систем поддержки принятия решений и распределённых интеллектуальных систем.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Интерпретатор Python версии 3.9 и выше

Среда разработки: Jupyter Notebook, VS Code или аналогичная

Установленные библиотеки: langgraph, langchain, langchain-community

Доступ к API языковой модели или локальной модели (например, через Ollama)

Методические указания по лабораторной работе

Перед выполнением лабораторной работы необходимо подготовить программную среду и убедиться в корректной установке библиотеки LangGraph.

В рамках лабораторной работы необходимо разработать интеллектуального агента, в котором управление выполнением задач осуществляется с использованием графовой модели.

Работа выполняется в следующей последовательности:

1. Определение структуры состояния системы, используемой для хранения данных между этапами выполнения.
2. Создание узлов графа, реализующих различные этапы обработки информации.
3. Реализация условных переходов между узлами графа на основе состояния системы.
4. Реализация потоков выполнения, включающих альтернативные сценарии обработки задачи.
5. Проведение тестирования системы и анализ корректности управления потоками выполнения.

Особое внимание необходимо уделить корректности обновления состояния системы и логике переходов между узлами графа.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями

Запрещается установка стороннего программного обеспечения без разрешения преподавателя

При работе с API языковых моделей необходимо соблюдать правила безопасности и не распространять ключи доступа

Типовые задачи

1. Реализовать граф выполнения, содержащий несколько узлов обработки запроса пользователя.
2. Реализовать структуру состояния агента и обеспечить передачу данных между этапами обработки.
3. Реализовать условный переход между узлами графа в зависимости от результата анализа пользовательского запроса.
4. Реализовать граф выполнения с несколькими альтернативными путями обработки задачи.
5. Реализовать механизм обновления состояния агента после выполнения каждого этапа обработки.

Контрольные вопросы

1. Что понимается под управлением потоками выполнения?
2. Что представляет собой состояние системы в интеллектуальных агентах?
3. Какие элементы включает граф выполнения задач?
4. Как осуществляется управление переходами между узлами графа?
5. Какие преимущества имеет графовая модель выполнения?
6. Какие задачи решает библиотека LangGraph?
7. Как состояние используется при выборе дальнейшего пути выполнения?
8. В чем отличие линейной и графовой модели выполнения?
9. Какие проблемы могут возникать при управлении состояниями системы?
10. В каких интеллектуальных системах используется управление потоками выполнения?

ЛАБОРАТОРНАЯ РАБОТА №8

Тема: Разработка системы взаимодействующих LLM-агентов и организация распределения задач

Цель работы

Освоение принципов построения многоагентных интеллектуальных систем на основе больших языковых моделей. Формирование навыков разработки системы взаимодействующих LLM-агентов и реализации механизмов распределения задач между агентами.

Теоретическое обоснование

Многоагентные системы представляют собой класс интеллектуальных систем, состоящих из нескольких автономных агентов, взаимодействующих друг с другом для решения общей задачи. Каждый агент обладает собственной областью ответственности, знаниями и набором доступных действий. Совместная работа агентов позволяет решать задачи, которые сложно или невозможно эффективно реализовать в рамках одной централизованной системы.

В современных интеллектуальных системах всё чаще используются агенты на основе больших языковых моделей. Такие агенты способны анализировать текстовую информацию, интерпретировать задачи, генерировать ответы и взаимодействовать с другими программными компонентами. Использование нескольких LLM-агентов позволяет распределять задачи между специализированными агентами, каждый из которых выполняет отдельную функцию.

В многоагентных системах важную роль играет механизм распределения задач. Он определяет, какой агент будет выполнять конкретную подзадачу. Распределение задач может осуществляться различными способами: централизованным управлением, кооперативным взаимодействием или механизмами переговоров между агентами.

В системах взаимодействующих LLM-агентов часто используется архитектура, в которой один агент выполняет роль координатора или планировщика. Такой агент анализирует задачу пользователя, разбивает её на подзадачи и направляет их соответствующим специализированным агентам. Каждый специализированный агент выполняет свою часть работы и возвращает результат координатору.

Использование взаимодействующих агентов позволяет создавать более гибкие и масштабируемые интеллектуальные системы. Такой подход применяется при разработке интеллектуальных ассистентов, аналитических систем, систем автоматизации бизнес-процессов и сложных информационных платформ.

Понимание принципов взаимодействия агентов и методов распределения задач является важным этапом изучения современных многоагентных систем.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Интерпретатор Python версии 3.9 и выше

Среда разработки: Jupyter Notebook, VS Code или аналогичная
Установленные библиотеки: langchain, langgraph, langchain-community
Доступ к API языковой модели или локальной модели (например, через Ollama)

Методические указания по лабораторной работе

Перед выполнением лабораторной работы необходимо подготовить программную среду и установить необходимые библиотеки.

В рамках лабораторной работы необходимо разработать систему из нескольких взаимодействующих LLM-агентов. Один из агентов должен выполнять функции координатора системы, а остальные агенты — специализированные функции обработки информации.

Работа выполняется в следующей последовательности:

1. Определение структуры многоагентной системы и ролей агентов.
2. Реализация агента-координатора, который принимает пользовательский запрос и определяет подзадачи.
3. Реализация специализированных агентов, выполняющих отдельные функции обработки информации.
4. Реализация механизма обмена сообщениями между агентами.
5. Тестирование системы на задачах, требующих распределения задач между агентами.

При выполнении лабораторной работы необходимо проанализировать эффективность распределения задач и корректность взаимодействия агентов.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями
Запрещается установка стороннего программного обеспечения без разрешения преподавателя

При работе с API языковых моделей необходимо соблюдать правила безопасности и не распространять ключи доступа

Типовые задачи

1. Разработать архитектуру многоагентной системы, включающей агента-координатора и несколько специализированных агентов.
2. Реализовать агента-координатора, который принимает пользовательский запрос и определяет, какой агент должен выполнять соответствующую подзадачу.
3. Реализовать двух специализированных агентов, выполняющих различные функции обработки информации (например, поиск информации и выполнение вычислений).
4. Реализовать механизм обмена сообщениями между агентами и передачу результатов выполнения задач.
5. Провести тестирование разработанной системы на задачах, требующих последовательного выполнения нескольких подзадач разными агентами.

Контрольные вопросы

1. Что представляет собой многоагентная система?
2. Какие преимущества имеют многоагентные системы по сравнению с централизованными системами?
3. Какие роли могут выполнять агенты в многоагентной системе?
4. Что понимается под распределением задач между агентами?
5. Какие методы распределения задач используются в многоагентных системах?
6. Как осуществляется взаимодействие между агентами?
7. Какие функции может выполнять агент-координатор?
8. Какие преимущества дает использование нескольких LLM-агентов?
9. Какие проблемы могут возникать при взаимодействии агентов?
10. В каких областях применяются многоагентные системы?

ЛАБОРАТОРНАЯ РАБОТА №9

Тема: Реализация механизмов координации и обмена сообщениями между агентами

Цель работы

Освоение методов координации действий интеллектуальных агентов и реализации механизмов обмена сообщениями в многоагентных системах. Формирование навыков разработки взаимодействующих программных агентов и организации совместного решения задач.

Теоретическое обоснование

Многоагентные системы представляют собой распределённые интеллектуальные системы, состоящие из множества автономных агентов, которые взаимодействуют между собой для достижения общей цели. В таких системах каждый агент может обладать собственной информацией, набором действий и правилами принятия решений.

Одним из ключевых аспектов функционирования многоагентных систем является координация действий агентов. Координация позволяет согласовывать действия различных агентов, избегать конфликтов и обеспечивать эффективное распределение ресурсов и задач.

Для организации взаимодействия между агентами используются механизмы обмена сообщениями. Сообщения могут содержать запросы на выполнение задач, результаты вычислений, информацию о состоянии системы или сигналы управления. Передача сообщений позволяет агентам обмениваться знаниями и принимать решения на основе коллективной информации.

В классических многоагентных системах используются различные протоколы взаимодействия, включая протоколы переговоров, кооперации и распределения задач. В современных системах на основе больших языковых моделей взаимодействие агентов может реализовываться через текстовые сообщения, которые интерпретируются и обрабатываются языковой моделью.

Эффективная организация обмена сообщениями является важным условием построения масштабируемых и устойчивых многоагентных систем. Реализация механизмов координации позволяет агентам совместно решать сложные задачи, требующие распределённой обработки информации.

Изучение механизмов взаимодействия агентов является важным этапом освоения принципов построения современных интеллектуальных и многоагентных систем.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Интерпретатор Python версии 3.9 и выше

Среда разработки: Jupyter Notebook, VS Code или аналогичная

Установленные библиотеки: langchain, langgraph, langchain-community

Доступ к API языковой модели или локальной модели (например, через Ollama)

Методические указания по лабораторной работе

Перед выполнением лабораторной работы необходимо подготовить программную среду и убедиться в корректной установке необходимых библиотек.

В рамках лабораторной работы необходимо разработать систему взаимодействующих агентов, которые обмениваются сообщениями для координации своих действий. Агенты должны передавать друг другу сообщения, содержащие информацию о выполняемых задачах и результатах обработки данных.

Работа выполняется в следующей последовательности:

1. Определение структуры многоагентной системы и ролей агентов.
2. Реализация агентов, выполняющих различные функции обработки информации.
3. Реализация механизма передачи сообщений между агентами.
4. Реализация механизма координации действий агентов при выполнении общей задачи.
5. Проведение тестирования системы и анализ эффективности взаимодействия агентов.

В процессе выполнения лабораторной работы необходимо обратить внимание на структуру сообщений, корректность передачи информации и устойчивость системы при выполнении распределённых задач.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями

Запрещается установка стороннего программного обеспечения без разрешения преподавателя

При работе с API языковых моделей необходимо соблюдать правила безопасности и не распространять ключи доступа

Типовые задачи

1. Реализовать систему из нескольких агентов, выполняющих различные функции обработки информации.
2. Реализовать механизм передачи сообщений между агентами для обмена данными.

3. Реализовать механизм координации действий агентов при выполнении общей задачи.
4. Реализовать протокол взаимодействия агентов, включающий передачу запросов и ответов.
5. Провести тестирование системы на задаче, требующей кооперации нескольких агентов.

Контрольные вопросы

1. Что понимается под координацией в многоагентных системах?
2. Какие методы координации используются в многоагентных системах?
3. Что представляет собой сообщение в системе взаимодействующих агентов?
4. Какие типы сообщений могут использоваться в многоагентных системах?
5. Какие задачи решает обмен сообщениями между агентами?
6. Какие проблемы могут возникать при взаимодействии агентов?
7. Что представляет собой протокол взаимодействия агентов?
8. Какие преимущества имеют распределённые многоагентные системы?
9. Какие методы используются для предотвращения конфликтов между агентами?
10. В каких прикладных системах используются механизмы взаимодействия агентов?

ЛАБОРАТОРНАЯ РАБОТА №10

Тема: Создание прикладного AI-агента с использованием платформы Agno

Цель работы

Освоение принципов разработки прикладных интеллектуальных агентов с использованием платформы Agno. Формирование навыков создания AI-агентов, интеграции языковых моделей, подключения инструментов и реализации прикладных сценариев обработки пользовательских запросов.

Теоретическое обоснование

Современные интеллектуальные системы всё чаще реализуются в виде программных агентов, способных воспринимать входные данные, анализировать их и выполнять действия для достижения поставленной цели. Такие агенты могут использовать различные методы искусственного интеллекта, включая большие языковые модели, инструменты обработки данных и внешние информационные сервисы.

Платформа Agno предназначена для разработки прикладных AI-агентов, способных решать практические задачи обработки информации. Она предоставляет средства для создания интеллектуальных агентов, интеграции языковых моделей, организации взаимодействия с внешними инструментами и построения логики выполнения задач.

В архитектуре AI-агента обычно выделяются несколько ключевых компонентов: модуль обработки пользовательских запросов, языковая модель, механизм управления задачами и набор инструментов, которые агент может использовать при решении задач. Использование таких компонентов позволяет агенту выполнять сложные операции, включая анализ текстовой информации, генерацию ответов, поиск данных и выполнение вычислений.

При разработке прикладных AI-агентов важную роль играет проектирование сценария взаимодействия с пользователем. Агент должен корректно интерпретировать входной запрос, определить последовательность действий и сформировать результат, соответствующий поставленной задаче.

Использование специализированных платформ разработки, таких как Agno, позволяет значительно упростить создание интеллектуальных агентов и ускорить процесс разработки прикладных AI-систем. Такие системы применяются в интеллектуальных помощниках, системах анализа информации, системах поддержки принятия решений и других прикладных областях.

Изучение принципов разработки прикладных AI-агентов является важным этапом подготовки специалистов в области интеллектуальных и многоагентных систем.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Интерпретатор Python версии 3.9 и выше

Среда разработки: Jupyter Notebook, VS Code или аналогичная

Установленная платформа Agno и необходимые зависимости

Доступ к языковой модели через API или локальную модель (например, через Ollama)

Методические указания по лабораторной работе

Перед выполнением лабораторной работы необходимо установить платформу Agno и подготовить программную среду для разработки AI-агента.

В рамках лабораторной работы необходимо разработать прикладного интеллектуального агента, который сможет обрабатывать пользовательские запросы и выполнять заданные действия.

Работа выполняется в следующей последовательности:

1. Установка и настройка платформы Agno.
2. Создание базовой структуры AI-агента.
3. Подключение языковой модели и настройка обработки пользовательских запросов.
4. Добавление инструментов, которые агент может использовать при выполнении задач.
5. Реализация прикладного сценария работы агента и тестирование его поведения.

В процессе выполнения лабораторной работы необходимо проанализировать архитектуру созданного агента и оценить эффективность его работы при обработке различных типов запросов.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями

Запрещается установка стороннего программного обеспечения без разрешения преподава-

теля

При работе с API языковых моделей необходимо соблюдать правила безопасности и не публиковать ключи доступа

Типовые задачи

1. Установить платформу Agno и проверить корректность её работы в среде Python.
2. Создать базового AI-агента, способного принимать текстовый запрос пользователя и формировать ответ с использованием языковой модели.
3. Реализовать прикладной сценарий работы агента, включающий обработку пользовательских запросов и выполнение определённого действия (например, анализ текста или выполнение вычислений).
4. Подключить к агенту внешний инструмент и реализовать его использование при решении задачи.
5. Провести тестирование разработанного AI-агента на различных пользовательских запросах и проанализировать результаты его работы.

Контрольные вопросы

1. Что понимается под прикладным AI-агентом?
2. Какие компоненты включает архитектура интеллектуального агента?
3. Какие задачи решает платформа Agno?
4. Какие преимущества дает использование специализированных платформ разработки AI-агентов?
5. Как осуществляется взаимодействие AI-агента с языковой моделью?
6. Какие функции могут выполнять инструменты в архитектуре агента?
7. Какие этапы включает процесс разработки AI-агента?
8. Какие задачи могут решаться прикладными AI-агентами?
9. Какие ограничения существуют при использовании больших языковых моделей?
10. В каких областях применяются прикладные AI-агенты?

ЛАБОРАТОРНАЯ РАБОТА №11

Тема: Интеграция инструментов и управление рабочими процессами AI-агента в Agno

Цель работы

Освоение методов интеграции внешних инструментов и организации рабочих процессов интеллектуального агента с использованием платформы Agno. Формирование навыков построения прикладных AI-систем, в которых агент выполняет сложные задачи, управляя последовательностью действий и взаимодействием с внешними сервисами.

Теоретическое обоснование

Современные интеллектуальные агенты всё чаще используются для решения сложных прикладных задач, требующих не только обработки текста, но и выполнения различных действий: получения данных из внешних источников, выполнения вычислений, анализа документов и управления рабочими процессами. Для реализации таких возможностей используются механизмы интеграции инструментов и управления процессами выполнения задач.

Инструменты (tools) представляют собой программные компоненты, которые агент может использовать для выполнения конкретных операций. Это могут быть функции обработки данных, обращения к базам данных, API внешних сервисов, поисковые системы или вычислительные модули. Использование инструментов позволяет значительно расширить функциональные возможности интеллектуального агента.

При решении сложных задач агенту часто требуется выполнять несколько последовательных действий. Для этого используются механизмы управления рабочими процессами (workflows). Рабочий процесс представляет собой структуру выполнения задач, в которой определяются этапы обработки данных и последовательность их выполнения.

Платформа Agno предоставляет средства для построения таких систем. Она позволяет интегрировать языковые модели, подключать инструменты и реализовывать логику управления рабочими процессами. Благодаря этому можно создавать интеллектуальных агентов, способных выполнять комплексные задачи, включающие анализ данных, принятие решений и выполнение действий.

Организация рабочих процессов является важной частью разработки прикладных AI-систем. Правильно спроектированный рабочий процесс позволяет обеспечить устойчивость системы, повысить точность выполнения задач и упростить масштабирование интеллектуальных решений.

Изучение методов интеграции инструментов и управления рабочими процессами является важным этапом подготовки специалистов, занимающихся разработкой интеллектуальных и многоагентных систем.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Интерпретатор Python версии 3.9 и выше

Среда разработки: Jupyter Notebook, VS Code или аналогичная

Установленная платформа Agno и необходимые зависимости

Доступ к языковой модели через API или локальную модель (например, через Ollama)

Методические указания по лабораторной работе

Перед выполнением лабораторной работы необходимо установить платформу Agno и подготовить программную среду для разработки интеллектуального агента.

В рамках лабораторной работы необходимо разработать AI-агента, который использует несколько инструментов и выполняет задачу в соответствии с заданным рабочим процессом.

Работа выполняется в следующей последовательности:

1. Определение структуры прикладного AI-агента и набора инструментов, необходимых для решения задачи.
2. Реализация инструментов, которые агент может использовать при обработке пользовательского запроса.
3. Интеграция инструментов в архитектуру AI-агента.

4. Проектирование рабочего процесса выполнения задачи, включающего несколько этапов обработки информации.
5. Тестирование разработанной системы и анализ корректности выполнения рабочего процесса.

В процессе выполнения лабораторной работы необходимо обратить внимание на корректность взаимодействия агента с инструментами и последовательность выполнения этапов обработки задачи.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями
Запрещается установка стороннего программного обеспечения без разрешения преподавателя

При работе с API языковых моделей необходимо соблюдать правила безопасности и не распространять ключи доступа

Типовые задачи

1. Реализовать AI-агента на платформе Agno, который использует несколько инструментов для обработки пользовательских запросов.
2. Реализовать инструмент, выполняющий вычисления или обработку данных, и интегрировать его в систему агента.
3. Реализовать рабочий процесс, состоящий из нескольких этапов обработки информации.
4. Реализовать систему управления последовательностью выполнения задач в рамках рабочего процесса.
5. Провести тестирование AI-агента на задачах, требующих использования нескольких инструментов и последовательного выполнения действий.

Контрольные вопросы

1. Что понимается под инструментом в архитектуре AI-агента?
2. Какие функции могут выполнять инструменты интеллектуального агента?
3. Что представляет собой рабочий процесс (workflow) в интеллектуальных системах?
4. Какие преимущества дает использование рабочих процессов в AI-системах?
5. Какие задачи решает платформа Agno?
6. Как осуществляется интеграция инструментов в архитектуру AI-агента?
7. Какие этапы включает разработка прикладного AI-агента?
8. Какие проблемы могут возникать при управлении рабочими процессами?
9. Какие методы используются для организации последовательности выполнения задач?
10. В каких прикладных системах используются AI-агенты с управлением рабочими процессами?

Тема: Разработка прототипа многоагентной системы на основе изученных инструментов

Цель работы

Закрепление знаний и навыков разработки интеллектуальных и многоагентных систем путём создания прототипа прикладной многоагентной системы с использованием изученных инструментов и технологий. Формирование умений проектировать архитектуру системы, реализовывать взаимодействие агентов и организовывать распределённое выполнение задач.

Теоретическое обоснование

Многоагентные системы являются одним из ключевых направлений развития современных интеллектуальных технологий. Такие системы состоят из нескольких автономных программных агентов, каждый из которых выполняет определённую функцию и взаимодействует с другими агентами для достижения общей цели.

В отличие от централизованных систем, многоагентная архитектура позволяет распределять задачи между специализированными агентами, что повышает гибкость, масштабируемость и устойчивость системы. Каждый агент может обладать собственными знаниями, инструментами и алгоритмами обработки информации.

Современные технологии разработки интеллектуальных агентов включают использование больших языковых моделей, специализированных библиотек для построения AI-агентов и инструментов организации выполнения задач. В рамках курса были рассмотрены такие средства, как LangChain, LangGraph и платформа Agno, позволяющие реализовывать интеллектуальные агенты различной сложности.

При разработке многоагентной системы важно определить роли агентов, структуру их взаимодействия и механизм распределения задач. Один из распространённых подходов заключается в использовании агента-координатора, который принимает запрос пользователя, анализирует его и распределяет подзадачи между специализированными агентами.

В процессе взаимодействия агенты могут обмениваться сообщениями, передавать результаты вычислений и совместно формировать итоговый результат. Эффективность системы во многом зависит от корректной организации взаимодействия агентов и архитектуры всей системы.

Разработка прототипа многоагентной системы позволяет на практике применить полученные знания и продемонстрировать основные принципы построения интеллектуальных распределённых систем.

Аппаратура и материалы

Персональный компьютер с установленной операционной системой Windows, macOS или Linux

Интерпретатор Python версии 3.9 и выше

Среда разработки: Jupyter Notebook, VS Code или аналогичная

Установленные библиотеки: langchain, langgraph, langchain-community

Установленная платформа Agno

Доступ к языковой модели через API или локальной модели (например, через Ollama)

Методические указания по лабораторной работе

Перед выполнением лабораторной работы необходимо подготовить программную среду и убедиться в корректной работе всех ранее изученных инструментов.

В рамках лабораторной работы необходимо разработать прототип многоагентной системы, в которой несколько интеллектуальных агентов совместно решают поставленную задачу. Система должна включать несколько специализированных агентов и механизм их взаимодействия.

Работа выполняется в следующей последовательности:

1. Определение прикладной задачи, которую будет решать многоагентная система.
2. Проектирование архитектуры системы и определение ролей агентов.
3. Реализация агента-координатора, принимающего пользовательский запрос.
4. Реализация специализированных агентов, выполняющих различные функции обработки информации.
5. Реализация механизма взаимодействия агентов и обмена результатами.
6. Тестирование разработанной системы и анализ корректности её работы.

В процессе выполнения лабораторной работы необходимо уделить внимание архитектуре системы, распределению задач между агентами и корректности взаимодействия между компонентами системы.

Требования по технике безопасности

Работа должна выполняться в хорошо освещённом помещении

Продолжительность непрерывной работы за компьютером не должна превышать 2 академических часов

Рабочее место должно быть организовано в соответствии с санитарными требованиями

Запрещается установка стороннего программного обеспечения без разрешения преподавателя

При работе с API языковых моделей необходимо соблюдать правила безопасности и не распространять ключи доступа

Типовые задачи

1. Определить прикладную задачу для разработки многоагентной системы (например, анализ текста, обработка информации или решение вычислительных задач).
2. Спроектировать архитектуру системы, включающую агента-координатора и несколько специализированных агентов.
3. Реализовать взаимодействие между агентами через обмен сообщениями.
4. Реализовать механизм распределения задач между агентами.
5. Провести тестирование разработанной системы и проанализировать эффективность взаимодействия агентов.

Контрольные вопросы

1. Что представляет собой многоагентная система?
2. Какие преимущества имеют многоагентные системы по сравнению с централизованными системами?
3. Какие роли могут выполнять агенты в многоагентной системе?

4. Что понимается под распределением задач между агентами?
5. Какие методы взаимодействия используются в многоагентных системах?
6. Какие технологии могут использоваться при разработке интеллектуальных агентов?
7. Какую роль выполняет агент-координатор?
8. Какие проблемы могут возникать при взаимодействии агентов?
9. Какие методы используются для тестирования многоагентных систем?
10. В каких областях применяются многоагентные системы?

ЗАКЛЮЧЕНИЕ

Лабораторные работы, представленные в рамках дисциплины «Интеллектуальные агенты и многоагентные системы», являются важной частью учебного процесса и направлены на формирование у студентов практических навыков разработки и исследования интеллектуальных программных агентов и многоагентных систем. Практическая направленность лабораторных заданий обеспечивает закрепление теоретических знаний и позволяет студентам применять современные технологии искусственного интеллекта при решении прикладных задач.

В ходе выполнения лабораторных работ студенты последовательно осваивают методы агентного моделирования, проектирования архитектуры интеллектуальных агентов и организации взаимодействия между ними. Особое внимание уделяется использованию современных инструментов разработки AI-агентов, включая среду агентного моделирования NetLogo, библиотеки LangChain и LangGraph, а также платформу Agno для построения прикладных интеллектуальных систем на основе больших языковых моделей.

Комплекс лабораторных работ способствует развитию у студентов навыков алгоритмического и системного мышления, проектирования распределённых программных систем, анализа поведения сложных моделей и проведения вычислительных экспериментов. Практические задания ориентированы на решение задач проектирования интеллектуальных систем, организации взаимодействия агентов и построения многоагентных архитектур.

Выполнение лабораторных работ по дисциплине способствует формированию у обучающихся компетенций ОПК-1, ОПК-2 и ПК-2, связанных с применением методов математического анализа и моделирования, использованием современных информационных технологий и программных средств, а также проектированием и разработкой программных систем средней и высокой сложности. Полученные знания и навыки создают основу для дальнейшего освоения современных технологий искусственного интеллекта и разработки интеллектуальных программных систем в профессиональной деятельности.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература:

Болотский, А. В. Исследование операций и методы оптимизации Электронный ресурс / Болотский А. В., Кочеткова О. А. - Санкт-Петербург : Лань, 2020. - 116 с. - ISBN 978-5-8114-4568-4

Сараев, П. В. Методы машинного обучения Электронный ресурс : Методические указания и задания к лабораторным работам по курсу / П. В.

Сараев. - Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2017. - 48 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397

[illegible]

Дополнительная литература:

Методические рекомендации к практическим занятиям по дисциплине "Теория и практика машинного перевода" : Направление подготовки 45.04.02 Лингвистика. Магистерская программа "Перевод и переводоведение". Квалификация выпускника - магистр. Форма обучения - очная. Учебный план 2015 года. Изучается в 3 семестре.

- Ставрополь : СКФУ, 2015. - 9 с.

Неделько, В. М. Основы статистических методов машинного обучения Электронный ресурс : Учебное пособие / В. М. Неделько. - Новосибирск : Новосибирский государственный технический университет, 2010. - 72 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-7782-1385-2

Рапаков, Г. Г. Методы и алгоритмы машинного обучения при принятии управленческих решений в региональной системе медицинской профилактики (опыт Вологодской области) Электронный ресурс / Рапаков Г. Г., Касимов Р. А. : монография. - Вологда : ВоГУ, 2014. - 143 с. - ISBN 978-5-87851-548-1

Учебно-методическая литература:

Сараев, П. В. Методы машинного обучения Электронный ресурс : Методические указания и задания к лабораторным работам по курсу / П. В. Сараев. - Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2017. - 48 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по организации и проведению самостоятельной работы
по дисциплине «Интеллектуальные агенты и многоагентные системы»
для студентов направления подготовки

09.03.01 Информатика и вычислительная техника

Направленность (профиль)

«Программное обеспечение средств вычислительной техники и автоматизированных систем»

Содержание

Введение

Самостоятельная работа как важнейшая форма учебного процесса

Цели и основные задачи СРС

Виды самостоятельной работы

Организация СРС

Общие рекомендации по организации самостоятельной работы

Учебно-методическое и информационное обеспечение дисциплины

ВВЕДЕНИЕ

1. Цель и задачи освоения дисциплины (модуля)

Цель дисциплины – формирование, согласно образовательной программы, компетенций ОПК-1, ОПК-2, ПК-2 обучающегося.

Для достижения этой цели в процессе изучения дисциплины решаются следующие задачи: ознакомление с основными понятиями, архитектурами и направлениями развития интеллектуальных агентов и многоагентных систем; изучение моделей интеллектуальных агентов, их структуры, механизмов восприятия, принятия решений и взаимодействия; освоение принципов построения и анализа многоагентных систем, включая кооперацию, конкуренцию и распределённое принятие решений; изучение методов агентного и имитационного моделирования сложных систем; получение практических навыков построения и анализа агентных моделей в среде NetLogo; освоение современных подходов к разработке программных AI-агентов с использованием библиотек LangChain, LangGraph и Agno; знакомство с возможностями использования локальных больших языковых моделей и LLM-агентов, включая применение платформы Ollama; развитие навыков анализа поведения многоагентных систем и интерпретации результатов моделирования; формирование умений применять агентные и многоагентные подходы при решении задач интеллектуального управления и координации в сложных распределённых системах; развитие навыков проектирования и представления решений в области интеллектуальных и многоагентных систем.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина относится к факультативным дисциплинам.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ОПК-1. Способен применять естественнонаучные и инженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ИД-2 _{ОПК-1} решает стандартные профессиональные задачи с применением естественнонаучных и инженерных знаний, методов математического анализа и моделирования	Применение естественнонаучных и инженерных знаний при решении задач, связанных с интеллектуальными агентами и многоагентными системами; применение методов математического анализа при исследовании поведения агентов и их взаимодействия; применение методов моделирования при описании, анализе и решении стандартных профессиональных задач в области интеллектуальных и распределённых систем.

	ИД-3 _{ОПК-1} осуществляет теоретические и экспериментальные исследования объектов профессиональной деятельности.	Понимание методов теоретического исследования интеллектуальных агентов и многоагентных систем; понимание принципов экспериментального исследования и имитационного моделирования сложных систем; понимание методов анализа и интерпретации результатов моделирования и вычислительных экспериментов.
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ИД-2 _{ОПК-2} использует современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	Понимание принципов работы современных информационных технологий и программных средств, применяемых при разработке интеллектуальных агентов и многоагентных систем; понимание архитектур программных средств для моделирования и разработки интеллектуальных систем; понимание принципов использования современных программных средств и инструментальных сред при решении профессиональных задач в области интеллектуальных и распределённых систем.
ПК-2. Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	ИД-1 _{ПК-2} реализует методы планирования проектных работ; применяет методы классического системного анализа; использует теорию управления бизнес-процессами, методы концептуального проектирования, методы оценки качества программных систем; осуществляет решение задач оптимизации при разработке бизнес-требований к системе.	Понимание методов классического системного анализа; понимание теории управления бизнес-процессами; понимание методов концептуального проектирования; понимание методов оценки качества программных систем.

	ИД-3 _{ПК-2} составляет графики контрольных мероприятий; разрабатывает бизнес-требования к системе; способен определять ключевые свойства и ограничения системы; выделяет подсистемы системы	Понимание принципов формулирования требований к интеллектуальным и распределённым системам; понимание методов выделения ключевых свойств и ограничений сложных программных систем; понимание принципов декомпозиции системы на подсистемы при проектировании интеллектуальных и многоагентных систем; понимание методов планирования и контроля этапов разработки программных систем.
	ИД-11 _{ПК-2} Реализует конструкции распределенного и параллельного программирования	Понимание принципов распределённого и параллельного программирования; понимание моделей взаимодействия компонентов в распределённых программных системах; понимание методов организации параллельного выполнения вычислений и координации взаимодействия программных компонентов в интеллектуальных и многоагентных системах.

4. Объем учебной дисциплины (модуля) и формы контроля *

Объем занятий: всего: 3 з.е. 108 ак.ч.	ОФО, В акад. часах
Контактная работа:	54.00/36.00
Лекции/из них практическая подготовка	18.00/0
Лабораторных работ/из них практическая подготовка	36.00/0
Практических занятий/из них практическая подготовка	
Самостоятельная работа	54.00
Формы контроля	
Экзамен	нет
Зачет	нет
Зачет с оценкой	да

Курсовая работа	нет
-----------------	-----

* Дисциплина (модуль) предусматривает применение электронного обучения, дистанционных образовательных технологий (если иное не установлено образовательным стандартом)

САМОСТОЯТЕЛЬНАЯ РАБОТА КАК ВАЖНЕЙШАЯ ФОРМА УЧЕБНОГО ПРОЦЕССА

Самостоятельная работа студентов является неотъемлемой частью освоения дисциплины *«Интеллектуальные агенты и многоагентные системы»* и играет ключевую роль в формировании устойчивых и глубоких знаний. Современные методы машинного обучения требуют не только теоретического понимания алгоритмов, но и уверенного владения инструментами, навыков анализа данных, интерпретации результатов и критического мышления.

В процессе самостоятельной работы студент получает возможность:

- закрепить и углубить материал, изученный на лекциях и практических занятиях;
- научиться самостоятельно находить и изучать профессиональные источники информации;
- применять изученные алгоритмы к реальным или приближенным к реальности задачам;
- развивать практические навыки программирования, работы с библиотеками Python и визуализации данных;
- осваивать различные подходы к решению одной и той же задачи, выбирать оптимальные методы, обосновывать сделанный выбор;
- готовиться к итоговому контролю, демонстрируя уверенность в базовых понятиях и алгоритмах.

Кроме того, самостоятельная работа развивает дисциплину, инициативность, умение планировать собственное обучение и адаптироваться к постоянно развивающейся предметной области. Эти навыки особенно важны в сфере искусственного интеллекта, где темпы технологических изменений требуют постоянного самообновления знаний.

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения. Обучение в ВУЗе включает в себя две, практически одинаковые по объему и взаимовлиянию части – процесса обучения и процесса самообучения. Поэтому СРС должна стать эффективной и целенаправленной работой студента.

Концепцией модернизации российского образования определены основные задачи профессионального образования - "подготовка квалифицированного работника соответствующего уровня и профиля, конкурентоспособного на рынке труда, компетентного, ответственного, свободно владеющего своей профессией и ориентированного в смежных областях деятельности, способного к эффективной работе по специальности на уровне мировых стандартов, готового к постоянному профессиональному росту, социальной и профессиональной мобильности".

Решение этих задач невозможно без повышения роли самостоятельной работы студентов над учебным материалом, усиления ответственности преподавателей за

развитие навыков самостоятельной работы, за стимулирование профессионального роста студентов, воспитание творческой активности и инициативы.

К современному специалисту общество предъявляет достаточно широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной финансовой ситуации. Формирование такого умения происходит в течение всего периода обучения через участие студентов в лабораторных занятиях, выполнение контрольных заданий и тестов, написание курсовых и выпускных квалификационных работ. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Формы самостоятельной работы студентов разнообразны. По данной дисциплине «Интеллектуальные агенты и многоагентные системы» предусмотрены следующие:

Использование материала учебно-методического комплекса дисциплины

На первом этапе необходимо ознакомиться с обязательным минимумом содержания основной образовательной программы по изучаемой дисциплине. Далее необходимо изучить рабочую программу дисциплины, в которой рассмотрено содержание тем дисциплины лекционного курса, взаимосвязь тем лекций с лабораторными занятиями, темы и виды самостоятельной работы. По каждому виду самостоятельной работы предусмотрены определённые формы отчетности, представленные в учебной программе по дисциплине «Интеллектуальные агенты и многоагентные системы».

Работа с литературой

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации, которые представлены в учебной программе.

Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.

ЦЕЛИ И ОСНОВНЫЕ ЗАДАЧИ СРС

Цели самостоятельной работы:

- Углубление теоретических знаний, полученных на лекциях;
- Закрепление практических навыков, полученных на семинарах;
- Развитие умений решать прикладные задачи машинного обучения на практике;
- Формирование критического подхода к выбору моделей, параметров и метрик.

Формы самостоятельной работы:

- Изучение рекомендованных источников и материалов лекций;
- Анализ и решение предложенных задач;
- Выполнение индивидуальных практических заданий в Jupyter Notebook;
- Подготовка мини-отчётов и интерпретация полученных результатов;
- Самостоятельное повторение материала и подготовка к тестированию.

Ведущая цель организации и осуществления СРС должна совпадать с целью обучения студента – подготовкой бакалавра с высшим образованием. При организации СРС важным и необходимым условием становятся формирование умения

самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности.

Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

ВИДЫ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

В образовательном процессе высшего профессионального образовательного учреждения выделяется два вида самостоятельной работы – аудиторная, под руководством преподавателя, и внеаудиторная. Тесная взаимосвязь этих видов работ предусматривает дифференциацию и эффективность результатов ее выполнения и зависит от организации, содержания, логики учебного процесса (межпредметных связей, перспективных знаний и др.):

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Основными видами самостоятельной работы студентов без участия преподавателя по данной дисциплине являются:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- подготовка к лабораторным работам, их оформление.

Основными видами самостоятельной работы студентов с участием преподавателей являются:

- текущие консультации;
- прием и защита лабораторных работ (во время проведения л/р).

ОРГАНИЗАЦИЯ СРС

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей дисциплины «Интеллектуальные агенты и многоагентные системы», объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Рекомендуемые темы и задания для самостоятельной проработки:

1. Основы интеллектуальных агентов

- Изучить основные свойства интеллектуальных агентов: автономность, реактивность, проактивность и социальность.
- Задание: сформулировать примеры интеллектуальных агентов из различных областей (робототехника, веб-сервисы, игровые системы) и описать их среду, действия и цели.

2. Типы агентных сред

- Изучить классификацию сред: наблюдаемость, детерминированность, динамичность, эпизодичность.
- Задание: описать несколько различных сред (например, шахматы, дорожное движение, электронная торговля) и классифицировать их по указанным параметрам.

3. Архитектуры интеллектуальных агентов

- Изучить реактивные, делиберативные и гибридные архитектуры агентов.
- Задание: составить сравнительную таблицу архитектур агентов и привести примеры задач, для которых каждая архитектура наиболее подходит.

4. Модель BDI

- Изучить модель убеждений, желаний и намерений (Belief-Desire-Intention).
- Задание: описать пример интеллектуального агента на основе BDI-модели, указав его убеждения, цели и планы действий.

5. Основы агентного моделирования

- Изучить принципы агентного моделирования сложных систем.
- Задание: выбрать систему из реального мира (например, дорожное движение или распространение информации в сети) и описать её в виде агентной модели.

6. Знакомство со средой NetLogo

- Изучить структуру программы NetLogo: интерфейс, процедуры, агенты turtles, patches и links.
- Задание: установить NetLogo и изучить несколько готовых моделей из библиотеки Model Library, описав поведение агентов в выбранной модели.

7. Создание простой агентной модели в NetLogo

- Изучить основные команды управления агентами (ask, forward, right, set).
- Задание: создать простую модель движения агентов по среде и наблюдать изменение их поведения при изменении параметров.

8. Эксперименты с агентной моделью

- Изучить методы проведения вычислительных экспериментов.
- Задание: изменить параметры агентной модели NetLogo и проанализировать, как это влияет на поведение системы.

9. Коллективное поведение агентов

- Изучить механизмы самоорганизации и коллективного поведения.

- Задание: исследовать модель *flocking* (поведение стаи) в NetLogo и объяснить правила, определяющие движение агентов.
10. **Введение в LLM-агентов**
 - Изучить концепцию интеллектуальных агентов на основе больших языковых моделей.
 - Задание: описать архитектуру LLM-агента и его основные компоненты (модель, память, инструменты).
 11. **Основы работы с LangChain**
 - Изучить архитектуру библиотеки LangChain и её основные компоненты.
 - Задание: создать простую цепочку обработки запросов (Chain) для взаимодействия с языковой моделью.
 12. **Использование инструментов в LangChain**
 - Изучить механизм подключения инструментов (tools) к агенту.
 - Задание: реализовать агента, использующего инструмент (например, калькулятор или поиск информации).
 13. **Память агента в LangChain**
 - Изучить механизмы хранения контекста и истории взаимодействия.
 - Задание: реализовать пример агента, который использует память для обработки последовательных запросов.
 14. **Основы LangGraph**
 - Изучить принципы построения графов выполнения действий агента.
 - Задание: создать простой граф выполнения задач агента с несколькими узлами.
 15. **Stateful-агенты в LangGraph**
 - Изучить методы управления состоянием агента.
 - Задание: реализовать агента, который меняет своё поведение в зависимости от состояния.
 16. **Многоагентные системы**
 - Изучить принципы взаимодействия агентов в многоагентных системах.
 - Задание: описать архитектуру системы из нескольких взаимодействующих агентов.
 17. **Координация и распределение задач**
 - Изучить методы распределения задач между агентами.
 - Задание: разработать концепцию системы, где несколько агентов совместно решают одну задачу.
 18. **Платформа Agno**
 - Изучить возможности платформы Agno для разработки AI-агентов.
 - Задание: реализовать простой агентный сценарий с использованием Agno.
 19. **Проектирование многоагентной системы**
 - Изучить принципы архитектурного проектирования агентных систем.
 - Задание: разработать концептуальную архитектуру многоагентной системы для выбранной прикладной задачи.
 20. **Экспериментальное исследование агентной системы**
 - Изучить методы анализа результатов работы агентных систем.
 - Задание: провести эксперимент с разработанной моделью или агентной системой и проанализировать полученные результаты.

Типовые задания для СРС:

Общий подход к оцениванию:

- Каждое задание оценивается по **10-балльной шкале**;
- Общая оценка по блоку самостоятельной работы может рассчитываться как среднее значение или сумма (по усмотрению преподавателя);

- Возможно использование **частичных баллов (0.5, 1 и т.д.)** при частично выполненных пунктах задания;
- При копировании без осмысления, неработающем коде или отсутствующей интерпретации баллы **не начисляются**.

№	Задание	Критерии оценки (макс. 10 баллов)
1	Описать основные свойства интеллектуальных агентов и привести 3 примера их применения	корректность определения — 4; примеры — 4; логичность изложения — 2
2	Сравнить реактивные и делиберативные архитектуры агентов	корректность сравнения — 4; полнота характеристик — 4; вывод — 2
3	Описать модель BDI и привести пример её применения	корректность описания — 4; пример — 4; ясность объяснения — 2
4	Классифицировать различные типы агентных сред	корректность классификации — 5; аргументация — 3; оформление — 2
5	Описать структуру интеллектуального агента	полнота структуры — 5; корректность терминов — 3; логика — 2
6	Изучить библиотеку моделей NetLogo и описать одну модель	анализ модели — 4; описание поведения агентов — 4; вывод — 2
7	Создать простую модель движения агентов в NetLogo	корректность модели — 5; работа программы — 3; комментарии — 2
8	Изменить параметры модели NetLogo и проанализировать результаты	проведение эксперимента — 4; анализ — 4; вывод — 2
9	Исследовать модель flocking в NetLogo	объяснение правил — 4; анализ поведения — 4; вывод — 2
10	Разработать простую агентную модель распространения информации	корректность модели — 5; реализация — 3; анализ — 2
11	Описать принцип агентного моделирования	корректность определения — 4; объяснение принципов — 4; примеры — 2
12	Исследовать команды NetLogo для управления агентами	корректность использования — 5; описание команд — 3; вывод — 2
13	Создать модель взаимодействия нескольких агентов	корректность реализации — 5; демонстрация взаимодействия — 3; анализ — 2
14	Изучить концепцию LLM-агентов	корректность описания — 4; структура агента — 4; вывод — 2
15	Описать архитектуру LLM-агента	корректность компонентов — 4; полнота — 4; логичность — 2
16	Создать простой Chain в LangChain	корректность реализации — 5; работа программы — 3; комментарии — 2
17	Реализовать агента LangChain с использованием инструмента	корректность реализации — 5; использование инструмента — 3; вывод — 2
18	Реализовать память агента в LangChain	корректность реализации — 5; демонстрация работы — 3; комментарии — 2
19	Исследовать работу prompt-engineering	корректность примеров — 4; анализ — 4; вывод — 2

№	Задание	Критерии оценки (макс. 10 баллов)
20	Создать простой граф выполнения задач в LangGraph	корректность графа — 5; реализация — 3; комментарии — 2
21	Реализовать stateful-агента в LangGraph	корректность реализации — 5; управление состояниями — 3; анализ — 2
22	Изучить взаимодействие нескольких LLM-агентов	корректность модели — 4; анализ взаимодействия — 4; вывод — 2
23	Описать механизм кооперации агентов	корректность определения — 4; примеры — 4; вывод — 2
24	Описать механизм конкуренции агентов	корректность определения — 4; примеры — 4; вывод — 2
25	Исследовать протоколы взаимодействия агентов	корректность описания — 4; примеры — 4; вывод — 2
26	Описать распределение задач в многоагентной системе	корректность объяснения — 4; пример — 4; вывод — 2
27	Разработать схему взаимодействия нескольких агентов	корректность схемы — 4; логика взаимодействия — 4; оформление — 2
28	Изучить архитектуру многоагентных систем	корректность описания — 4; классификация — 4; вывод — 2
29	Реализовать простой пример распределённого взаимодействия агентов	корректность реализации — 5; демонстрация взаимодействия — 3; комментарии — 2
30	Исследовать масштабируемость агентной системы	проведение эксперимента — 4; анализ — 4; вывод — 2
31	Описать параллельное выполнение задач агентами	корректность объяснения — 4; пример — 4; вывод — 2
32	Описать распределённую архитектуру системы	корректность описания — 4; структура — 4; вывод — 2
33	Разработать концепцию многоагентной системы для прикладной задачи	корректность концепции — 4; архитектура — 4; вывод — 2
34	Определить роли агентов в системе	корректность ролей — 4; аргументация — 4; оформление — 2
35	Выполнить декомпозицию сложной системы на подсистемы	корректность декомпозиции — 4; обоснование — 4; оформление — 2
36	Разработать архитектуру агентной системы	корректность архитектуры — 4; логика взаимодействия — 4; оформление — 2
37	Разработать сценарий взаимодействия агентов	корректность сценария — 4; логика — 4; оформление — 2
38	Реализовать взаимодействие агентов через обмен сообщениями	корректность реализации — 5; демонстрация работы — 3; комментарии — 2
39	Проанализировать эффективность агентной системы	анализ — 4; интерпретация результатов — 4; вывод — 2
40	Сравнить централизованные и распределённые системы	корректность сравнения — 4; полнота — 4; вывод — 2
41	Изучить применение многоагентных систем в логистике	анализ примера — 4; описание системы — 4; вывод — 2
42	Изучить применение агентных систем в	анализ — 4; описание модели — 4; вывод

№	Задание	Критерии оценки (макс. 10 баллов)
	экономике	— 2
43	Изучить применение агентных систем в робототехнике	анализ — 4; описание системы — 4; вывод — 2
44	Описать архитектуру программного AI-агента	корректность компонентов — 4; полнота — 4; вывод — 2
45	Разработать прототип AI-агента	корректность реализации — 5; функциональность — 3; комментарии — 2
46	Реализовать взаимодействие агента с внешним инструментом	корректность реализации — 5; демонстрация — 3; комментарии — 2
47	Разработать многоагентную систему из нескольких LLM-агентов	корректность архитектуры — 4; взаимодействие — 4; вывод — 2
48	Провести эксперимент с разработанной агентной системой	проведение эксперимента — 4; анализ — 4; вывод — 2
49	Подготовить отчёт о результатах моделирования	полнота отчёта — 4; анализ результатов — 4; оформление — 2
50	Подготовить презентацию разработанной агентной системы	содержание — 4; демонстрация решения — 4; оформление — 2

Деятельность студентов по формированию и развитию навыков учебной самостоятельной работы

В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления, саморефлексии и становится активным самостоятельным субъектом учебной деятельности.

Выполняя самостоятельную работу под контролем преподавателя студент должен:

- освоить минимум содержания, выносимый на самостоятельную работу студентов и предложенный преподавателем в соответствии с образовательными стандартами высшего образования;

- планировать самостоятельную работу в соответствии с графиком самостоятельной работы, предложенным преподавателем;

- самостоятельную работу студент должен осуществлять в организационных формах, предусмотренных учебным планом и рабочей программой преподавателя;

- выполнять самостоятельную работу и отчитываться по ее результатам в соответствии с графиком представления результатов, видами и сроками отчетности по самостоятельной работе студентов.

студент может:

- сверх предложенного преподавателем (при обосновании и согласовании с ним) и минимума обязательного содержания, определяемого программой по данной дисциплине:

- самостоятельно определять уровень (глубину) проработки содержания материала;

- предлагать темы и вопросы для самостоятельной проработки;

- в рамках общего графика выполнения самостоятельной работы предлагать обоснованный индивидуальный график выполнения и отчетности по результатам самостоятельной работы;

- предлагать свои варианты организационных форм самостоятельной работы;

- использовать для самостоятельной работы методические пособия, учебные пособия, разработки сверх предложенного преподавателем перечня;

– использовать не только контроль, но и самоконтроль результатов самостоятельной работы в соответствии с методами самоконтроля, предложенными преподавателем или выбранными самостоятельно.

Самостоятельная работа студентов должна оказывать важное влияние на формирование личности будущего бакалавра, она планируется студентом самостоятельно. Каждый студент самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием по каждой дисциплине. Он выполняет внеаудиторную работу по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Вопросы для проверки СРС

1. Что понимается под интеллектуальным агентом?
2. Какие основные свойства характеризуют интеллектуального агента?
3. В чем заключается принцип автономности интеллектуальных агентов?
4. Что такое агентная среда и какие её характеристики выделяют?
5. Чем полностью наблюдаемая среда отличается от частично наблюдаемой?
6. Что означает детерминированная и стохастическая среда?
7. В чем различие между статической и динамической средой?
8. Что такое рациональный агент?
9. Что понимается под функцией полезности агента?
10. Как формально описывается поведение интеллектуального агента?
11. Какие основные типы архитектур интеллектуальных агентов существуют?
12. В чем особенности реактивной архитектуры агентов?
13. Чем делиберативные агенты отличаются от реактивных?
14. Что представляет собой гибридная архитектура агента?
15. Что такое модель BDI (Belief–Desire–Intention)?
16. Какие компоненты включает модель BDI?
17. Что такое планирование действий агента?
18. Как агент может использовать модель среды?
19. Что понимается под обучающимся агентом?
20. Какие этапы включает процесс принятия решений агентом?
21. Что такое многоагентная система?
22. Какие преимущества имеют многоагентные системы по сравнению с централизованными системами?
23. Какие типы взаимодействия агентов существуют?
24. Что понимается под кооперацией агентов?
25. Что означает конкуренция между агентами?
26. Что такое координация в многоагентных системах?
27. Что такое протокол взаимодействия агентов?
28. Какие способы обмена сообщениями используются в многоагентных системах?
29. Что такое распределение задач между агентами?
30. Что означает самоорганизация в многоагентных системах?
31. Что понимается под агентным моделированием?
32. В чем преимущества агентного моделирования сложных систем?
33. Какие этапы включает процесс разработки агентной модели?
34. Что представляет собой агентная модель среды?
35. Какие типы агентов используются в среде NetLogo?
36. Что такое turtles в NetLogo?
37. Что такое patches в NetLogo?
38. Что такое links в NetLogo?
39. Как выполняется управление агентами в NetLogo?

40. Какие методы используются для анализа результатов моделирования?
41. Что такое LLM (Large Language Model)?
42. Какую роль играют LLM в разработке интеллектуальных агентов?
43. Что понимается под LLM-агентом?
44. Какие основные компоненты включает архитектура LLM-агента?
45. Что такое prompt и какую роль он играет при работе с LLM?
46. Что понимается под prompt engineering?
47. Что такое цепочка обработки (Chain) в библиотеке LangChain?
48. Какие функции выполняют инструменты (tools) в архитектуре агента?
49. Что такое память агента в системах на основе LLM?
50. Как осуществляется взаимодействие агента с внешними сервисами?
51. Что представляет собой библиотека LangChain?
52. Какие задачи решает библиотека LangGraph?
53. Что такое граф выполнения задач в LangGraph?
54. Какие элементы включает графовая модель агента?
55. Что понимается под stateful-агентом?
56. Что представляет собой платформа Agno?
57. Какие возможности предоставляет Agno для разработки AI-агентов?
58. Что понимается под распределённым программированием?
59. В чем состоит отличие распределённых и параллельных вычислений?
60. Какие основные этапы включает проектирование многоагентной системы?

ОБЩИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение, рекомендованной литературы, активное участие на лабораторных занятиях. Но для успешной учебной деятельности, ее интенсификации, необходимо учитывать следующие субъективные факторы:

1. Знание школьного программного материала, наличие прочной системы знаний, необходимой для усвоения основных вузовских курсов. Это особенно важно для математических дисциплин. Необходимо отличать пробелы в знаниях, затрудняющие усвоение нового материала, от малых способностей. Затратив силы на преодоление этих пробелов, студент обеспечит себе нормальную успеваемость и поверит в свои способности.

2. Наличие умений, навыков умственного труда:

- а) умение конспектировать на лекции и при работе с книгой;
- б) владение логическими операциями: сравнение, анализ, синтез, обобщение, определение понятий, правила систематизации и классификации.

3. Специфика познавательных психических процессов: внимание, память, речь, наблюдательность, интеллект и мышление. Слабое развитие каждого из них становится серьезным препятствием в учебе.

4. Хорошая работоспособность, которая обеспечивается нормальным физическим состоянием. Ведь серьезное учение — это большой многосторонний и разнообразный труд. Результат обучения оценивается не количеством сообщаемой информации, а качеством ее усвоения, умением ее использовать и развитием у себя способности к дальнейшему самостоятельному образованию.

5. Соответствие избранной деятельности, профессии индивидуальным способностям. Необходимо выработать у себя умение саморегулировать свое эмоциональное состояние и устранять обстоятельства, нарушающие деловой настрой, мешающие намеченной работе.

6. Овладение оптимальным стилем работы, обеспечивающим успех в деятельности. Чередование труда и пауз в работе, периоды отдыха, индивидуально обоснованная норма продолжительности сна, предпочтение вечерних или утренних занятий, стрессоустойчивость на экзаменах и особенности подготовки к ним,

7. Уровень требований к себе, определяемый сложившейся самооценкой.

Адекватная оценка знаний, достоинств, недостатков - важная составляющая самоорганизации человека, без нее невозможна успешная работа по управлению своим поведением, деятельностью.

Одна из основных особенностей обучения в высшей школе заключается в том, что постоянный внешний контроль заменяется самоконтролем, активная роль в обучении принадлежит уже не столько преподавателю, сколько студенту.

Зная основные методы научной организации умственного труда, можно при наименьших затратах времени, средств и трудовых усилий достичь наилучших результатов.

Эффективность усвоения поступающей информации зависит от работоспособности человека в тот или иной момент его деятельности.

Работоспособность - способность человека к труду с высокой степенью напряженности в течение определенного времени. Различают внутренние и внешние факторы работоспособности.

К внутренним факторам работоспособности относятся интеллектуальные особенности, воля, состояние здоровья.

К внешним:

- организация рабочего места, режим труда и отдыха;
- уровень организации труда - умение получить справку и пользоваться информацией;
- величина умственной нагрузки.

Выдающийся русский физиолог Н. Е. Введенский выделил следующие условия продуктивности умственной деятельности:

- во всякий труд нужно входить постепенно;
- мерность и ритм работы. Разным людям присущ более или менее разный темп работы;
- привычная последовательность и систематичность деятельности;
- правильное чередование труда и отдыха.

Отдых не предполагает обязательного полного бездействия со стороны человека, он может быть достигнут простой переменой дела. В течение дня работоспособность изменяется. Наиболее плодотворным является *утреннее время* (с 8 до 14 часов), причем максимальная работоспособность приходится на период с 10 до 13 часов, затем *послеобеденное* - (с 16 до 19 часов) и *вечернее* (с 20 до 24 часов). Очень трудный для понимания материал лучше изучать в начале каждого отрезка времени (лучше всего утреннего) после хорошего отдыха. Через 1-1,5 часа нужны перерывы по 10 - 15 мин, через 3 - 4 часа работы отдых должен быть продолжительным - около часа.

Составной частью научной организации умственного труда является овладение техникой умственного труда.

Время, которым располагает студент для выполнения учебного плана, складывается из двух составляющих: одна из них - это аудиторная работа в вузе по расписанию занятий, другая - внеаудиторная самостоятельная работа. Задания и материалы для самостоятельной работы выдаются во время учебных занятий по расписанию, на этих же занятиях преподаватель осуществляет контроль за самостоятельной работой, а также оказывает помощь студентам по правильной организации работы.

Самостоятельные занятия потребуют интенсивного умственного труда, который необходимо не только правильно организовать, но и стимулировать. При этом очень важно

уметь поддерживать устойчивое внимание к изучаемому материалу. Выработка внимания требует значительных волевых усилий. Именно поэтому, если студент замечает, что он часто отвлекается во время самостоятельных занятий, ему надо заставить себя сосредоточиться. Подобную процедуру необходимо проделывать постоянно, так как это является тренировкой внимания. Устойчивое внимание появляется тогда, когда человек относится к делу с интересом.

Следует правильно организовать свои занятия по времени: 50 минут - работа, 5-10 минут - перерыв; после 3 часов работы перерыв - 20-25 минут. Иначе нарастающее утомление повлечет неустойчивость внимания. Очень существенным фактором, влияющим на повышение умственной работоспособности, являются систематические занятия физической культурой. Организация активного отдыха предусматривает чередование умственной и физической деятельности, что полностью восстанавливает работоспособность человека.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература:

Болотский, А. В. Исследование операций и методы оптимизации Электронный ресурс / Болотский А. В., Кочеткова О. А. - Санкт-Петербург : Лань, 2020. - 116 с. - ISBN 978-5-8114-4568-4

Сараев, П. В. Методы машинного обучения Электронный ресурс : Методические указания и задания к лабораторным работам по курсу / П. В. Сараев. - Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2017. - 48 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397

[illegible]

Дополнительная литература:

Методические рекомендации к практическим занятиям по дисциплине "Теория и практика машинного перевода" : Направление подготовки 45.04.02 Лингвистика. Магистерская программа "Перевод и переводоведение". Квалификация выпускника - магистр. Форма обучения - очная. Учебный план 2015 года. Изучается в 3 семестре.

- Ставрополь : СКФУ, 2015. - 9 с.

Неделько, В. М. Основы статистических методов машинного обучения Электронный ресурс : Учебное пособие / В. М. Неделько. - Новосибирск : Новосибирский государственный технический университет, 2010. - 72 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-7782-1385-2

Рапаков, Г. Г. Методы и алгоритмы машинного обучения при принятии управленческих решений в региональной системе медицинской профилактики (опыт Вологодской области) Электронный ресурс / Рапаков Г. Г., Касимов Р. А. : монография. - Вологда : ВоГУ, 2014. - 143 с. - ISBN 978-5-87851-548-1

Учебно-методическая литература:

Сараев, П. В. Методы машинного обучения Электронный ресурс : Методические указания и задания к лабораторным работам по курсу / П. В. Сараев. - Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2017. - 48 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397