11411

МИНИСТЕРСТВО ЦИФРОВЫХ ТЕХНОЛОГИЙ
РЕСПУБЛИКИ УЗБЕКИСТАН ТАШКЕНТСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
ИМЕНИ МУХАММАДА АЛЬ-ХОРАЗМИЙ

Факультет «СФИТ ТУИТ-БГУИР»
Кафедра «Информационно-компьютерные технологии и программирование»

Яхшибоев Р.Э., Довлетова С.Б.

«ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЕ ПРАКТИЧЕСКИХ РАБОТ
для студентов кафедры информационно-компьютерные технологии и программирование

Ташкент – 2023

Авторы: Яхшибоев Р.Э., Довлетова С.Б.

«Проектирование программного обеспечения интеллектуальных систем» / ТУИТ имени Мухаммада аль-Хоразмий, 28 с. Ташкент, 2023 г.

Методические указания представляют собой указания для изучения технологий проектирования, реализации и дальнейшего развития надежных, гибких и эффективных программных систем (организация работ, анализ предметной области, спецификация требований, разработка технического задания, проектирование, моделирование, конструирование, реализация, отладка).

Изучение данной учебной дисциплины способствует созданию условий для формирования интеллектуально развитой личности обучающегося, которой присущи стремление к профессиональному совершенствованию, активному участию в экономической и социально-культурной жизни страны, гражданская ответственность и патриотизм.

Методическое указание предназначено студентам по 60610700 - Искусственный интеллект (1-40 03 01 Искусственный интеллект) направлениям совместного факультета Ташкентский университет информационных технологии имени Мухаммада ал-Хоразмий и Белорусский государственный университет информатики и радиоэлектроники.

Рецензенты :

Исмаилов О.М. - д.т.н., профессор кафедры "Информационно-компьютерные технологий и программирования" ТУИТ имени Мухаммада аль-Хоразмий.

Махсудов В. Г. - PhD кафедры "Биомедицинская инженерия, информатика и биофизика", ТМА

© Ташкентский университет информационных технологий имени Мухаммада ал-Хоразми 2023 год

Практическая работа № 1

Тема: Знакомство с семантической памятью.

Цель работы: ознакомиться с принципами работы с семантической памяти.

Теоретическая сведения

Семантическая память — это один из двух типов явного память (или декларативная память) (наша память фактов или событий, которая явно сохраняется и извлекается). Семантическая память относится к общему мировому знанию, мы накопили за свою жизнь. Эти общие знания (факты, идеи, значения и концепции) (факты, идеи, значения и концепции) зависят от культуры. Семантическая память отличается от эпизодической памяти, которая представляет собой нашу память о переживаниях и конкретных событиях, происходящих в течение жизни, которые мы воссоздаем в любой момент.

Суть семантической памяти состоит в том, что ее содержание не привязано к какому-либо конкретному случаю опыта, как в эпизодической памяти. Вместо этого в семантической памяти хранится «суть» опыта, абстрактная структура, которая применяется к широкому спектру эмпирических объектов и очерчивает категориальные и функциональные отношения между такими объектами.

Семантическая память может быть представлена в виде баз знаний - структурированных совокупностей фактов и правил, описывающих определенную область знаний. Такие базы знаний используются в различных областях, например, в медицине, финансах, лингвистике и т.д.

Основная задача семантической памяти - обеспечить эффективную обработку и хранение информации, что позволяет создавать более интеллектуальные и автоматизированные системы. Она может быть использована для автоматического анализа текстов, распознавания речи, систем управления базами данных и т.д.

Важно отметить, что семантическая память является одной из основных составляющих искусственного интеллекта. Она позволяет

MUHAMMAD AL-XORAZMIY NOMID. TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
AXBOROT-RESURS MARKAZI

создавать более сложные и адаптивные системы, которые способны к самообучению и анализу больших объемов информации.

Семантическая память может быть представлена в виде базы знаний, содержащей информацию об объектах, понятиях, связях между ними и правилах, по которым они функционируют. Например, база знаний об автомобилях может содержать информацию о марках автомобилей, их характеристиках, цвете, максимальной скорости и т.д.

В современной науке существует множество подходов к созданию баз знаний, включая реляционные базы данных, онтологии, графовые базы данных и другие. Однако, независимо от конкретной технологии, основными элементами баз знаний являются факты и правила.

Факты представляют собой утверждения о свойствах и отношениях между объектами. Например, "собака - это животное", "кошка имеет хвост", "человек имеет две руки". Факты могут использоваться для формирования более сложных конструкций, таких как правила.

Правила определяют, какие действия должны выполняться на основе имеющихся фактов. Например, "если животное имеет четыре ноги, то это, скорее всего, собака или кошка". Правила также могут использоваться для моделирования процессов и определения правильного порядка действий.

Семантическая память широко используется в различных областях, таких как искусственный интеллект, компьютерные игры, машинный перевод и другие. Она является важным инструментом для описания мира вокруг нас и помогает нам понимать и взаимодействовать с ним более эффективно.

Задания

1. Изучите определение семантической памяти и сравните ее с эпизодической памятью. Напишите краткое эссе, где вы расскажете, чем эти две формы памяти отличаются друг от друга.
2. Создайте базу знаний на тему "география мира". Ваша база знаний должна включать информацию о странах, городах, реках, горах и других географических объектах.
3. Разработайте набор правил для базы знаний, которая описывает работу организации. Например, "если сотрудник проходит

испытательный срок и не выполняет поставленные задачи, то он может быть уволен".

4. Изучите онтологии и их применение в семантической памяти. Создайте онтологию на тему "библиотека" и определите классы объектов, их свойства и отношения между ними. Например, вы можете определить классы "книга", "автор", "издательство" и связи между ними, такие как "автор может написать несколько книг" и "книга может быть издана несколькими издательствами".

Содержание отчёта

1. Использовать листы формата А4
2. Использовать шрифта Times New Roman, размер кегля 14
3. Интервал 1,5
4. Написать тему, цель и задачи практического работы
5. Выполнить задание
6. Написать выводы и ответить письменно на вопросы.

Контрольные вопросы.

1. Что такое семантическая память?
2. Какие задачи можно решать с помощью семантической памяти?
3. Какие виды семантической памяти существуют?
4. Как происходит представление знаний в семантической памяти?
5. Какие методы используются для извлечения знаний из текстов?
6. Какие языки разметки данных используются для описания свойств и отношений между объектами?
7. Какие существуют системы семантической памяти и как они используются в обработке естественного языка?
8. Какие методы визуализации семантической памяти существуют и как они могут быть использованы для анализа знаний?

Практическая работа № 2

Тема: Настройка инструментальной среды для работы с семантической памятью.

Цель работы: овладеть навыками настройки инструментальной среды для работы с семантической памятью

Теоретическая сведения

Семантическая память — это структурированное представление знаний, которые хранятся в человеческом мозге. Эта память позволяет нам понимать язык, обобщать знания и использовать их для принятия решений. В компьютерных науках концепция семантической памяти используется для обозначения способа хранения и использования знаний компьютерной программой.

Настройка инструментальной среды для работы с семантической памятью начинается с выбора подхода к моделированию знаний. Существует несколько подходов к моделированию знаний, таких как онтологии, графовые базы данных и нейросетевые модели.

Онтологии — это формальные описания понятий и отношений между ними. Они обычно представлены в виде графа, где узлы представляют понятия, а связи между узлами - отношения. Онтологии могут быть использованы для автоматического анализа текстов и решения других задач, связанных со знаниями.

Графовые базы данных — это базы данных, которые хранят данные в виде графов. Они могут быть использованы для хранения семантической памяти и для обработки запросов к ней. Некоторые графовые базы данных, такие как Neo4j, OrientDB и ArangoDB, поддерживают специальные запросы на языке Cypher или AQL для работы с графами.

Нейросетевые модели — это модели машинного обучения, которые позволяют автоматически извлекать знания из больших объемов данных. Нейросети могут использоваться для решения задач, связанных с семантической памятью, таких как распознавание именованных сущностей или классификация текстов.

Для работы с семантической памятью также может быть полезно использовать инструменты для обработки естественного языка

(NLP) и машинного обучения, такие как TensorFlow, Keras, NLTK и Spacy. Эти инструменты позволяют обрабатывать тексты, извлекать из них знания и создавать модели для автоматической обработки данных.

Кроме того, при работе с семантической памятью может быть полезно использовать базы знаний, такие как Wikidata, Freebase или DBpedia.

Эти базы знаний содержат структурированные данные о различных объектах, событиях, местах и других концепциях, которые можно использовать для создания связей между понятиями и для извлечения знаний.

При настройке инструментальной среды для работы с семантической памятью также важно учитывать форматы данных, которые будут использоваться. Существуют различные форматы для представления знаний, такие как RDF, OWL, JSON-LD и другие. Каждый из этих форматов имеет свои преимущества и недостатки, и выбор формата должен быть основан на потребностях конкретного проекта.

Наконец, при настройке инструментальной среды для работы с семантической памятью следует учитывать потребности пользователей и специфику проекта. Например, если проект связан с обработкой медицинских данных, то необходимо использовать специализированные онтологии и базы данных, которые содержат соответствующую информацию.

В целом, настройка инструментальной среды для работы с семантической памятью может быть сложным процессом, но правильный выбор подхода и инструментов может значительно улучшить эффективность и точность работы с знаниями.

Практическая часть

При настройке инструментальной среды для работы с семантической памятью следует руководствоваться следующими шагами:

1. Определение целей проекта и области применения. Это поможет определить необходимые онтологии, базы знаний и инструменты.
2. Выбор подходящего инструментария. Существует много инструментов для работы с семантической памятью, таких как RDF-тройки, SPARQL-запросы, OWL-онтологии и другие.

Выбор инструментов должен основываться на целях проекта и его области применения.

3. Создание онтологии и базы знаний. Онтология представляет собой формализованную модель знаний, а база знаний содержит данные, которые соответствуют этой модели. При создании онтологии следует определить основные понятия и связи между ними, а при создании базы знаний - заполнить ее соответствующими данными.
4. Разработка запросов и алгоритмов для извлечения знаний. Запросы и алгоритмы должны быть разработаны с учетом целей проекта и области применения, а также формата данных, используемого в базе знаний.
5. Тестирование и оптимизация. На этом этапе необходимо протестировать инструментальную среду и выполнить необходимые корректировки для улучшения ее эффективности и точности работы.

Примером инструментов для работы с семантической памятью может служить RDF-тройки и SPARQL-запросы. RDF-тройки представляют собой три элемента: субъект, предикат и объект, которые определяют связь между двумя понятиями. SPARQL-запросы используются для извлечения информации из базы знаний, которая представлена в формате RDF.

Для создания онтологии и базы знаний можно использовать инструменты, такие как Protege и RDFox. Protege — это свободно распространяемый инструмент для создания и редактирования онтологий, а RDFox — это программное обеспечение для хранения и обработки RDF-данных.

Задания

1. Определите цели проекта и бизнес-требования: чтобы разработать инструментальную среду для работы с семантической памятью, необходимо четко определить цели проекта и бизнес-требования.
2. Выберите семантический редактор: для работы с онтологиями и базами знаний необходимо выбрать подходящий семантический редактор.
3. Создайте онтологию: после выбора семантического редактора необходимо создать онтологию, которая будет служить основой для

базы знаний.

4. Настройте базу знаний: после создания онтологии необходимо создать базу знаний, которая будет использоваться для хранения знаний.

5. Заполните базу знаний: после настройки базы знаний и создания онтологии необходимо заполнить базу знаний с помощью информации, полученной от сотрудников и других источников.

6. Настройте систему управления доступом и безопасность: для защиты базы знаний и управления доступом к данным необходимо настроить правила управления доступом и безопасности.

Содержание отчёта

1. Использовать листы формата А4
2. Использовать шрифта Times New Roman, размер кегля 14
3. Интервал 1,5
4. Написать тему, цель и задачи практического работы
5. Выполнить задание
6. Написать выводы и ответить письменно на вопросы.

Контрольные вопросы.

1. Каковы этапы настройки инструментальной среды для работы с семантической памятью?
2. Какие типы данных могут быть включены в базу знаний для управления знаниями в организации?
3. Какие преимущества может дать использование инструментальной среды для работы с семантической памятью в организации?
4. Какие задачи нужно выполнить перед внедрением системы управления знаниями в организации?
5. Как проводится тестирование системы управления знаниями перед ее внедрением?
6. Почему важно проводить мониторинг и анализ результатов после внедрения системы управления знаниями?

Практическая работа № 3

Тема: Изучение основных операция семантической памяти

Цель работы: овладеть практическими навыками работы с семантической памятью

Теоретическая сведения

Семантическая память — это компьютерная система, которая предназначена для хранения, организации и поиска знаний и информации. С помощью семантической памяти можно выполнять множество операций, которые помогают организовать информацию и сделать ее более доступной и понятной. Ниже представлены основные операции, которые можно выполнять с помощью семантической памяти:

1. Поиск информации - с помощью семантической памяти можно осуществлять поиск информации, по ключевым словам, терминам, категориям и другим параметрам. Это позволяет быстро находить необходимую информацию и делать более информированные решения.
2. Сравнение понятий - семантическая память позволяет сравнивать понятия между собой, выявлять сходства и различия, а также идентифицировать зависимости между ними.
3. Классификация и категоризация данных - с помощью семантической памяти можно классифицировать и категоризировать данные по различным критериям, что позволяет организовать информацию более эффективно и сделать ее более понятной для пользователей.
4. Анализ контента - с помощью семантической памяти можно анализировать содержание документов и выделять ключевые термины, концепты и понятия, что помогает улучшить поиск и организацию информации.
5. Создание онтологий - семантическая память может быть использована для создания онтологий, которые описывают формат, структуру и содержание данных. Онтологии могут быть использованы для улучшения поиска и организации информации.
6. Автоматическое аннотирование - с помощью семантической памяти можно автоматически аннотировать документы и

тексты, добавляя к ним метаданные, которые облегчают поиск и организацию информации.

7. Рекомендации - семантическая память может использоваться для создания систем рекомендаций, которые анализируют предпочтения пользователя и на основе этого делают рекомендации по интересующим его темам.

Операциями над семантической памятью можно управлять с помощью различных инструментов. Некоторые из наиболее распространенных операций включают в себя:

1. Добавление новых понятий и связей между ними. Эта операция позволяет расширять семантическую память, добавляя новые понятия и устанавливая связи между ними. Например, можно добавить новое понятие "компьютерная мышь" и установить связь между ним и понятием "компьютер".
2. Изменение существующих связей. Эта операция позволяет изменять связи между существующими понятиями. Например, можно изменить связь между понятием "собака" и понятием "домашнее животное" на связь между понятием "собака" и понятием "животное".
3. Удаление понятий и связей. Эта операция позволяет удалить понятия и связи между ними из семантической памяти. Например, можно удалить понятие "микроволновка" и все связи, связанные с ним.
4. Поиск связей между понятиями. Эта операция позволяет находить связи между двумя понятиями в семантической памяти. Например, можно найти связь между понятием "автомобиль" и понятием "двигатель".
5. Поиск понятий по определенным параметрам. Эта операция позволяет находить понятия, соответствующие определенным параметрам. Например, можно найти все понятия, связанные с темой "музыка".

Изучение основных операций семантической памяти позволяет более эффективно управлять ею и использовать для решения различных задач, связанных с обработкой естественного языка, компьютерным зрением, рекомендательными системами и другими областями, где требуется обработка больших объемов информации.

Задания

1. Создайте список слов, которые имеют общую тему, например, "фрукты" или "животные". Попросите учащихся запомнить список слов, а затем задайте им вопросы, которые требуют использования этих слов в контексте

2. Предложите учащимся составить иерархию слов, которые имеют общую тему. Например, если тема - "автомобили", они могут начать с самых общих понятий

3. Предложите учащимся составить список слов, которые имеют сходные значения, например, "голодный", "проголодаться", "сытый".

4. Предложите учащимся составить и запомнить список абстрактных понятий, таких как "любовь", "свобода", "счастье". Затем попросите их объяснить каждое понятие с помощью конкретных примеров или ситуаций.

Содержание отчёта

1. Использовать листы формата А4
2. Использовать шрифта Times New Roman, размер кегля 14
3. Интервал 1,5
4. Написать тему, цель и задачи практического работы
5. Выполнить задание
6. Написать выводы и ответить письменно на вопросы.

Контрольные вопросы.

1. Какие основные типы связей могут быть в семантической сети?
2. Как можно организовать информацию в семантической памяти для более эффективного извлечения?
3. Какие методы используются для тестирования семантической памяти у людей?
4. Как семантическая память связана с когнитивнымипроцессами, такими как восприятие и мышление?
5. Какие факторы могут влиять на эффективность семантической памяти?
6. Как семантическая память используется в приложениях и технологиях, таких как поисковые системы и обработка естественного языка?

Практическая работа № 4

Тема: Разработка теоретико-графовых программ

Цель работы: изучить различные алгоритмы и методы работы с графами, используя выбранную программную среду или язык программирования.

Теоретическая сведения

Разработка теоретико-графовых программ – это процесс создания программного обеспечения, которое использует графы и теорию графов для решения задач.

Такие программы могут быть полезными в различных областях, таких как наука о данных, социальные сети, биология, информационная безопасность и другие.

Вот несколько шагов, которые можно предпринять для разработки теоретико-графовых программ:

1. Определить цель программы: прежде чем начинать разработку программы, нужно определить, какую задачу она должна решать. Например, программа может использоваться для поиска определенного пути в графе, анализа связей между элементами графа или для создания графических представлений данных.
2. Выбрать язык программирования: при выборе языка программирования нужно учитывать функциональные возможности языка, его производительность и удобство для разработчика. Например, для разработки теоретико-графовых программ можно использовать Python, Java, C++, R или другие языки.
3. Выбрать подход к представлению графа: Граф можно представить разными способами, например, в виде списка смежности, матрицы смежности или объектов. Выбор подхода зависит от целей программы и требований к производительности.
4. Разработать алгоритмы: для решения задач, связанных с графами, нужно разработать алгоритмы. Например, для поиска кратчайшего пути в графе можно использовать алгоритм Дейкстры или алгоритм A*.

5. Написать код: после того как алгоритмы разработаны, можно приступать к написанию кода программы.
6. Протестировать программу: после написания кода нужно протестировать программу на различных тестовых данных. Это поможет обнаружить ошибки и улучшить производительность программы.
7. Оптимизировать программу: при необходимости можно оптимизировать программу для улучшения производительности.
8. Документировать программу: Хорошая документация поможет другим разработчикам легче понимать код и использовать программу.
9. Разместить программу: после завершения разработки программу можно разместить на платформе для распространения программного обеспечения. Это может быть открытый исходный код на GitHub, приложение в App Store или Google Play, или распространение через другие каналы.

Некоторые примеры теоретико-графовых программ:

1. NetworkX – это библиотека Python для работы с графами. Она включает в себя множество алгоритмов и функций для создания, манипулирования и анализа графов.

2. Gephi – это программное обеспечение с открытым исходным кодом для визуализации и анализа графов. Оно позволяет создавать красивые графические представления данных и проводить анализ связей между элементами графа.

3. Graph-tool – это библиотека Python для работы с графами. Она включает в себя множество алгоритмов для анализа графов, в том числе алгоритмы для поиска кратчайших путей, обнаружения сообществ и многое другое.

4. Cytoscape – это программа для визуализации и анализа графов. Она позволяет создавать красивые графические представления данных и проводить анализ связей между элементами графа.

5. Neo4j – это графовая база данных, которая позволяет хранить и манипулировать графами. Она предоставляет мощные возможности для работы с графовыми данными, включая запросы на языке Cypher и API для многих языков программирования.

Задания

1. Разработка программы для создания графов. Напишите программу, которая позволяет пользователю создавать графы с заданными вершинами и ребрами. Реализуйте возможность задавать веса ребер и другие параметры. Программа должна позволять сохранять графы в файл и загружать их из файла.

2. Реализация алгоритмов поиска кратчайших путей. Реализуйте алгоритм Дейкстры для поиска кратчайшего пути в графе. Добавьте возможность визуализации найденного пути. Реализуйте также алгоритм Флойда-Уоршелла для поиска всех кратчайших путей между всеми парами вершин в графе.

3. Анализ свойств графов. Напишите программу, которая анализирует свойства графа, такие как количество вершин и ребер, связность, наличие циклов и т.д. Реализуйте возможность визуализации графа и его свойств.

4. Разработка визуализации графов. Разработайте программу, которая позволяет пользователю визуализировать графы в различных форматах, таких как изображения, графические интерфейсы и т.д. Добавьте возможность масштабирования, перемещения и изменения параметров графа.

5. Реализация алгоритмов обхода графов. Реализуйте алгоритм поиска в глубину и поиск в ширину для обхода графа. Добавьте возможность визуализации обхода графа.

6. Интеграция с базами данных графов. Реализуйте программу, которая будет взаимодействовать с базой данных графов, загружать данные, анализировать их и создавать новые графы. Реализуйте возможность визуализации загруженных графов.

7. Разработка алгоритмов сортировки графов. Реализуйте алгоритм топологической сортировки для сортировки графа. Добавьте возможность визуализации отсортированного графа.

Содержание отчёта

1. Использовать листы формата А4
2. Использовать шрифта Times New Roman, размер кегля 14
3. Интервал 1,5
4. Написать тему, цель и задачи практического работы
5. Выполнить задание
6. Написать выводы и ответить письменно на вопросы.

Контрольные вопросы.

1. Что такое граф и какие основные элементы он содержит?
2. Какие алгоритмы обхода графа существуют и как они работают?
3. Что такое алгоритм Дейкстры и как он используется для поиска кратчайшего пути в графе?
4. Что такое алгоритм Флойда-Уоршелла и как он используется для поиска всех кратчайших путей между всеми парами вершин в графе?
5. Что такое топологическая сортировка и для чего она используется?
6. Какие программные средства можно использовать для разработки теоретико-графовых программ?

Практическая работа № 5

Тема: Реализация на языке C++ конкретного варианта программы

Цель работы: научиться реализовывать программы на языке C++ и применять полученные знания для разработки конкретного приложения

Теоретическая сведения

```
#include <iostream>
int main() {
    int num1, num2;
    std::cout << "Enter two numbers: ";
    std::cin >> num1 >> num2;
    int result = num1 * num2;
    std::cout << "The product of " << num1 << " and " << num2 << " is
" << result << std::endl;
    return 0;
}
```

Эта программа использует библиотеку iostream для работы с вводом и выводом. В функции main() определяются две переменные

num1 и num2 для хранения чисел, введенных пользователем. Затем программа запрашивает у пользователя два числа и сохраняет их в переменных num1 и num2.

Затем программа выполняет умножение num1 и num2 и сохраняет результат в переменную result. Наконец, программа выводит результат умножения на экран.

Обратите внимание на использование оператора <<для вывода текстовых сообщений и значений переменных. Также обратите внимание на использование оператора>> для получения значений переменных из ввода пользователя. В конце функции main() программа возвращает 0, чтобы указать, что она успешно завершилась.

Конкретный вариант программы для реализации на языке C++ может быть различным в зависимости от поставленной задачи. В данной теоретической части приведены основные концепции и инструменты языка C++, которые могут быть использованы при написании любой программы на этом языке.

Язык программирования C++ является мультипарадигменным языком, который поддерживает процедурное, объектно-ориентированное и обобщенное программирование. В языке C++ используется компилируемая модель выполнения программы, что позволяет достичь высокой скорости работы программы. Основными конструкциями языка являются переменные, операторы и функции.

Переменные в C++ представляют собой область памяти, которая может содержать значение определенного типа. В языке C++ определено множество типов данных, таких как целочисленные, вещественные, символьные и т.д. Для объявления переменной используется ключевое слово "тип" и имя переменной. Например, int a; объявляет переменную a типа целое число.

Операторы в C++ используются для выполнения различных операций над переменными и значениями. Операторы могут быть арифметическими, логическими, условными и т.д. Например, оператор "+" используется для выполнения сложения двух чисел.

Функции в C++ позволяют объединить блоки кода, которые могут быть многократно использованы в программе. Функция может принимать аргументы и возвращать значение.

MUHAMMAD AL-XORAZMIY NOMID.
TOSHKENT AXBOROT
TEXNOLOGIYALARI UNIVERSITETI
AXBOROT-RESURS MARKAZI

Для объявления функции используется ключевое слово "тип_возвращаемого_значения" имя_функции (список_аргументов) { тело_функции }.

Например, int sum(int a, int b) { return a + b; } объявляет функцию с именем sum, которая принимает два аргумента типацелое число и возвращает их сумму.

Для работы с вводом и выводом данных в C++ используется стандартная библиотека iostream. Для вывода данных на экран используется объект cout, а для ввода данных - объект cin. Например, cout <<«Hello, world!"; выводит на экран строку "Hello, world!".

Для отладки программы в C++ используются различные методы, такие как вывод на экран значений переменных, использование отладчика и т.д. Кроме того, при написании программы следует придержив аться принципов чистоты кода и хорошей практики программирования, таких как использование понятных имен переменных и функций, избегание дублирования кода, разбиение кода на логические блоки и т.д.

При реализации конкретного варианта программы на языке C++ необходимо четко понимать поставленную задачу и использовать соответствующие конструкции и инструменты языка для ее решения. Например, при написании программы для вычисления среднего значения массива необходимо использовать циклы и операторы математических вычислений, а при написании программы для работы с файлами - функции работы с файлами.

Важным аспектом при реализации программы на языке C++ является проверка корректности ввода и обработка ошибок. Например, при вводе числа пользователем необходимо проверять его корректность и предотвращать возможность ошибок, таких как деление на ноль или выход за пределы массива.

В целом, реализация на языке C++ конкретного варианта программы требует хорошего понимания основных конструкций и инструментов языка, а также умения применять их для решения поставленных задач.

Задания

1. Определение требований к программе: перед началом работы необходимо определить цель программы, ее основные функции и возможности, а также ее архитектуру.

2. Проектирование программы: на основе требований, определенных на первом шаге, следующий шаг - это проектирование программы. Этот этап включает в себя выбор структур данных, алгоритмов и методов программирования, а также создание диаграмм классов и последовательности, которые позволят увидеть взаимодействие между объектами и компонентами программы.
3. Написание кода программы: после проектирования следует написание кода программы, используя синтаксис языка программирования C++. Код должен соответствовать требованиям, определенным на первом этапе.
4. Тестирование программы: после написания кода, необходимо провести тестирование программы, чтобы проверить ее работоспособность и соответствие требованиям.
5. Отладка и оптимизация: если в процессе тестирования выявлены ошибки, необходимо провести отладку кода. Кроме того, можно произвести оптимизацию кода, чтобы улучшить его производительность.
6. Документирование программы: после успешной отладки и тестирования программы необходимо документировать ее исходный код, создать инструкции для пользователя, а также описание ее работы и возможностей.

Содержание отчёта

1. Использовать листы формата А4
2. Использовать шрифта Times New Roman, размер кегля 14
3. Интервал 1,5
4. Написать тему, цель и задачи практического работы
5. Выполнить задание
6. Написать выводы и ответить письменно на вопросы.

Контрольные вопросы.

1. Какие шаги нужно выполнить перед началом написания кода программы на C++?
2. Какие основные этапы проектирования программы вы можете назвать?
3. Какие структуры данных и алгоритмы вы можете использовать для решения задач в C++?

4. Какие принципы ООП используются в языке C++?
5. Как провести тестирование программы на C++?
6. Какие методы отладки программы вы можете использовать?

Практическая работа № 6

Тема: Программы, использующей семантическую память

Цель работы: изучение и практическое применение концепции семантической памяти в разработке программного обеспечения.

Теоретическая сведения

Программы, использующие семантическую память, могут иметь различные цели и функциональность, но их общая особенность состоит в использовании знаний и связей между ними для решения задач.

Некоторые примеры программ, использующих семантическую память:

1. Рекомендательные системы: многие рекомендательные системы, такие как системы рекомендаций товаров или фильмов, используют семантическую память для анализа предпочтений пользователей и выявления связей между товарами или фильмами.
2. Поисковые системы: поисковые системы используют семантическую память для анализа и классификации текстовых данных, чтобы обеспечить более точный поиск и релевантные результаты.
3. Машинный перевод: программы машинного перевода используют семантическую память для анализа и перевода текстов с одного языка на другой, учитывая связи между словами и их значениями.
4. Анализаторы текстов: анализаторы текстов используют семантическую память для анализа и обработки текстов, выявления ключевых слов и фраз, а также для создания связей между различными текстовыми элементами.
5. Системы распознавания речи: системы распознавания речи используют семантическую память для анализа и понимания

речи, выявления ключевых слов и фраз, а также для создания связей между различными звуковыми элементами.

6. Интеллектуальные ассистенты: интеллектуальные ассистенты используют семантическую память для анализа и понимания команд и запросов пользователей, а также для создания связей между различными элементами информации, чтобы предоставлять более точные и полезные ответы и рекомендации.

Это лишь некоторые примеры программ, использующих семантическую память. Семантическая память может использоваться в широком диапазоне приложений, где необходимо понимать и использовать знания и связи между различными элементами информации.

Семантическая память - это вид памяти, который используется в искусственном интеллекте для хранения знаний и понятий, а также для их обработки и использования в различных задачах. В отличие от обычной оперативной памяти, семантическая память позволяет хранить и обрабатывать данные более сложных типов, таких как текстовые строки, образы, звуковые записи и т.д.

Программы, использующие семантическую память, могут использоваться в различных областях, включая машинный перевод, распознавание речи, обработку естественного языка и другие задачи, связанные с обработкой текстов и знаний.

В программировании на языке C++ существует несколько библиотек и фреймворков, которые позволяют работать с семантической памятью. Например, библиотека Semantic Cognition Library (SCL) предоставляет возможности для создания иуправления семантической памятью в C++. Она позволяет описывать знания в виде онтологий и использовать их для решения задач в области искусственного интеллекта.

Также существуют специализированные фреймворки для обработки естественного языка, такие как Natural Language Toolkit (NLTK) и Stanford CoreNLP, которые также используют семантическую память для обработки и анализа текстовых данных.

Задания

1. Напишите программу на языке программирования Python, котора будет использовать семантическую память для ответа

на вопросы пользователя. Программа должна запрашивать у пользователя вопросы на естественном языке и выводить ответы, используя семантическую память.

2. Для реализации семантической памяти можно использовать базу знаний или онтологию, представляющую собой совокупность понятий и их связей друг с другом.
3. Программа должна быть способна обрабатывать вопросы на различные темы, например, науку, искусство, спорт и т.д.

Содержание отчёта

1. Использовать листы формата А4
2. Использовать шрифта Times New Roman, размер кегля 14
3. Интервал 1,5
4. Написать тему, цель и задачи практического работы
5. Выполнить задание
6. Написать выводы и ответить письменно на вопросы.

Контрольные вопросы.

1. Какие примеры программ могут использовать семантическую память для решения задач?
2. Какие преимущества может предоставить использование семантической памяти в различных областях?
3. Какие проблемы могут возникать при использовании семантической памяти, и как их можно решить?
4. Каким образом можно оценить качество работы программы, использующей семантическую память?
5. Какие перспективы есть в развитии семантической памяти и ее применения в будущем?

ЗАКЛЮЧЕНИЕ

В заключение можно сказать, что проектирование программного обеспечения для интеллектуальных систем — это сложный и ответственный процесс, который требует от разработчиков не только высокой технической подготовки, но и глубокого понимания того, как работает конкретная интеллектуальная система и какие задачи она должна решать.

Важно учитывать, что интеллектуальные системы по своей природе являются динамическими и изменчивыми объектами, что усложняет процесс их проектирования и требует гибкого и адаптивного подхода к разработке программного обеспечения.

При проектировании программного обеспечения для интеллектуальных систем необходимо учитывать такие факторы, как сложность системы, требования к производительности, масштабируемость, надежность и безопасность.

Важно также использовать современные методы и инструменты разработки, такие как алгоритмы машинного обучения, нейронные сети, инструменты автоматизации тестирования и т.д.

Интеллектуальные системы являются одной из самых важных областей разработки программного обеспечения в настоящее время, и в перспективе их роль только увеличится. Поэтому разработка программного обеспечения для интеллектуальных систем является перспективной и востребованной областью для профессионалов в области IT.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Чистая Архитектура. Искусство разработки программного обечения. Роберт Мартин -Питер 2018. – **ISBN 978-5-4461-0772**
2. Предметно-ориентированное проектирование. Самое основное. Вон Вернон 2020. – **ISBN 978-5-9908463-8-8**
3. Соммервилл И., Сойер Л. Программная инженерия. Москва: Техносфера, 2016.
4. Паттерсон Д., Хеннесси Дж. Архитектура компьютера и проектирование компьютерных систем. Москва: Вильямс, 2009.
5. Рассел С., Норвиг П. Искусственный интеллект: современный подход. Москва: Вильямс, 2007.
6. Джексон М. Программирование в среде Windows. Москва: Питер, 2007.
7. Гуд С. Теория алгоритмов: полный курс. Москва: Вильямс, 2010.
8. Бурлак С. Логическое программирование на Prolog для начинающих. Москва: БИНОМ. Лаборатория знаний, 2015.
9. Джонсон-Лэйрд П.Н. Как мы мыслим: зрительный ум и умозаключение. Москва: МИФИ, 2006.
10. Рассел С., Норвиг П. Искусственный интеллект: современный подход. Москва: Вильямс, 2007.
11. Самарский А.А. Численные методы: учебник для вузов. Москва: Наука, 2003.
12. Дуглас К., Кинг П. Современная компьютерная графика: Программирование трехмерных приложений. Москва: ДМК Пресс, 2006.

СОДЕРЖАНИЕ

2600

Методические указания по выполнение практических работ попредмету «Проектирование программного обеспечения интеллектуальных систем».

Рассмотрено на заседании кафедры «Информационно-компьютерные технологии и программирование»
от «___»_____2023 г. Протокол №_____

Рассмотрено на заседании факультета «СФИТ ТУИТ-БГУИР» от «____»_____2023 г. Протокол №_____

Рассмотрено и рекомендовано к изданию на заседании научно–методического Совета ТУИТ
от «____»_____2023 г. Протокол №_____

Составители: Р.Э. Яхшибоев
С.Б. Довлетова

Рецензенты: О.М. Исмаилов

В.Г. Махсудов

Ответственный редактор: Р.Э. Яхшибоев

Зав.кафедрой З.Алламуратова

Формат 60x84 1/16. Печ.лист 1,75.
Заказ №44. Тираж 20.
Отпечатано в «Редакционно издательском» отделе при ТУИТ.
Ташкент ул. Амир Темур, 108.