Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 23.09.2026 19:10:44
Уникальный программный ключ:
637517d24e103c3db032acf37e839d98ec1c5bb2f5eb89c29abfcd7f03985447

**Образовательное частное учреждение высшего образования**

**«МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С. ГРИБОЕДОВА»**

(ИМПЭ им. А.С. Грибоедова)

**МЕЖДУНАРОДНЫЙ ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И БИЗНЕС-ИНФОРМАТИКИ**

**УТВЕРЖДАЮ**
Директор Международного института информационных технологий и бизнес-информатики
_____ Т.В. Новикова
«17» декабря 2025г.

# ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

**по учебной дисциплине:**
**ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ**

**Направление подготовки**
**09.03.03 Прикладная информатика**
**(уровень бакалавриат)**

**Направленность (профиль):**
**«Анализ данных»**

**Форма обучения**: **очная, заочная**

**Москва**

Фонд оценочных средств для дисциплины «Интеллектуальные информационные системы». Направление подготовки/специальность 09.03.03 Прикладная информатика, направленность (профиль/специализация): Анализ Данных/ В.Н. Назаров – М.: ИМПЭ им. А.С. Грибоедова.

Фонд оценочных средств является неотъемлемой частью рабочей программы дисциплины.

Разработчик: В.Н. Назаров

Заведующий кафедрой: _____/ Н. Н. Загускин

## 1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Настоящий Фонд оценочных средств (ФОС) по дисциплине «Интеллектуальные информационные системы» является неотъемлемым приложением к рабочей программе дисциплины (РПД) «Интеллектуальные информационные системы». На данный ФОС распространяются все реквизиты утверждения, представленные в РПД по данной дисциплине.

## 2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

Для определения качества освоения обучающимися учебного материала по дисциплине используются следующие оценочные средства:

| № п/п | Оценочное средство | Краткая характеристика оценочного средства | Представление оценочного средства в ФОС |
|---|---|---|---|
| 1 | Тестирование | Вид контроля, позволяющий оценить изученный теоретический материал. | Вопросы для проведения тестирования |
| 2 | Практические задания | Вид контроля, позволяющий оценить умение обучающегося применять осваиваемую компетенцию в практических ситуациях и при решении производственных задач | Задания к практическому занятию |
| 3 | Контрольная работа | Вид контроля, позволяющий определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций | Задания контрольной работы |
| 4 | Самостоятельная работа | Вид контроля, позволяющий оценить проработку теоретического материала, изучение рекомендуемой литературы, выполнение практико-ориентированных заданий (заполнение таблиц, проведение сравнительного анализа, составление схем и др.), решение практических задач, создание презентаций, написание рефератов, подборку нормативного и иного материала и выполнение других заданий | Задания самостоятельной работы |
| 5 | Курсовая работа | Вид контроля, позволяющий выявить степень владения базовыми знаниями, умениями и навыками, необходимыми для обучения, и определить уровень владения новым материалом | Индивидуальные задания (темы) для курсовой работы |
| 6 | Зачет/Зачет с оценкой/Экзамен | Вид контроля, позволяющий выявить степень овладения знаниями, умениями и навыками, необходимыми для дальнейшего освоения образовательной программы подготовки | Вопросы для подготовки к зачету/зачету с оценкой/экзамену |

## 3. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЕ

### 3.1. Сопроводительная информация.

| | |
|---|---|
| Разработчик | В.Н. Назаров |
| Кафедра | Прикладной информатики и информационных технологий |
| Наименование дисциплины | Интеллектуальные информационные системы |
| Факультет / институт | Институт международной экономики, лидерства и менеджмента |
| Направление подготовки / специальность | 09.03.03 Прикладная информатика |
| Количество вопросов в оценочных заданиях (диапазон) | От 12 до 24 |
| Общее время тестирования (мин) | 90 |
| Общее количество вопросов/заданий в ФОС | 24 |
| Размещенность на сайте Университета примерного перечня вопросов, заданий ФОС – для подготовки обучающихся к прохождению оценки (да / нет) | Да |

### 3.2. Модели контролируемых компетенций:

| Код компетенции | Формулировка компетенции | Индикаторы достижения компетенции |
|---|---|---|
| **ПК-7** | Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов<br>**ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов |

## 4. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ТЕКУЩЕГО КОНТРОЛЯ

### 4.1. ТИПОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ.

Тесты содержат набор вопросов, в полном объеме охватывающие изученный теоретический материал по указанной теме. Выполнение тестов позволяет определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций. Индивидуальный тестовый сеанс для каждого обучающегося формируется по специальному алгоритму, обеспечивающему заданную тематическую структуру и пропорциональное наличие вопросов разного типа и сложности.

***При формировании тестов необходимо использовать задания следующих типов***:

Тип задания 1. Задания закрытого типа на установление соответствия.

Тип задания 2. Задания закрытого типа на установление последовательности.

Тип задания 3. Задания комбинированного типа, предполагающие выбор одного правильного ответа из предложенных

Тип задания 4. Задания комбинированного типа, предполагающие выбор нескольких ответов из предложенных

Тип задания 5. Задания открытого типа с развернутым ответом.

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| **Тема 1.1** Искусственный интеллект: предпосылки, развитие, современное состояние разработок | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов | **Задание 1**<br>Вопрос: Какие из перечисленных технологий и фреймворков активно используются в современной разработке систем искусственного интеллекта и поддерживаются или имеют аналоги в российской ИТ-экосистеме?<br>1. PyTorch<br>2. TensorFlow<br>3. Oracle Database<br>4. Фонд «ВЭБ.РФ»<br>Правильные ответы: 1, 2<br>Пояснение:<br>● PyTorch и TensorFlow являются правильными ответами. Это мировые стандарты в области глубокого обучения. Российские компании и исследователи активно их используют, а также развивают собственные решения и надстройки (например, от Sber AI и Yandex), совместимые с этими фреймворками.<br>● Oracle Database является неправильным ответом. Это проприетарная система управления базами данных американской компании, которая не является специализированным инструментом для разработки AI и её использование в новых российских проектах ограничено.<br>● Фонд «ВЭБ.РФ» является неправильным ответом. Это государственный институт развития, а не технология, язык программирования или программная среда.<br><br>**Задание 2**<br>Вопрос: Установите правильную последовательность этапов внедрения отечественного программного обеспечения в организации в соответствии с принципами импортозамещения.<br>1. Анализ и аудит существующей ИТ-инфраструктуры на наличие зарубежного ПО.<br>2. Пилотная эксплуатация выбранного российского ПО в одном из департаментов. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 3. Массовый переход и полное развертывание отечественного решения по всей организации.<br>4. Тестирование и выбор подходящего российского аналога из реестра Минцифры.<br>5. Обучение сотрудников работе с новым программным обеспечением.<br>Правильный порядок: 1, 4, 2, 5, 3<br>Пояснение: Логика начинается с аудита (1), чтобы понять объем предстоящей работы. Далее следует выбор конкретного решения из одобренного перечня (4). Перед полным внедрением обязателен пилотный запуск (2) для выявления скрытых проблем. Параллельно или сразу после пилота проводится обучение пользователей (5), и только затем — полномасштабное развертывание (3).<br><br>**Задание 3**<br>Вопрос: Расположите в хронологическом порядке появление следующих ключевых технологий и сервисов в области искусственного интеллекта от российских компаний.<br>1. Запуск фреймворка SberVision для компьютерного зрения.<br>2. Создание нейросети «Крибрум» для анализа интернет-трафика.<br>3. Появление голосового помощника «Алиса» от Яндекса.<br>4. Разработка языковой модели GigaChat от Сбера.<br>Правильный порядок: 2, 3, 1, 4<br>Пояснение: Нейросеть «Крибрум» одна из первых получила известность в середине 2010-х. Голосовой помощник «Алиса» был представлен Яндексом в 2017 году. Фреймворк SberVision был анонсирован в рамках развития экосистемы SberAI в начале 2020-х. Модель GigaChat, как прямой аналог ChatGPT, появилась одной из последних, в 2023 году.<br><br>**Задание 4**<br>Вопрос: Установите соответствие между российскими компаниями-разработчиками и их ключевыми продуктами или сервисами в сфере искусственного интеллекта и информационных технологий.<br>**Задание 5** |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Установите соответствие между российскими компаниями-разработчиками и их ключевыми продуктами или сервисами в сфере искусственного интеллекта и информационных технологий.<br><br>Список А (Компания):<br>1. Яндекс<br>2. Сбер<br>3. ОС «Альт»<br>4. Kaspersky<br><br>Список Б (Продукт/Сервис):<br>A. GigaChat — многофункциональная нейросетевая модель<br>B. Tarantool — высокопроизводительная СУБД и платформа приложений<br>C. YandexGPT — генеративная языковая модель<br>D. Дистрибутивы Linux для бизнеса и госсектора<br>E. Neural Networks — фреймворк для кибербезопасности<br><br>Ответ: 1C, 2A, 3D, 4E<br><br>Пояснение: Соответствие строится на основе публично известной информации о флагманских продуктах этих компаний. Яндекс и Сбер активно развивают генеративный ИИ (YandexGPT и GigaChat). Компания «Базальт СПО» развивает линейку ОС «Альт». Лаборатория Касперского инвестирует в AI для задач кибербезопасности. Продукт Tarantool (B) не является ключевым для Яндекса в контексте ИИ для широкой аудитории.<br><br>**Задание 6**<br>Вопрос: Установите соответствие между типами программного обеспечения и их популярными российскими аналогами.<br>Вопрос: Установите соответствие между типами программного обеспечения и их популярными российскими аналогами. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Список А (Тип ПО / Мировой аналог):<br>1. Операционная система (Windows)<br>2. Офисный пакет (Microsoft Office)<br>3. Система управления базами данных (Oracle)<br>4. ERP-система (SAP)<br><br>Список Б (Российский аналог):<br>A. 1С:Предприятие<br>B. Р7-Офис<br>C. Astra Linux<br>D. Postgres Pro<br><br>Ответ: 1C, 2B, 3D, 4A<br><br>Пояснение: Astra Linux — ведущая отечественная ОС. Р7-Офис — один из основных российских офисных пакетов. Postgres Pro — высокопроизводительная российская СУБД на базе открытого PostgreSQL. «1С:Предприятие» — флагманская российская платформа для автоматизации управления бизнесом, прямой аналог ERP-систем<br><br>Пояснение: Astra Linux — ведущая отечественная ОС. Р7-Офис — один из основных российских офисных пакетов. Postgres Pro — высокопроизводительная российская СУБД на базе открытого PostgreSQL. «1С:Предприятие» — флагманская российская платформа для автоматизации управления бизнесом, прямой аналог ERP-систем.<br>**Задание 7**<br>Вопрос: Какая российская высокопроизводительная СУБД, основанная на открытом коде PostgreSQL, широко используется в государственном и корпоративном секторе для создания критичных к нагрузкам информационных систем?<br>1. MySQL<br>2. Tarantool<br>3. Postgres Pro<br>4. IBM Db2 |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Правильный ответ: 3<br>Пояснение:<br>● Postgres Pro является правильным ответом, так как это именно российская коммерческая версия PostgreSQL с существенными улучшениями производительности и функциональности.<br>● MySQL является неправильным ответом, так как это СУБД, изначально созданная шведской компанией, а сейчас принадлежащая американской Oracle Corporation.<br>● Tarantool является неправильным ответом. Это не СУБД в классическом понимании, а платформа для in-memory вычислений и база данных в виде приложения.<br>● IBM Db2 является неправильным ответом, так как это проприетарная СУБД американской компании IBM.<br>**Задание 8**<br>Вопрос: Какой фреймворк от российской компании Sber, предоставляющий инструменты для построения полного цикла ML-разработки — от исследования до инференса, является частью экосистемы SberAI?<br>1. MLflow<br>2. Airflow<br>3. ML Space<br>4. H2O.ai<br>Правильный ответ: 3<br>Пояснение:<br>● ML Space является правильным ответом. Это облачная платформа от SberBank для полного цикла работ по машинному обучению и анализу данных, ключевой продукт в экосистеме SberAI.<br>MLflow, Airflow и H2O.ai являются неправильными ответами. Это популярные международные open-source фреймворки и платформы (изначально американские), а не продукты, разработанные российской компанией.<br>**Задание 9**<br>Вопрос: Как называется основная государственная информационная |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | система, являющаяся реестром российских программ для электронных вычислительных машин и баз данных, созданная для поддержки импортозамещения? Ответ должен быть представлен в виде общепринятой аббревиатуры.<br>Правильный ответ: ЕРП (Единый реестр российских программ)<br>Пояснение: ЕРП (Единый реестр российских программ) — это система, которую ведет Минцифры России. Наличие программы в этом реестре является обязательным условием для ее допуска к закупкам государственными и муниципальными учреждениями в соответствии с постановлением Правительства РФ № 1236.<br><br>**Задание 10**<br>Вопрос: Какая российская компания, являющаяся лидером на рынке поисковых систем и интернет-сервисов, разработала и активно развивает генеративную языковую модель YandexGPT, а также фреймворк для машинного обучения CatBoost? Ответ состоит из одного слова.<br>Правильный ответ: Яндекс<br>Пояснение: Яндекс — это одна из крупнейших российских ИТ-компаний, которая с самого начала своей деятельности была сфокусирована на разработке алгоритмов искусственного интеллекта для поиска, а теперь создает и флагманские генеративные модели (GPT) и всемирно известные open-source библиотеки для ML (CatBoost, MatrixNet). |
| **Тема 1.2** Данные и знания | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов | **Задание 1**<br>Какие из перечисленных технологий и языков используются для работы со структурированными данными и знаниями в системах искусственного интеллекта?<br>1. SQL для выполнения запросов к реляционным базам данных, хранящим факты и онтологии.<br>2. Язык OWL (Web Ontology Language) для формального описания знаний и онтологий. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 3. Язык разметки HTML для создания веб-интерфейсов.<br>4. Фреймворк TensorFlow для работы с векторными представлениями слов (Word2Vec, GloVe).<br><br>Ответ: 1, 2, 4<br><br>Пояснение: SQL является правильным ответом, так это основной язык для взаимодействия с реляционными БД, которые часто используются для хранения структурированных фактов в экспертных системах и других ИИ-приложениях. OWL является правильным ответом, так как это стандартный язык семантической паутины для создания сложных онтологий, позволяющих машинам "понимать" смысл данных. TensorFlow является правильным ответом, так как он используется для создания моделей, которые преобразуют текстовые данные (неструктурированные знания) в числовые векторные представления, с которыми может работать ИИ.<br><br>**Задание 2**<br>Какие из перечисленных компонентов и технологий являются ключевыми для построения систем, основанных на знаниях (Knowledge-Based Systems)?<br>1. База данных PostgreSQL для хранения сырых транзакционных данных.<br>2. База знаний, содержащая правила и факты о предметной области.<br>3. Механизм логического вывода (Inference Engine).<br>4. Графический процессор (GPU) для ускорения матричных вычислений.<br><br>Ответ: 2, 3<br><br>Пояснение: База знаний является правильным ответом, так как это ядро любой системы, основанной на знаниях; она содержит символическое представление экспертных знаний. Механизм логического вывода |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | является правильным ответом, так как это "мозг" системы, который применяет логические правила к базе знаний для получения новых фактов и ответов.<br>**Задание 3**<br>Установите последовательность этапов преобразования данных в знания в конвейере обработки информации интеллектуальной системой.<br>1. Применение знаний: Использование моделей и правил для принятия решений, прогнозирования или генерации insights.<br>2. Интерпретация: Преобразование очищенных данных в структурированный формат, пригодный для анализа (например, таблицы, графы).<br>3. Интеграция знаний: Формализация и сохранение извлеченных закономерностей в базе знаний или в виде моделей машинного обучения.<br>4. Сбор данных: Получение сырых данных из различных источников (сенсоры, базы данных, документы).<br>5. Извлечение знаний: Обнаружение закономерностей, паттернов и взаимосвязей в данных с помощью алгоритмов Data Mining или ML.<br><br>Ответ: 4, 2, 5, 3, 1<br><br>Пояснение: Процесс начинается со сбора сырых данных (4). Затем данные приводятся к структурированному виду (2). На следующем этапе из них извлекаются скрытые закономерности (5). Эти закономерности формализуются и сохраняются как знания (3). Наконец, накопленные знания используются для решения практических задач (1).<br><br>**Задание 4**<br>Расположите в логическом порядке этапы разработки и внедрения отечественной интеллектуальной системы, работающей с данными и знаниями.<br>1. Формализация знаний экспертов и проектирование онтологии предметной области. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 2. Установка и настройка российской операционной системы (Astra Linux) и СУБД (Postgres Pro).<br>3. Интеграция системы с производственными процессами, обучение пользователей.<br>4. Разработка программных модулей (на Python/Java): ETL-процессы, механизм вывода, пользовательский интерфейс.<br>5. Анализ бизнес-требований и выбор архитектурного решения.<br>Правильный порядок: 5, 2, 1, 4, 3<br>Пояснение: Начинается все с анализа требований и проектирования архитектуры (5). Далее подготавливается инфраструктура: ОС и СУБД (2). Затем формализуется самая ценная часть — знания о предметной области (1). После этого создается программное обеспечение, реализующее логику (4). Финальный этап — внедрение и интеграция (3).<br><br>**Задание 5**<br>Вопрос: Установите соответствие между концепцией представления знаний и ее описанием.<br><br>Список А (Концепция):<br>1. Онтология<br>2. Семантическая сеть<br>3. Фрейм<br>4. Логика предикатов первого порядка<br><br>Список Б (Описание):<br>A. Продукционное правило вида "ЕСЛИ (условие) ТО (действие)".<br>B. Формальное, явное описание понятий (классов) и отношений между ними в предметной области.<br>C. Структура данных для представления стереотипной ситуации, состоящая из слотов (атрибутов).<br>D. Граф, узлы которого представляют понятия или объекты, а дуги — отношения между ними.<br>E. Формальная система, использующая кванторы и предикаты для |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | выражения сложных утверждений об объектах.<br><br>Ответ: 1B, 2D, 3C (Фрейм — структура со слотами)<br>● 4 - E (Логика предикатов — формальная система с кванторами)<br>Пояснение: Онтология — это строгое формальное описание (1-B). Семантическая сеть — это графовая модель знаний (2-D). Фрейм — это модель объекта через его атрибуты (3-C). Логика предикатов — это математический фундамент для многих систем представления знаний (4-E). Вариант A описывает форму правила, но не является исчерпывающим описанием.<br><br>**Задание 7**<br>Установите соответствие между типом данных/знаний и технологией или языком для работы с ним в контексте ИИ.<br><br>Список А (Тип данных/знаний):<br>1. Структурированные факты в таблицах<br>2. Онтологии и семантические графы (RDF)<br>3. Логические правила и символьные знания<br>4. Неструктурированные текстовые данные<br><br>Список Б (Технология/Язык):<br>A. Язык запросов SPARQL<br>B. Язык SQL<br>C. Язык Prolog<br>D. Библиотеки Python (NLTK, spaCy) для обработки естественного языка<br><br>Ответ: 1B, 2A, 3C, 4D<br><br>Пояснение: SQL — инструмент для работы с реляционными данными (1-B). SPARQL — язык запросов к семантическим данным (2-A). Prolog — язык логического программирования (3-C). Библиотеки NLP преобразуют текст в структурированную информацию (4-D). |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | **Задание 8**<br>Вопрос: Какой язык программирования, основанный на логике и исчислении предикатов, является классическим выбором для создания систем, основанных на знаниях, благодаря встроенному механизму поиска с возвратом (backtracking) и унификации?<br>1. Python<br>2. Java<br>3. Prolog<br>4. C++<br><br>Правильный ответ: 3<br><br>Пояснение:<br>● Prolog является правильным ответом, так как его вычислительная модель напрямую соответствует задачам логического вывода. Механизмы унификации и автоматического перебора с возвратами позволяют ему "доказывать" теоремы на основе базы фактов и правил, что является сутью работы многих систем, основанных на знаниях.<br>● Python, Java, C++ являются неправильными ответами. Это языки общего назначения (императивные или ООП). Хотя на них можно реализовать механизм логического вывода, для этого потребуется писать его "вручную", в то время как в Prolog он является частью самой языка.<br><br>**Задание 9**<br><br>Вопрос: Какая из перечисленных российских систем управления базами данных, являющаяся развитием PostgreSQL и входящая в реестр Минцифры, может эффективно использоваться не только для хранения структурированных данных, но и для выполнения сложных аналитических запросов, необходимых для подготовки данных для моделей ИИ? |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 1. SQLite<br>2. MySQL<br>3. Postgres Pro<br>4. Microsoft SQL Server<br><br>Правильный ответ: 3<br><br>Пояснение:<br>● Postgres Pro является правильным ответом, так как это российский продукт, унаследовавший все мощные возможности PostgreSQL для аналитики (оконные функции, индексы для полнотекстового поиска, поддержка JSON), что делает его отличным выбором для этапа подготовки данных в ИИ-проектах.<br>● SQLite является неправильным ответом, так как это встраиваемая БД, не предназначенная для сложной аналитической обработки больших объемов данных.<br>● MySQL и Microsoft SQL Server являются неправильными ответами, так как это продукты американских компаний (Oracle Corporation и Microsoft соответственно).<br><br>**Задание 10**<br>Вопрос: Как называется основной открытый язык для создания веб-онтологий, позволяющий определять сложные иерархии классов, свойств и ограничений, что делает знания машиночитаемыми и пригодными для логического вывода? Ответ состоит из аббревиатуры из трех букв.<br><br>Правильный ответ: OWL<br><br>Пояснение: OWL (Web Ontology Language) — это стандарт консорциума W3C, разработанный специально для представления богатых и сложных знаний о предметных областях. Он основан на |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | логике и позволяет делать выводы, которые не были явно заявлены в данных.<br><br>**Задание 11**<br>Вопрос: В контексте машинного обучения, как называется процесс преобразования неструктурированных или категориальных данных в числовой формат, пригодный для обучения алгоритмов? Примером является one-hot encoding или создание векторных представлений слов.<br>Правильный ответ: Векторизация (или Кодирование)<br>Пояснение: Векторизация (или числовое кодирование) — это фундаментальный этап подготовки данных для большинства моделей машинного обучения. Поскольку алгоритмы ML работают с числами, любые тексты, категории или изображения должны быть сначала преобразованы в числовые векторы или матрицы. Без этого шага обучение моделей невозможно. |
| **Тема 1.3** Моделирование знаний о предметных областях как основа интеллектуальных систем | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 1**<br>Вопрос: Какие из перечисленных технологий и языков программирования являются наиболее релевантными для разработки компонентов интеллектуальных систем, связанных с хранением, извлечением и логическим выводом на основе знаний?<br>**1.** Prolog<br>**2.** HTML<br>**3.** SQL<br>**4.** SPARQL<br>Правильные ответы: 1, 3, 4<br>Пояснение:<br>● Prolog является правильным ответом, так как это язык логического программирования, созданный specifically для работы с символическими знаниями и правилами логического вывода, что является основой многих экспертных систем.<br>● SQL является правильным ответом, так как это основной язык для работы с реляционными базами данных, которые часто |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | используются для хранения структурированных фактов и онтологий в информационных системах.<br>● SPARQL является правильным ответом, так как это язык запросов к семантическим базам данных (RDF), который является ключевым технологическим стеком для Веба Данных и сложных онтологий.<br>● HTML является неправильным ответом, так как это язык разметки для создания веб-страниц, а не для программирования логики или работы со знаниями.<br><br>**Задание 2**<br>Вопрос: Для разработки какой компонентной архитектуры современной информационной системы можно эффективно использовать следующие отечественные программные продукты?<br>**1.** PostgreSQL<br>**2.** 1С:Предприятие<br>**3.** Adobe Photoshop<br>**4.** Astra Linux<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● PostgreSQL является правильным ответом, так как это мощная СУБД (российская версия — Postgres Pro), которая может служить backend'ом для хранения данных и моделей знаний интеллектуальной системы.<br>● 1С:Предприятие является правильным ответом, так как эта платформа позволяет моделировать сложные бизнес-процессы и предметные области, что является формой представления знаний, и может интегрироваться с другими компонентами ИС.<br>● Astra Linux является правильным ответом, так как это сертифицированная отечественная ОС, на которой может быть развернута вся инфраструктура информационной системы, включая интеллектуальные модули.<br>● Adobe Photoshop является неправильным ответом, так как это графический редактор, не имеющий отношения к разработке |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | архитектуры информационных систем.<br><br>**Задание 3**<br>Вопрос: Установите логическую последовательность этапов создания базы знаний для интеллектуальной системы.<br>1. Формализация знаний (создание онтологии, продукционных правил).<br>2. Извлечение знаний у экспертов предметной области.<br>3. Верификация и тестирование базы знаний на реальных задачах.<br>4. Выбор способа представления знаний (фреймы, онтологии, правила).<br>5. Реализация базы знаний в выбранной программной среде.<br>Правильный порядок: 2, 4, 1, 5, 3<br>Пояснение: Процесс начинается со сбора информации у экспертов (2). На основе этой информации выбирается подходящий метод ее структурирования (4). После выбора метода знания формализуются согласно этому методу (1). Затем формализованная модель реализуется в виде программного кода или конфигурации (5). Финальный этап — проверка работоспособности созданной системы (3).<br><br>**Задание 4**<br>Вопрос: Установите последовательность уровней абстракции при моделировании предметной области в архитектуре информационной системы, от физического хранения до пользовательского представления.<br>1. Уровень логической схемы (таблицы, сущности, атрибуты в СУБД).<br>2. Уровень физической схемы (файлы данных, индексы на диске).<br>3. Уровень представления (пользовательский интерфейс, отчеты, API).<br>4. Уровень концептуальной схемы (ER-диаграмма, онтология).<br>Правильный порядок: 2, 1, 4, 3<br>Пояснение: Классическая трехуровневая архитектура СУБД (ANSI/SPARC) и систем моделирования знаний начинается с физического хранения (2). Над ним надстраивается уровень логического |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | представления данных для системы (1). Концептуальная схема описывает предметную область в целом, абстрагируясь от деталей реализации (4). Уровень представления адаптирует данные и логику для конечного пользователя или внешней системы (3)<br><br>**Задание 5**<br>Установите соответствие между концепцией моделирования знаний и её описанием или примером технологии.<br><br>Список А (Концепция):<br>1. Онтология<br>2. Фрейм<br>3. Семантическая сеть<br>4. Логический вывод<br><br>Список Б (Описание / Технология):<br>A. Продукционное правило "ЕСЛИ... ТО..."<br>B. Язык Web Ontology Language (OWL)<br>C. Структура данных со слотами для описания стереотипного объекта<br>D. Граф, узлы которого — понятия, а дуги — отношения между ними<br>E. Процесс получения новых фактов на основе существующих с помощью механизма, такого как RacerPro<br><br>Ответ: 1B, 2C, 3D, 4E<br><br>Пояснение: Онтология формально описывает предметную область, и OWL является стандартным языком для этого. Фрейм — это концепция из ИИ для представления объекта через его атрибуты (слоты). Семантическая сеть — это модель знаний в виде графа. Логический вывод — это процесс применения правил к существующим знаниям для генерации новых. Вариант А описывает форму правила, но не является исчерпывающим описанием вывода.<br><br>**Задание 6** |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Вопрос: Установите соответствие между типами программного обеспечения и их ролью в контексте разработки информационных систем, основанных на знаниях.<br>Список А (Тип ПО / Язык):<br>1. SQL<br>2. Prolog<br>3. Git<br>4.Docker<br><br>Список Б (Роль в разработке ИС):<br>A. Язык логического программирования для создания ядра экспертной системы<br>B. Инструмент для управления версиями кода и совместной разработки<br>C. Язык запросов для извлечения структурированных данных из реляционной БД<br>D. Среда выполнения для изоляции и развертывания компонентов системы<br><br>Ответ: 1C, 2A, 3B, 4D<br><br>Пояснение: Соответствие строится на основной функции каждого инструмента. SQL работает с данными в БД. Prolog предназначен для символических вычислений и логики. Git необходим для командной работы над любым ПО. Docker стандартизирует процесс развертывания.<br><br>**Задание 7**<br>Вопрос: Какой язык программирования, основанный на формальной логике и исчислении предикатов, традиционно считается одним из основных для создания "символического" ядра интеллектуальных систем и прототипов экспертных систем?<br>1. Java<br>2. Prolog<br>3. C++<br>4. Python |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Правильный ответ: 2<br>Пояснение:<br>● Prolog является правильным ответом, так как его парадигма — логическое программирование — напрямую соответствует задачам представления знаний (в виде фактов и правил) и автоматического логического вывода на их основе.<br>● Java, C++, Python являются неправильными ответами. Это языки общего назначения (императивные или объектно-ориентированные). Хотя на них *можно* реализовать интеллектуальную систему (и на Python это часто делается для ML), они не являются специализированными языками для символического ИИ и логического вывода, в отличие от Prolog.<br><br>**Задание 8**<br>Вопрос: Какая из перечисленных открытых объектно-реляционных СУБД, имеющая российскую версию (Postgres Pro), является мощным инструментом для хранения сложных структурированных данных и часто используется как компонент backend'а для информационных систем, в том числе с элементами ИИ?<br>1. SQLite<br>2. MySQL<br>3. PostgreSQL<br>4. MongoDB<br>Правильный ответ: 3<br>Пояснение:<br>● PostgreSQL является правильным ответом, так как это именно та СУБД, которая известна своей надежностью, поддержкой сложных типов данных и продвинутых функций SQL, и на основе которой создана российская Postgres Pro.<br>● SQLite является неправильным ответом, так как это встраиваемая легковесная БД, не предназначенная для высоконагруженных информационных систем.<br>● MySQL является неправильным ответом. Хотя это популярная СУБД, исторически она была менее функциональной в части |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | сложных типов данных и стандартов SQL по сравнению с PostgreSQL.<br>● MongoDB является неправильным ответом, так как это документоориентированная NoSQL база данных, которая использует принципиально другую, нереляционную, модель данных.<br>**Задание 9**<br>Вопрос: Как называется стандартный язык World Wide Web Consortium (W3C), предназначенный для декларативного описания онтологий в рамках семантической паутины (Semantic Web) и моделирования знаний о предметной области? Ответ состоит из аббревиатуры из трех букв.<br>Правильный ответ: OWL<br>Пояснение: OWL (Web Ontology Language) — это именно тот стандарт W3C, который позволяет формально описывать онтологии, включая классы, свойства, индивидов и отношения между ними, делая знания машиночитаемыми и пригодными для логического вывода.<br><br>**Задание 10**<br>Вопрос: Какая популярная среда разработки от компании JetBrains, имеющая российские корни, предоставляет полнофункциональную поддержку таких языков, как Python, Java и SQL, и широко используется профессионалами для создания информационных систем? Ответ состоит из одного слова.<br>Правильный ответ: PyCharm (или IntelliJ IDEA)<br>Пояснение: Оба продукта являются корректными ответами. PyCharm — специализированная IDE для Python, который часто используется в ML и Data Science. IntelliJ IDEA — IDE для JVM-языков (Java, Kotlin), широко используемая в enterprise-разработке. JetBrains — компания с российскими основателями, и ее продукты являются отраслевым стандартом. |
| **Тема 1.4** Основные компоненты | **ПК-7** Способен осуществлять проектирование | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 1**<br>Вопрос: Какие из перечисленных компонентов являются обязательными или ключевыми для архитектуры классической интеллектуальной |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| интеллектуальной информационной системы | программных интерфейсов | | информационной системы (ИИС)?<br>**1.** База знаний<br>**2.** Решатель (механизм логического вывода)<br>**3.** Графический редактор<br>**4.** Подсистема объяснения решений<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● База знаний является правильным ответом, так как это ядро ИИС, содержащее формализованные знания о предметной области (факты, правила, онтологии).<br>● Решатель (механизм логического вывода) является правильным ответом, так как это "мозг" системы, который обрабатывает знания из базы и входные данные для формирования решений.<br>● Подсистема объяснения решений является правильным ответом, так как она повышает доверие к системе, показывая пользователю цепочку рассуждений, которая привела к результату.<br>● Графический редактор является неправильным ответом, так как это вспомогательный инструмент, не входящий в обязательную архитектуру ИИС.<br><br>**Задание 2**<br>Вопрос: Какие из перечисленных технологий и языков программирования могут быть использованы для реализации компонентов интеллектуальной системы, работающих с базами данных и логическим выводом?<br>**1.** Prolog<br>**2.** SQL<br>**3.** Microsoft Paint<br>**4.** Python с библиотеками (например, SQLAlchemy, PyKE)<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● Prolog является правильным ответом, так как это язык логического программирования, идеально подходящий для |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | создания механизма вывода и работы с символическими знаниями.<br>● SQL является правильным ответом, так как это стандартный язык для взаимодействия с реляционными базами данных, которые часто используются для хранения фактов и онтологий.<br>● Python с библиотеками является правильным ответом, так как Python является универсальным языком с богатой экосистемой библиотек для работы с данными (SQLAlchemy) и даже для создания экспертных систем (PyKE).<br>● Microsoft Paint является неправильным ответом, так как это простой графический редактор, не имеющий отношения к программированию логики ИИС.<br>**Задание 3**<br>Вопрос: Установите правильную последовательность этапов работы интеллектуальной информационной системы при обработке запроса пользователя.<br>1. Получение нового знания и его добавление в базу знаний (опционально).<br>2. Обращение к базе знаний и исходным данным.<br>3. Логический вывод на основе имеющихся знаний и правил.<br>4. Формирование ответа пользователю, при необходимости — с объяснением.<br>5. Получение и анализ входного запроса от пользователя.<br>Правильный порядок: 5, 2, 3, 4, 1<br>Пояснение: Процесс начинается с получения запроса (5). Система анализирует его и обращается к хранилищу знаний и данных (2). Затем активируется механизм вывода, который применяет правила к известным фактам (3). После нахождения решения система формирует и выдает ответ (4). На последнем этапе система может "обучиться", сохранив результат обработки для будущего использования (1).<br><br>**Задание 4**<br>Вопрос: Расположите в правильном порядке уровни стека технологий для развертывания современной интеллектуальной системы, начиная с |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | базового.<br>1. Программные среды разработки (IDE, например, PyCharm)<br>2. Операционная система (например, Astra Linux)<br>3. Прикладные библиотеки ИИ (например, TensorFlow, OWL API)<br>4. Аппаратное обеспечение (серверы, GPU)<br>5. Интеллектуальное приложение<br>Правильный порядок: 4, 2, 1, 3, 5<br>Пояснение: Фундаментом является аппаратное обеспечение (4). На нем работает операционная система, управляющая ресурсами (2). Среда разработки (1) используется для создания кода, который использует специализированные библиотеки ИИ (3). Венчает стек готовое интеллектуальное приложение (5).<br>**Задание 5**<br>Установите соответствие между технологией/языком и компонентом интеллектуальной системы, для реализации которого он преимущественно используется.<br><br>Список А (Технология/Язык):<br>1. Prolog<br>2. SQL<br>3. OWL (Web Ontology Language)<br>4. Python (с библиотеками Flask/Django)<br><br>Список Б (Компонент ИИС):<br>A. Хранение структурированных фактов и онтологий<br>B. Реализация механизма логического вывода<br>C. Формальное описание онтологий предметной области<br>D. Разработка пользовательского интерфейса и веб-интерфейса для системы<br><br>Ответ: 1B, 2A, 3C, 4D<br><br>Пояснение: Prolog по своей парадигме идеален для вывода. SQL — для работы с реляционными БД. OWL — стандарт для описания онтологий. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Python с веб-фреймворками — популярный инструмент для создания фронтенда и бэкенда приложений.<br><br>**Задание 6**<br>Вопрос: Какой компонент интеллектуальной системы отвечает за применение продукционных правил (типа "ЕСЛИ-ТО") к фактам из базы знаний для формирования новых фактов и принятия решений?<br>1. База знаний<br>2. Решатель (механизм вывода)<br>3. Пользовательский интерфейс<br>4. База данных<br>Правильный ответ: 2<br>Пояснение:<br>● Решатель (механизм вывода) является правильным ответом, так как это именно тот компонент, который содержит алгоритмы (прямая или обратная цепочка рассуждений) для манипуляции знаниями и получения новых результатов.<br>● База знаний является неправильным ответом, так как это пассивное хранилище информации, а не активный обработчик.<br>● Пользовательский интерфейс является неправильным ответом, так как он только обеспечивает взаимодействие с пользователем.<br>● База данных является неправильным ответом, так как это система для хранения структурированных данных, но не для выполнения логического вывода.<br><br>**Задание 7**<br>Вопрос: Какая из перечисленных СУБД, имеющая российскую версию Postgres Pro, является надежным решением для хранения структурированных данных, которые могут использоваться в качестве основы или компонента базы знаний интеллектуальной системы?<br>1. SQLite<br>2. PostgreSQL<br>3. Microsoft Access |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 4. MongoDB<br>Правильный ответ: 2<br>Пояснение:<br>● PostgreSQL является правильным ответом, так как это мощная, надежная и полнофункциональная объектно-реляционная СУБД, на основе которой создан отечественный продукт Postgres Pro. Она подходит для хранения сложных структур данных.<br>● SQLite является неправильным ответом, так как это встраиваемая БД для легковесных приложений, не предназначенная для высоконагруженных систем.<br>● Microsoft Access является неправильным ответом, так как это настольная СУБД, не рассчитанная на масштабирование и использование в качестве бэкенда для серьезных ИИС.<br>● MongoDB является неправильным ответом, так как это документоориентированная NoSQL база, которая использует другую модель данных и не является реляционной, что может быть критично для ряда задач логического вывода.<br>**Задание 8**<br>Вопрос: Как называется центральный компонент интеллектуальной системы, который содержит формализованные знания о предметной области в виде фактов, правил и онтологий? Ответ состоит из двух слов.<br>Правильный ответ: База знаний<br>Пояснение: "База знаний" — это общепринятый термин в искусственном интеллекте, который отличает такое хранилище от обычной "базы данных" за счет наличия семантики и структуры, пригодной для логического вывода.<br><br>**Задание 9**<br>Вопрос: Какой язык программирования, основанный на логике и исчислении предикатов, традиционно используется для создания ядра экспертных систем и механизмов логического вывода? Ответ состоит из одного слова.<br>Правильный ответ: Prolog<br>Пояснение: Prolog (сокращение от "Programming in Logic") — это самый |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | известный язык логического программирования, чья парадигма напрямую соответствует задачам представления знаний и автоматического вывода в интеллектуальных системах. |
| **Тема 2.1** Структуры и стратегии поиска в пространстве состояний | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 1**<br>Вопрос: Какие из перечисленных структур данных являются наиболее эффективными и часто используемыми для реализации алгоритмов поиска в пространстве состояний (таких как поиск в глубину, ширину или A*)?<br>1. Очередь (Queue)<br>2. Массив (Array)<br>3. Стек (Stack)<br>4. Приоритетная очередь (Priority Queue)<br>Правильные ответы: 1, 3, 4<br>Пояснение:<br>● Очередь (Queue) является правильным ответом, так как она используется для реализации поиска в ширину (BFS), где состояния обрабатываются в порядке их поступления (FIFO).<br>● Стек (Stack) является правильным ответом, так как он используется для реализации поиска в глубину (DFS), где обрабатывается последнее добавленное состояние (LIFO).<br>● Приоритетная очередь (Priority Queue) является правильным ответом, так как она является основой для алгоритма A*, где состояния извлекаются в зависимости от приоритета, вычисляемого как f(n) = g(n) + h(n).<br>● Массив (Array) является неправильным ответом. Хотя массив можно использовать для хранения состояний, он не эффективен для управления порядком их обхода в контексте поиска. Алгоритмы поиска требуют структур данных, обеспечивающих определенный порядок извлечения элементов (FIFO, LIFO, по приоритету), что не является сильной стороной простого массива. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | **Задание 2**<br>Вопрос: Для реализации и отладки алгоритмов поиска в пространстве состояний, а также для работы с базами данных, хранящими графы состояний, можно эффективно использовать следующие языки программирования и технологии:<br>**1.** Python с библиотеками (например, networkx для работы с графами)<br>2. Язык логического программирования Prolog<br>3. HTML и CSS<br>4. SQL для выполнения запросов к базе данных, хранящей состояния и переходы<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● Python с библиотеками является правильным ответом, так как Python идеально подходит для быстрого прототипирования алгоритмов, а библиотека networkx предоставляет готовые реализации многих алгоритмов на графах.<br>● Язык логического программирования Prolog является правильным ответом, так как его модель вычислений с backtracking (перебор с возвратами) по сути является встроенным механизмом поиска в пространстве состояний, определяемом логическими правилами.<br>● SQL является правильным ответом, так как граф состояний можно представить в реляционной базе данных (таблицы "Состояния" и "Переходы"), а для поиска путей использовать рекурсивные SQL-запросы.<br>● HTML и CSS являются неправильными ответами, так как это языки разметки и стилей для веб-страниц, не предназначенные для программирования логики и алгоритмов поиска.<br>**Задание 3**<br>Вопрос: Установите последовательность шагов для реализации алгоритма поиска в ширину (BFS) с использованием очереди.<br>1. Проверить, является ли извлеченное состояние целевым.<br>2. Поместить все непосещенные состояния-потомки текущего |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | состояния в очередь.<br>3. Извлечь состояние из начала очереди.<br>4. Поместить начальное состояние в очередь.<br>5. Если очередь пуста, завершить алгоритм (решение не найдено).<br>6. Пометить извлеченное состояние как посещенное.<br>Правильный порядок: 4, 3, 6, 1, 2, 5<br>Пояснение: Алгоритм начинается с инициализации очереди начальным состоянием (4). Затем в цикле извлекается состояние из очереди (3), помечается как посещенное (6) и проверяется на цель (1). Если цель не достигнута, в очередь добавляются все его непосещенные потомки (2). Цикл продолжается, пока очередь не опустеет (5), что означает отсутствие решения.<br><br>**Задание 4**<br>Вопрос: Расположите в правильном порядке этапы разработки программного модуля, реализующего алгоритм поиска пути, с использованием современных инструментов.<br>1. Реализация алгоритма поиска (например, A*) на выбранном языке (напр., Python).<br>2. Проектирование структуры данных для представления состояния и пространства поиска.<br>3. Тестирование и отладка алгоритма на тестовых данных.<br>4. Выбор языка программирования и программной среды (IDE, напр., PyCharm).<br>5. Визуализация процесса поиска и результатов для верификации.<br>Правильный порядок: 4, 2, 1, 3, 5<br>Пояснение: Процесс начинается с выбора инструментов (4). Далее проектируется модель данных (2), которая затем реализуется в коде (1). После реализации следует этап тестирования логики (3). Финальный, необязательный, но важный этап — визуализация (5) для наглядной проверки корректности работы алгоритма.<br><br>**Задание 5**<br>Установите соответствие между алгоритмом поиска в пространстве |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | состояний и его ключевой характеристикой.<br><br>Список А (Алгоритм):<br>1. Поиск в ширину (BFS)<br>2. Поиск в глубину (DFS)<br>3. A* (A-star)<br>4. Жадный поиск по первому наилучшему соответствию<br><br>Список Б (Ключевая характеристика):<br>A. Использует эвристическую функцию для оценки стоимости пути через узел, что позволяет найти оптимальный путь.<br>B. Систематически исследует все узлы на текущей глубине, прежде чем перейти на следующую. Гарантирует нахождение кратчайшего пути в ненагруженном графе<br>C. Использует стратегию "последний пришел — первый ушел" (LIFO), что может привести к быстрому нахождению решения, но не гарантирует его оптимальность.<br>D. Использует эвристическую функцию для выбора следующего узла, но игнорирует стоимость уже пройденного пути.<br><br>Ответ: 1B, 2C, 3A, 4D<br><br>Пояснение: Соответствие строится на фундаментальных свойствах этих алгоритмов. BFS известен своим свойством находить кратчайший путь. DFS использует стек (LIFO). A* является оптимальным за счет суммы пройденного пути и эвристики. Жадный алгоритм фокусируется только на эвристике до цели.<br><br>**Задание 6**<br>Вопрос: Какой алгоритм поиска в пространстве состояний гарантированно находит кратчайший путь в ненагруженном графе (где все ребра имеют одинаковую стоимость), используя структуру данных "очередь"?<br>1. Поиск в глубину (DFS) |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 2. Поиск в ширину (BFS)<br>3. Жадный поиск по первому наилучшему соответствию<br>4. Поиск с ограничением глубины<br>Правильный ответ: 2<br>Пояснение:<br>● Поиск в ширину (BFS) является правильным ответом, так как он исследует все состояния на расстоянии 1, затем на расстоянии 2 и т.д., поэтому первое найденное целевое состояние будет достигнуто по кратчайшему пути.<br>● Поиск в глубину (DFS) является неправильным ответом, так как он может "уйти" далеко вглубь по одному направлению и найти не самый короткий путь.<br>● Жадный поиск по первому наилучшему соответствию является неправильным ответом, так как он руководствуется только эвристикой до цели, а не длиной пройденного пути.<br>● Поиск с ограничением глубины является неправильным ответом, так как это модификация DFS, которая также не гарантирует нахождение кратчайшего пути.<br><br>**Задание 7**<br>Вопрос: Какая из перечисленных российских операционных систем, включенных в реестр Минцифры, может быть использована в качестве надежной платформы для развертывания серверной части информационной системы, реализующей сложные алгоритмы поиска и работающей с базами данных?<br>1. Microsoft Windows Server<br>2. Astra Linux<br>3. Ubuntu Server<br>4. macOS Server<br>Правильный ответ: 2<br>Пояснение:<br>● Astra Linux является правильным ответом, так как это отечественная ОС, специально разработанная для применения в государственных структурах и силовых ведомствах, имеющая |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | все необходимые сертификаты и надежную систему защиты.<br>• Microsoft Windows Server является неправильным ответом, так как это продукт американской корпорации Microsoft.<br>• Ubuntu Server является неправильным ответом, так как это дистрибутив Linux, разрабатываемый международной компанией Canonical, не являющийся отечественной разработкой.<br>• macOS Server является неправильным ответом, так как это продукт американской компании Apple, к тому же в настоящее время его развитие свернуто<br>З**адание 8**<br>Вопрос: Какая структура данных, работающая по принципу "первый пришел — первый ушел" (FIFO), является основной для реализации алгоритма поиска в ширину (BFS)? Ответ состоит из одного слова.<br>Правильный ответ: Очередь<br>Пояснение: Очередь (Queue) — это абстрактный тип данных, который точно следует дисциплине FIFO. Это свойство критически важно для BFS, который должен обрабатывать состояния в том порядке, в котором они были обнаружены, что и обеспечивает нахождение кратчайшего пути.<br><br>**Задание 9**<br>Вопрос: Какой язык программирования, использующий механизм автоматического перебора с возвратами (backtracking) для поиска решений, основанных на заданных фактах и правилах, является классическим выбором для задач, формулируемых в терминах логики? Ответ состоит из одного слова.<br>Правильный ответ: Prolog<br>Пояснение: Prolog — это язык, чья вычислительная модель основана на автоматическом поиске с возвратами и унификации. Программа на Prolog описывает проблему в виде логических утверждений, а интерпретатор автоматически осуществляет поиск в пространстве возможных решений, что напрямую соотносится с поиском в пространстве состояний |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | |
| **Тема 2.2** Эвристический поиск | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 1.1**<br>Вопрос: Какие из перечисленных утверждений справедливы для эвристических функций, используемых в алгоритмах поиска, таких как A*?<br>**1.** Эвристическая функция должна быть допустимой (admissible), то есть никогда не переоценивать стоимость достижения цели.<br>**2.** Эвристическая функция может быть произвольной, ее точность не влияет на результат.<br>**3.** Чем точнее эвристическая функция (чем ближе к реальной стоимости), тем меньше узлов будет рассмотрено алгоритмом.<br>**4.** Эвристическая функция должна быть всегда равна нулю для корректной работы.<br>Правильные ответы: 1, 3<br>Пояснение:<br>● Утверждение 1 является правильным, так как допустимость — ключевое свойство для гарантии оптимальности алгоритма A*.<br>● Утверждение 3 является правильным, так как более точная эвристика лучше направляет поиск, уменьшая пространство перебора.<br>● Утверждение 2 является неправильным, так как качество эвристики напрямую влияет на эффективность поиска; плохая эвристика может свести алгоритм к полному перебору.<br>● Утверждение 4 является неправильным, так как эвристика, тождественно равная нулю, является допустимой, но превращает A* в алгоритм Дейкстры, что неэффективно для поиска пути к конкретной цели.<br><br>**Задание 1.2**<br>Вопрос: Для практической реализации и тестирования алгоритмов эвристического поиска (например, A*) на графах большого размера наиболее целесообразно использовать следующие инструменты и технологии:<br>1. Язык C++ для высокопроизводительных вычислений. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 2. Язык Python с библиотеками (например, heapq для приоритетной очереди, networkx для работы с графами).<br>3. Система управления базами данных (например, PostgreSQL) для хранения графа и выполнения рекурсивных запросов.<br>4. Графический редактор, такой как Adobe Photoshop.<br>Правильные ответы: 1, 2, 3<br>Пояснение:<br>● C++ является правильным ответом, так как он обеспечивает высокую производительность, что критично для работы с большими графами.<br>● Python с библиотеками является правильным ответом, так как он позволяет быстро прототипировать и тестировать алгоритмы благодаря обширным библиотекам для работы с данными и структурами.<br>● Система управления базами данных является правильным ответом, так как для очень больших графов, которые не помещаются в оперативную память, хранение в БД и использование рекурсивных запросов или хранимых процедур является практичным решением.<br>● Графический редактор является неправильным ответом, так как это инструмент для работы с изображениями, а не для<br>**Задание 2.1**<br>Вопрос: Установите последовательность шагов алгоритма A* на одном шаге его работы (после инициализации).<br>1. Если очередь пуста, завершить алгоритм (путь не найден).<br>2. Поместить начальное состояние в приоритетную очередь с приоритетом f(start) = g(start) + h(start).<br>3. Извлечь из очереди состояние с наименьшим значением f(n).<br>4. Если извлеченное состояние является целевым, завершить алгоритм успешно.<br>5. Для каждого соседа извлеченного состояния рассчитать g(n) и f(n) и добавить его в очередь (или обновить приоритет, если найден более короткий путь).<br>Правильный порядок: 2, 3, 4, 1, 5 |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Пояснение: Алгоритм начинается с инициализации очереди начальным узлом (2). Далее в цикле извлекается узел с наименьшей общей стоимостью f(n) (3). Этот узел проверяется на соответствие цели (4). Если цель не достигнута и очередь не пуста (1), обрабатываются все соседи текущего узла (5). Проверка на пустоту очереди (1) логически следует после попытки извлечения узла (3), но в корректном алгоритме она является условием продолжения цикла, следующим за успешным извлечением и проверкой на цель.<br><br>**Задание 2.2**<br>Вопрос: Расположите в логическом порядке этапы разработки программного обеспечения, реализующего эвристический поиск в распределенной системе.<br>1. Реализация核心 алгоритма (напр., A*) на серверной стороне.<br>2. Проектирование схемы базы данных для хранения графа состояний.<br>3. Выбор стека технологий (язык сервера, СУБД, фреймворк для API).<br>4. Создание клиентского приложения (веб- или мобильного) для отправки запросов и визуализации результатов.<br>5. Разработка REST API для взаимодействия между клиентом и сервером.<br>Правильный порядок: 3, 2, 1, 5, 4<br>Пояснение: Разработка начинается с выбора технологий (3). Затем проектируется структура хранения данных (2). После этого реализуется основная бизнес-логика — алгоритм поиска (1). Далее создается интерфейс для взаимодействия с этой логикой извне (5). Финальный этап — разработка клиента, который использует этот API (4).<br>**Задание 3.1**<br>Установите соответствие между алгоритмом поиска и его ключевым свойством.<br><br>Список А (Алгоритм): |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 1. A*<br>2. Жадный поиск по первому наилучшему соответствию<br>3. Алгоритм Дейкстры<br>4. Поиск в ширину<br><br>Список Б (Свойство):<br>A. Оптимален и эффективен при использовании допустимой и монотонной эвристики<br>B. Использует только эвристическую функцию h(n), игнорируя стоимость достижения узла g(n).<br>C. Оптимален для поиска кратчайшего пути без использования эвристики (фактически, A* с h(n) = 0).<br>D. Является частным случаем алгоритма Дейкстры для графов с единичной стоимостью ребер.<br><br>Ответ: 1A, 2B, 3C, 4D<br><br>Пояснение: A* сочетает стоимость пройденного пути g(n) и эвристику h(n). Жадный поиск использует только h(n). Алгоритм Дейкстры можно рассматривать как A*, где эвристика равна нулю. BFS — это Дейкстра, примененная к графу, где все ребра имеют одинаковую стоимость<br><br>**Задание 3.2**<br>Установите соответствие между компонентом/концепцией в разработке ПО для эвристического поиска и примером технологии или инструмента.<br><br>Список А (Компонент/Концепция):<br>1. Язык для высокопроизводительной реализации вычислительно сложных алгоритмов<br>2. Инструмент для быстрого прототипирования и визуализации графов<br>3. Система для хранения большого графа и выполнения запросов к нему<br>4. Отечественная ОС для развертывания серверной части системы |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Список Б (Технология/Инструмент):<br>A. Python с библиотекой heapq<br>B. C++ со STL (priority_queue)<br>C. СУБД PostgreSQL (с рекурсивными CTE)<br>D. Astra Linux<br><br>Ответ: 1B, 2A, 3C, 4D<br><br>Пояснение: C++ и его стандартная библиотека предоставляют эффективные структуры данных для высоконагруженных систем. Python известен скоростью разработки и богатыми библиотеками. PostgreSQL позволяет хранить и обрабатывать огромные объемы структурированных данных, включая графы. Astra Linux — это сертифицированная российская ОС для защищенных серверов.<br><br>**Задание 4.1**<br>Вопрос: Какое свойство эвристической функции h(n) является необходимым и достаточным для гарантии того, что алгоритм A* найдет оптимальное (самое короткое) решение?<br>1. h(n) должна быть монотонной.<br>2. h(n) должна быть допустимой (admissible).<br>3. h(n) должна быть равной нулю в целевом состоянии.<br>4. h(n) должна всегда быть больше нуля.<br>Правильный ответ: 2<br>Пояснение:<br>● "h(n) должна быть допустимой (admissible)" является правильным ответом. Это ключевая теорема для A*: если h(n) никогда не переоценивает действительную стоимость достижения цели, то A* гарантированно найдет оптимальный путь.<br>● "h(n) должна быть монотонной" является неправильным ответом. Монотонность — это более строгое условие, которое гарантирует, что функция f(n) не убывает вдоль любого пути, и упрощает реализацию A* (не требуется перерасчет |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | приоритетов), но оптимальность гарантируется уже допустимостью.<br>● "h(n) должна быть равной нулю в целевом состоянии" является неправильным ответом. Это верно для любой осмысленной эвристики, но не является достаточным условием для оптимальности.<br>● "h(n) должна всегда быть больше нуля" является неправильным ответом. Это неверно, так как для узла, являющегося целью, h(n) должна быть равна 0.<br><br>**Задание 4.2**<br>Вопрос: Какая структура данных является наиболее эффективной для реализации алгоритма A*, так как позволяет быстро извлекать узел с наименьшей общей стоимостью f(n)?<br>1. Стек (Stack)<br>2. Очередь (Queue)<br>3. Приоритетная очередь (Priority Queue)<br>4. Связный список (Linked List)<br>Правильный ответ: 3<br>Пояснение:<br>● Приоритетная очередь (Priority Queue) является правильным ответом, так как она оптимизирована для вставки элементов и извлечения элемента с наивысшим (или наименьшим) приоритетом, что идеально соответствует потребностям A*.<br>● Стек (Stack) и Очередь (Queue) являются неправильными ответами, так как они извлекают элементы строго в порядке LIFO и FIFO соответственно, не учитывая приоритет, основанный на f(n).<br>● Связный список (Linked List) является неправильным ответом, так как поиск элемента с минимальным значением f(n) в несортированном связном списке требует линейного времени O(n), что крайне неэффективно.<br>**Задание 5.1**<br>Вопрос: В алгоритме A* полная стоимость узла вычисляется по |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | формуле f(n) = g(n) + h(n). Что представляет собой функция g(n)? Ответ должен быть дан одним словом или короткой фразой.<br>Правильный ответ: Стоимость достижения узла n (от начального узла).<br>Пояснение: Функция g(n) — это фактическая стоимость пути от стартового узла до узла n. Она представляет собой сумму затрат (например, расстояний, временных интервалов) всех ребер, пройденных по оптимальному на данный момент пути до n.<br><br>**Задание 5.2**<br>Вопрос: Какая абстрактная структура данных, часто реализуемая на основе двоичной кучи (binary heap), используется в алгоритме A* для хранения и эффективного извлечения узла с наименьшей оценочной стоимостью f(n)? Ответ состоит из двух слов.<br>Правильный ответ: Приоритетная очередь<br>Пояснение: Приоритетная очередь — это структура данных, которая позволяет вставлять элементы с приоритетом и извлекать элемент с наивысшим (или наименьшим) приоритетом. В контексте A* приоритетом является значение f(n). Реализация на основе двоичной кучи обеспечивает эффективность операций вставки и извлечения минимума за O(log n) |
| **Тема 2.3** Управление поиском и его реализация в пространстве состояний | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 1.1**<br>Вопрос: Какие из перечисленных стратегий управления поиском в пространстве состояний являются систематическими (полными), то есть гарантируют нахождение решения, если оно существует?<br>**1.** Случайный поиск<br>**2.** Поиск в ширину (BFS)<br>**3.** Поиск с возвратом (Backtracking)<br>**4.** Поиск по первому наилучшему соответствию (жадный)<br>Правильные ответы: 2, 3<br>Пояснение:<br>● Поиск в ширину (BFS) является правильным ответом, так как он систематически исследует все узлы уровня за уровнем, гарантируя нахождение пути. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | • Поиск с возвратом (Backtracking) является правильным ответом, так как это метод полного перебора, который последовательно проверяет все возможные варианты, возвращаясь при тупиках.<br>• Случайный поиск является неправильным ответом, так как он не систематичен и может бесконечно блуждать в пространстве состояний, не находя решение.<br>• Поиск по первому наилучшему соответствию (жадный) является неправильным ответом, так как он руководствуется локально оптимальным выбором и может зациклиться или попасть в локальный оптимум, не гарантируя нахождение решения.<br><br>**Задание 1.2**<br>Вопрос: Какие из перечисленных технологий и языков программирования эффективно подходят для реализации систем, осуществляющих поиск в больших пространствах состояний (например, для задач планирования или анализа графов)?<br>**1.** Язык Prolog с его встроенным механизмом унификации и поиска с возвратами.<br>**2.** Язык Python с библиотеками для работы с графами (например, networkx) и структурами данных.<br>3. Язык разметки HTML.<br>4. Системы управления базами данных (например, PostgreSQL) с поддержкой рекурсивных запросов (WITH RECURSIVE).<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>• Prolog является правильным ответом, так как его вычислительная модель изначально предназначена для логического вывода и поиска с возвратами в пространстве, определяемом фактами и правилами.<br>• Python является правильным ответом, так как он позволяет гибко реализовывать различные алгоритмы поиска и имеет богатый набор библиотек для работы с графами и данными.<br>• Системы управления базами данных являются правильным ответом, так как они позволяют хранить огромные графы |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | состояний на диске и эффективно обходить их с помощью рекурсивных запросов.<br>● HTML является неправильным ответом, так как это язык разметки веб-страниц и он не предназначен для программирования логики поиска.<br>**Задание 2.1**<br>Вопрос: Установите правильную последовательность этапов в обобщенном алгоритме поиска в пространстве состояний.<br>1. Проверка цели: Проверить, является ли текущее состояние целевым.<br>2. Выбор состояния: Выбрать следующее состояние для анализа из множества пограничных.<br>3. Инициализация: Определить начальное состояние и добавить его в множество пограничных.<br>4. Расширение: Применить все возможные операторы к текущему состоянию, чтобы получить новые состояния, и добавить их в множество пограничных.<br>5. Проверка на неудачу: Если множество пограничных пусто, завершить с неудачей.<br>Правильный порядок: 3, 2, 1, 4, 5<br>Пояснение: Алгоритм начинается с инициализации (3). Затем в цикле выбирается состояние из пограничных (2) и проверяется на соответствие цели (1). Если цель не достигнута, состояние расширяется (4). Цикл продолжается, пока не будет найдено решение или не будет исчерпано пространство поиска (5).<br><br>**Задание 2.2**<br>Вопрос: Расположите в логическом порядке этапы настройки серверной среды для развертывания высокопроизводительной системы поиска, использующей реляционную базу данных.<br>1. Установка и настройка СУБД (например, настройка индексов для ускорения запросов).<br>2. Разработка и размещение кода приложения, реализующего алгоритм поиска. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 3. Выбор и установка операционной системы на сервере.<br>4. Наполнение базы данных тестовыми данными (графом состояний).<br>5. Создание схемы базы данных (таблицы для узлов и ребер).<br>Правильный порядок: 3, 1, 5, 4, 2<br>Пояснение: Сначала развертывается базовое программное обеспечение — ОС (3). Затем устанавливается и базово настраивается СУБД (1). После этого создается структура для хранения данных (5), которая затем наполняется информацией (4). Финальный шаг — размещение прикладного кода, который будет использовать эту базу (2)<br>**Задание 3.1**<br>Установите соответствие между стратегией поиска и ключевой структурой данных, которая для нее используется.<br><br>Список А (Стратегия поиска):<br>1. Поиск в ширину (BFS)<br>2. Поиск в глубину (DFS)<br>3. Поиск с возвратом (Backtracking)<br>4. Поиск по A*<br><br>Список Б (Структура данных):<br>A. Стек (Stack)<br>B. Очередь (Queue)<br>C. Множество (Set) для отслеживания посещенных состояний<br>D. Приоритетная очередь (Priority Queue)<br><br>Ответ: 1B, 2A, 3A, 4D<br><br>Пояснение: BFS обрабатывает состояния в порядке поступления (очередь). DFS и Backtracking идут "вглубь", обрабатывая последнее добавленное состояние (стек). A* извлекает состояние с наивысшим приоритетом (приоритетная очередь). Структура C (множество) является вспомогательной для многих стратегий, но не основной управляющей структурой. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | **Задание 3.2**<br>Установите соответствие между задачей управления поиском и технологическим решением для ее реализации.<br><br>Список А (Задача):<br>1. Хранение очень большого графа состояний, не помещающегося в оперативную память<br>2. Быстрое прототипирование и визуализация алгоритма поиска на графе<br>3. Обеспечение максимальной скорости вычислений для сложного поиска<br>4. Создание отказоустойчивого сервера для системы поиска в госсекторе<br><br>Список Б (Технологическое решение):<br>A. Реализация алгоритма на C++<br>B. Использование СУБД (напр., PostgreSQL)<br>C. Язык Python с библиотеками (NumPy, NetworkX)<br>D. Развертывание на отечественной ОС (Astra Linux)<br><br>Ответ: 1B, 2C, 3A, 4D<br><br>Пояснение: Для больших данных, не помещающихся в RAM, используют СУБД. Python идеален для быстрого создания прототипов. C++ обеспечивает высокую производительность. Astra Linux является стандартом для защищенных серверов в госсекторе РФ<br>**Задание 4.1**<br>Вопрос: Какая стратегия поиска характеризуется тем, что всегда выбирает для расширения самый глубокий узел в текущем направлении, используя структуру данных "стек"?<br>1. Поиск в ширину (Breadth-First Search)<br>2. Поиск в глубину (Depth-First Search)<br>3. Поиск по A* |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 4. Двунаправленный поиск (Bidirectional Search)<br>Правильный ответ: 2<br>Пояснение:<br>● Поиск в глубину (Depth-First Search) является правильным ответом, так как его стратегия — идти "вглубь" графа, пока это возможно, а при достижении тупика возвращаться (backtrack), что реализуется с помощью стека.<br>● Поиск в ширину (Breadth-First Search) является неправильным ответом, так как он использует очередь и исследует граф "слой за слоем".<br>● Поиск по A* является неправильным ответом, так как он использует приоритетную очередь и эвристическую функцию для выбора направления.<br>● Двунаправленный поиск (Bidirectional Search) является неправильным ответом, так как это общая стратегия, которая может использовать либо BFS, либо DFS внутри себя.<br><br>**Задание 4.2**<br>Вопрос: Какая из перечисленных российских систем управления базами данных, основанная на открытом коде PostgreSQL, является предпочтительным выбором для хранения и обработки данных в государственных информационных системах, требующих высокой надежности и производительности?<br>1. MySQL<br>2. SQLite<br>3. Postgres Pro<br>4. Oracle Database<br>Правильный ответ: 3<br>Пояснение:<br>● Postgres Pro является правильным ответом, так как это российская коммерческая СУБД, созданная на основе PostgreSQL, обладающая улучшенной производительностью и входящая в реестр отечественного ПО.<br>● MySQL является неправильным ответом, так как это СУБД, |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | принадлежащая американской компании Oracle Corporation.<br>• SQLite является неправильным ответом, так как это встраиваемая легковесная БД, не предназначенная для высоконагруженных серверных систем.<br>• Oracle Database является неправильным ответом, так как это проприетарная СУБД американской корпорации Oracle.<br>**Задание 5.1**<br>Вопрос: Как называется стратегия поиска, при которой два параллельных поисковых процесса запускаются одновременно: один от начального состояния, а другой от целевого, с целью обнаружить совпадение на их границах? Ответ состоит из двух слов.<br>Правильный ответ: Двунаправленный поиск<br>Пояснение: Двунаправленный поиск (Bidirectional Search) — это стратегия, предназначенная для уменьшения сложности по времени и памяти. Вместо исследования всего пространства от начала до конца, она ведет два поиска навстречу друг другу, что может значительно сократить общее количество рассмотренных состояний.<br><br>**Задание 5.2**<br>Вопрос: Какая структура данных, работающая по принципу "первым пришел — первым ушел" (FIFO), является фундаментальной для реализации алгоритма поиска в ширину (BFS)? Ответ состоит из одного слова.<br>Правильный ответ: Очередь<br>Пояснение: Очередь (Queue) — это абстрактный тип данных, который строго следует дисциплине FIFO. Это свойство критически важно для BFS, который должен обрабатывать состояния в том порядке, в котором они были обнаружены, что и обеспечивает нахождение кратчайшего пути в ненагруженном графе. |
| **Тема 2.4** Задача планировани я | **ПК-7** Способен осуществлять проектирование программных | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 1.1**<br>Вопрос: Какие из перечисленных компонентов являются критически важными для формализации задачи планирования в пространстве состояний? |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | интерфейсов | | **1.** Множество начальных состояний<br>**2.** Множество целевых состояний (или условие цели)<br>**3.** Множество действий (операторов) с их предусловиями и эффектами<br>**4.** Цветовая схема пользовательского интерфейса<br>Правильные ответы: 1, 2, 3<br>Пояснение:<br>● Множество начальных состояний является правильным ответом, так как определяет точку отсчета для плана.<br>● Множество целевых состояний является правильным ответом, так как определяет желаемый результат, к которому должен привести план.<br>● Множество действий является правильным ответом, так как описывает возможные преобразования состояний, из которых строится план.<br>● Цветовая схема пользовательского интерфейса является неправильным ответом, так как это элемент дизайна, не влияющий на логическую формализацию задачи планирования.<br><br>**Задание 1.2**<br>Вопрос: Какие из перечисленных технологий и инструментов могут быть эффективно использованы для прототипирования и реализации алгоритмов планирования (например, алгоритмов forward или backward search)?<br>**1.** Язык планирования PDDL (Planning Domain Definition Language)<br>**2.** Фреймворки для работы с онтологиями (например, OWL API)<br>**3.** Язык Python с библиотеками для символьных вычислений и работы с графами<br>**4.** Табличный процессор Microsoft Excel<br>Правильные ответы: 1, 2, 3<br>Пояснение:<br>● PDDL является правильным ответом, так как это стандартный язык для формального описания доменов и задач планирования.<br>● Фреймворки для работы с онтологиями являются правильным |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | ответом, так как онтологии могут формально описывать домен планирования, а рассуждения на их основе могут быть использованы для проверки предусловий и вычисления эффектов.<br>● Python является правильным ответом, так как его гибкость и богатый набор библиотек делают его идеальным для быстрой реализации поисковых алгоритмов и работы с графами состояний.<br>● Microsoft Excel является неправильным ответом, так как это инструмент для табличных вычислений и визуализации данных, а не для программирования сложных логических алгоритмов поиска.<br>**Задание 2.1**<br>Вопрос: Установите последовательность шагов в алгоритме прямого (forward) поиска планирования от начального состояния к целевому.<br>1. Проверить, удовлетворяет ли текущее состояние целевому условию.<br>2. Выбрать применимое действие (оператор), чьи предусловия выполнены в текущем состоянии.<br>3. Применить эффекты выбранного действия к текущему состоянию, получив новое состояние.<br>4. Начать с исходного состояния.<br>5. Добавить новое состояние в пространство поиска.<br>Правильный порядок: 4, 2, 3, 5, 1<br>Пояснение: Алгоритм начинается с исходного состояния (4). В цикле выбирается действие, применимое в текущей ситуации (2). Это действие применяется, изменяя состояние мира (3). Новое состояние фиксируется (5). Затем проверяется, является ли это состояние целевым (1). Процесс повторяется для новых состояний.<br><br>**Задание 2.2**<br>Вопрос: Расположите в логическом порядке этапы разработки программного комплекса для автоматического планирования с использованием современных технологий. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 1. Реализация алгоритма поиска плана (напр., на Python или Java).<br>2. Формализация домена планирования и конкретной задачи (напр., на языке PDDL).<br>3. Визуализация найденного плана и его верификация.<br>4. Выбор или разработка синтаксического анализатора (parser) для входного языка описания домена.<br>5. Интеграция решателя планирования в более крупную информационную систему.<br>Правильный порядок: 2, 4, 1, 3, 5<br>Пояснение: Сначала необходимо формально описать предметную область и задачу (2). Затем создать инструмент, который сможет прочитать это описание (4). После этого реализуется ядро системы — алгоритм, работающий с внутренним представлением домена (1). Далее следует этап визуализации и проверки результатов для пользователя (3). Финальный этап — встраивание модуля планирования в реальную систему (5).<br>**Задание 3.1**<br><br>Установите соответствие между этапом работы системы планирования и рекомендуемой технологией или инструментом для его реализации.<br><br>Список А (Этап):<br>1. Формальное описание планирования<br>2. Хранение онтологий и фактов о состоянии в постоянной памяти<br>3. Высокопроизводительные вычисления для сложного поиска в пространстве состояний<br>4. Быстрое прототипирование и визуализация графа состояний<br><br>Список Б (Технология/Инструмент):<br>A. Язык PDDL<br>B. Реляционная СУБД (напр., Postgres Pro)<br>C. Язык C++<br>D. Язык Python с библиотекой Matplotlib/NetworkX |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Ответ: 1A, 2B, 3C, 4D<br><br>Пояснение: PDDL — это стандарт для описания доменов планирования. СУБД надежно хранят структурированные данные (онтологии, состояния). C++ обеспечивает скорость для ресурсоемких вычислений. Python является лучшим выбором для исследований и визуализации<br>**Задание 4.1**<br>Вопрос: Какой из перечисленных алгоритмов является классическим примером подхода к планированию, который работает в прямом направлении от начального состояния, применяя действия, пока не будет достигнуто целевое состояние?<br>1. Алгоритм регрессионного планирования (обратный поиск)<br>2. Алгоритм STRIPS<br>3. Алгоритм частичного порядка планирования (POP)<br>4. Алгоритм планирования как удовлетворения ограничений (CSP)<br>Правильный ответ: 2<br>Пояснение:<br>● Алгоритм STRIPS является правильным ответом. Хотя STRIPS — это прежде всего *представление* для действий, алгоритм, использующий это представление, обычно реализуется как прямой поиск в пространстве состояний.<br>● Алгоритм регрессионного планирования является неправильным ответом, так как это обратный поиск, который стартует от цели и применяет действия в обратном порядке.<br>● Алгоритм частичного порядка планирования (POP) является неправильным ответом, так как он работает не в пространстве состояний, а в пространстве планов, связывая действия через угрозы и порядковые ограничения.<br>● Алгоритм планирования как удовлетворения ограничений (CSP) является неправильным ответом, так как это другой подход, где задача преобразуется в задачу удовлетворения ограничений.<br><br>**Задание 4.2**<br>Вопрос: Какая из перечисленных отечественных операционных систем, |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | включенных в реестр ПО Минцифры РФ, может быть использована как безопасная и надежная платформа для развертывания сервера, выполняющего критически важные задачи автоматического планирования?<br>1. Windows 11<br>2. Ubuntu Linux<br>3. Astra Linux<br>4. macOS<br>Правильный ответ: 3<br>Пояснение:<br>● Astra Linux является правильным ответом, так как это российская ОС, специально разработанная для создания защищенных автоматизированных систем, отвечающая требованиям регуляторов.<br>● Windows 11 и macOS являются неправильными ответами, так как это проприетарные ОС американских компаний.<br>● Ubuntu Linux является неправильным ответом, так как это международный проект, спонсируемый британской компанией Canonical, и он не является отечественной разработкой.<br>**Задание 5.1**<br>Вопрос: Как называется стандартный язык, используемый в исследованиях искусственного интеллекта для формального описания домена планирования (множества действий) и конкретной задачи планирования (начальное состояние и цель)? Ответ состоит из аббревиатуры из 4 букв.<br>Правильный ответ: PDDL<br>Пояснение: PDDL (Planning Domain Definition Language) — это де-факто стандартный язык для описания задач планирования. Он позволяет отдельно описать домен (типы, предикаты, действия) и задачу (объекты, начальное состояние, цель).<br><br>**Задание 5.2**<br>Вопрос: В контексте описания действия в задаче планирования (например, в PDDL или STRIPS), каким термином называется условие, |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | которое должно быть истинным в текущем состоянии, чтобы действие могло быть выполнено? Ответ состоит из одного слова.<br>Правильный ответ: Предусловие<br>Пояснение: Предусловие (Precondition) — это обязательная часть описания любого действия. Это множество фактов (литералов), которые должны присутствовать в состоянии, чтобы действие было применимо. Без корректно заданных предусловий план может быть логически невыполнимым. |
| **Тема 2.5** Планирование и осуществление действий в реальном мире | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 1.1**<br>Вопрос: Какие из перечисленных технологий и подходов используются для интеграции систем автоматического планирования с реальным миром через базы данных и программные интерфейсы?<br>**1.** REST API для получения текущего состояния мира от сенсоров и передачи плана исполнительным системам<br>**2.** Пакет программ Microsoft Office для визуализации отчетов<br>**3.** Системы управления базами данных (например, PostgreSQL) для хранения моделей мира, онтологий и истории выполненных планов<br>**4.** Язык разметки HTML для создания пользовательских форм ввода<br>Правильные ответы: 1, 3<br>Пояснение:<br>● REST API является правильным ответом, так как это стандартный способ взаимодействия распределенных систем, позволяющий планирующему агенту получать актуальные данные и отдавать команды.<br>● Системы управления базами данных являются правильным ответом, так как они обеспечивают надежное, постоянное хранение сложных структур данных (состояний, онтологий, планов), необходимых для работы в реальном мире.<br>● Пакет Microsoft Office является неправильным ответом, так как это инструмент для офисной работы, а не для интеграции программных систем в реальном времени. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | ● Язык разметки HTML является неправильным ответом, так как он предназначен для создания веб-страниц, а не для программирования логики взаимодействия с реальным миром.<br><br>**Задание 1.2**<br>Вопрос: При реализации системы, которая не только строит планы, но и исполняет их в динамической среде, критически важными являются следующие компоненты:<br>**1.** Модуль мониторинга для отслеживания изменений в мире и обнаружения отклонений от ожидаемого состояния<br>**2.** Механизм перепланирования для корректировки плана в случае непредвиденных событий<br>**3.** Статическая база знаний, которая никогда не обновляется после начала работы<br>**4.** Исполнительные механизмы (например, драйверы оборудования или вызовы внешних API) для выполнения запланированных действий<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● Модуль мониторинга является правильным ответом, так как в реальном мире состояние постоянно меняется, и система должна это отслеживать.<br>● Механизм перепланирования является правильным ответом, так как первоначальный план может стать неактуальным, и система должна уметь адаптироваться.<br>● Исполнительные механизмы являются правильным ответом, так как план является абстракцией, которая должна быть преобразована в конкретные команды для роботов, сервисов или других систем.<br>● Статическая база знаний является неправильным ответом, так как работа в реальном мире требует постоянного обновления знаний о текущей ситуации<br>**Задание 3.1**<br>Установите соответствие между задачей в контексте планирования |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | действий в реальном мире и технологическим решением для ее реализации.<br><br>Список А (Задача):<br>1. Непрерывное получение данных о состоянии среды<br>2. Надежное хранение истории планов и состояний для анализа<br>3. Преобразование абстрактного плана в конкретные команды для оборудования<br>4. Запуск и обеспечение отказоустойчивой работы всего программного стека<br><br>Список Б (Технологическое решение):<br>A. Датчики (сенсоры) и система сбора данных (SCADA)<br>B. Система управления базами данных (СУБД)<br>C. Исполнительные механизмы (актуаторы) и их драйверы<br>D. Операционная система реального времени (ОСРВ)<br><br>Ответ: 1A, 2B, 3C, 4D (Отказоустойчивость — ОСРВ)<br><br>Пояснение: Для восприятия мира нужны сенсоры (1-A). Для долговременной памяти — СУБД (2-B). Для воздействия на мир — актуаторы (3-C). Для гарантированного выполнения программ в строгие временные рамки требуется специализированная ОСРВ (4-D).<br><br>**Задание 3.2**<br>Установите соответствие между компонентом архитектуры автономной системы и его описанием.<br><br>Список А (Компонент):<br>1. Модуль восприятия<br>2. Модуль планирования<br>3. Модуль управления<br>4. Модуль исполнения |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Список Б (Описание):<br>A. Отвечает за генерацию последовательности действий для достижения цели.<br>B. Преобразует абстрактные действия плана в низкоуровневые команды для конкретного оборудования<br>C. Обеспечивает взаимодействие с пользователем или другими системами.<br>D. Обрабатывает сырые данные с датчиков для построения согласованной модели текущего состояния мира.<br><br>Ответ: 1D, 2A, 3C, 4B<br><br>Пояснение: Модуль восприятия превращает сигналы в знания (1-D). Планирование — это "мозг" системы (2-A). Управление часто связано с интерфейсами (3-C). Модуль исполнения — это "руки" системы (4-B)<br>**Задание 4.1**<br>Какой тип планирования является наиболее подходящим для работы в быстро меняющейся, непредсказуемой среде, где начальный план почти наверняка потребует частой корректировки?<br>1. Детальное планирование на много шагов вперед<br>2. Иерархическое планирование (Hierarchical Task Network planning)<br>3. Планирование в пространстве поведений (Behavior-based planning)<br>4. Планирование с повторным использованием планов (Case-Based Planning)<br><br>Ответ: 3<br><br>Пояснение:Планирование в пространстве поведений является правильным ответом, так как этот реактивный подход фокусируется на выборе следующего действия на основе текущего состояния, а не на составлении долгосрочного жесткого плана. Это позволяет быстро адаптироваться к изменениям. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | **Задание 4.2**<br>Какая из перечисленных российских СУБД, основанная на открытом коде и входящая в реестр Минцифры, предпочтительна для использования в качестве надежного хранилища данных для систем планирования в госсекторе и критической инфраструктуре?<br>1. MySQL<br>2. Postgres Pro<br>3. SQLite<br>4. Oracle Database<br><br>Ответ: 2<br><br>Пояснение: Postgres Pro является правильным ответом, так как это российская коммерческая СУБД, созданная на основе PostgreSQL. Она обладает улучшенной производительностью, безопасностью и поддерживается российскими разработчиками, что соответствует требованиям импортозамещения.<br>**Задание 5.1**<br>Как называется свойство реальной среды, которое описывает, могут ли действия агента изменять саму среду и каким образом? Это свойство требует от системы планирования постоянно обновлять свою внутреннюю модель мира.<br><br>Ответ: Динамичность<br><br>Пояснение: Динамичность — это ключевое свойство реальных сред, отличающее их от статических моделей. Оно означает, что состояние мира меняется со временем независимо от действий агента (например, погодные условия, движение других людей). Это напрямую влияет на архитектуру системы, требуя цикла "наблюдение-планирование-действие" и механизмов перепланирования.<br><br>**Задание 5.2** |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | В архитектуре автономных систем, работающих в реальном мире, как называется программный компонент, который напрямую взаимодействует с физическими датчиками, преобразуя сырые сигналы в структурированную информацию для модуля планирования?<br><br>Правильный ответ: Модуль восприятия<br><br>Пояснение: Модуль восприятия (Perception Module) — это критически важный компонент, который служит "органами чувств" системы. Он отвечает за фильтрацию шума, распознавание объектов, построение карты и в конечном итоге — за создание непротиворечивой и актуальной модели мира, на основе которой уже работает планировщик. Без него планировщик будет работать с устаревшими или неверными данными |
| **Тема 2.6** Интеллектуализация процедур поиска и управления | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 2.1**<br>Вопрос: Установите последовательность этапов в цикле «наблюдение-планирование-действие» для автономного агента, работающего в реальном мире.<br>1. Действие: Выполнить следующее действие из текущего плана с помощью исполнительных механизмов.<br>2. Наблюдение: Получить актуальные данные о состоянии среды через сенсоры или базу данных.<br>3. Планирование: На основе текущей модели мира и цели сгенерировать или скорректировать план.<br>4. Обновление модели мира: Интегрировать полученные данные во внутреннюю модель мира агента.<br>5. Оценка: Сравнить текущее состояние с ожидаемым. Обнаружить рассогласование.<br>Правильный порядок: 2, 4, 5, 3, 1<br>Пояснение: Цикл начинается с наблюдения за средой (2). Полученные данные используются для обновления внутренней модели мира агента (4). Затем агент оценивает, все ли идет по плану (5). Если выявлено рассогласование или план отсутствует, запускается модуль |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | планирования (3). После того как план создан или скорректирован, агент исполняет следующее действие (1).<br><br>**Задание 2.2**<br>Вопрос: Расположите в логическом порядке этапы развертывания отечественной программной платформы для управления автономным роботом.<br>1. Реализация и тестирование алгоритмов планирования и перепланирования на языке C++/Python.<br>2. Установка и настройка российской операционной системы (например, Astra Linux) на бортовом компьютере робота.<br>3. Интеграция с аппаратным обеспечением: написание драйверов для сенсоров и исполнительных механизмов.<br>4. Проектирование архитектуры программного обеспечения: модули восприятия, планирования, управления.<br>5. Полевые испытания и отладка системы в реальных условиях.<br>Правильный порядок: 4, 2, 1, 3, 5<br>Пояснение: Сначала проектируется общая архитектура системы (4). Затем выбирается и настраивается базовое программное обеспечение — ОС (2). Далее реализуется ключевая интеллектуальная составляющая — алгоритмы планирования (1). После этого обеспечивается связь интеллектуальной части с "физическим телом" робота (3). Финальный этап — проверка всей системы в реальных условиях (5)<br>**Задание 3.1**<br>Установите соответствие между методом интеллектуализации поиска/управления и его описанием.<br><br>Список А (Метод):<br>1. Эвристический поиск<br>2. Управление на основе моделей (Model-based Reasoning)<br>3. Система, основанная на правилах (Rule-Based System)<br>4. Многоагентные системы (Multi-Agent Systems)<br><br>Список Б (Описание):<br>А. Подход, при котором система строит модель окружающей среды и |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | планирует свои действия для достижения целей.<br>B. Использование оценочных функций для направления поиска в пространстве состояний в наиболее перспективную область.<br>C. Подход, при котором система хранит знания в виде "условие-действие" и активирует правила при выполнении условий.<br>D. Архитектура, где несколько интеллектуальных агентов взаимодействуют для решения сложных задач, координируя свои действия.<br><br>Ответ: 1B, 2A, 3C, 4D<br><br>Пояснение: Эвристики направляют поиск (1-B). Модель позволяет предсказывать последствия и планировать (2-A). Правила — это прямой способ кодирования знаний эксперта (3-C). Многоагентный подход решает задачи за счет распределения и кооперации (4-D).<br><br>**Задание 3.2**<br>Установите соответствие между компонентом интеллектуальной системы управления и рекомендуемой технологией для его реализации.<br><br>Список А (Компонент):<br>1. Платформа для развертывания серверной части системы<br>2. Хранилище для онтологий, моделей и истории состояний<br>3. Язык для быстрого прототипирования алгоритмов машинного обучения и анализа данных<br>4. Среда разработки для создания кода на C++/Python<br><br>Список Б (Отечественная технология):<br>A. Astra Linux<br>B. Postgres Pro<br>C. Python<br>D. JetBrains PyCharm<br><br>Ответ: 1A, 2B, 3C, 4D |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Пояснение: Astra Linux — это ключевая отечественная серверная ОС (1-A). Postgres Pro — СУБД для надежного хранения данных (2-B). Python является мировым стандартом для Data Science и ML (3-C). Продукты JetBrains являются стандартом для профессиональной разработки (4-D).<br>**Задание 4.1**<br>Вопрос: Какой тип планирования является наиболее подходящим для работы в быстро меняющейся, непредсказуемой среде, где начальный план почти наверняка потребует частой корректировки?<br>1. Детальное планирование на много шагов вперед<br>2. Иерархическое планирование (Hierarchical Task Network planning)<br>3. Планирование в пространстве поведений (Behavior-based planning)<br>4. Планирование с повторным использованием планов (Case-Based Planning)<br>Правильный ответ: 3<br>Пояснение:<br>● Планирование в пространстве поведений является правильным ответом, так как этот реактивный подход фокусируется на выборе следующего действия на основе текущего состояния, а не на составлении долгосрочного жесткого плана. Это позволяет быстро адаптироваться к изменениям.<br>● Детальное планирование на много шагов вперед является неправильным ответом, так как в динамической среде такой план быстро устаревает, и его постоянное пересчитывание требует больших вычислительных затрат.<br>● Иерархическое планирование является неправильным ответом. Хотя оно уменьшает сложность, оно все равно создает достаточно жесткий план, который может быть нарушен непредвиденными событиями.<br>● Планирование с повторным использованием планов является неправильным ответом. Оно эффективно в стабильных средах, где прошлый опыт релевантен, но в быстро меняющейся среде |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | старые планы могут быть бесполезны.<br><br>**Задание 4.2**<br>Вопрос: Какая из перечисленных российских СУБД, основанная на открытом коде и входящая в реестр Минцифры, предпочтительна для использования в качестве надежного хранилища данных для систем планирования в госсекторе и критической инфраструктуре?<br>1. MySQL<br>2. Postgres Pro<br>3. SQLite<br>4. Oracle Database<br>Правильный ответ: 2<br>Пояснение:<br>● Postgres Pro является правильным ответом, так как это российская коммерческая СУБД, созданная на основе PostgreSQL. Она обладает улучшенной производительностью, безопасностью и поддерживается российскими разработчиками, что соответствует требованиям импортозамещения.<br>● MySQL является неправильным ответом, так как изначально разрабатывался шведской компанией, а сейчас принадлежит американской Oracle Corporation.<br>● SQLite является неправильным ответом, так как это встраиваемая библиотека, а не полноценная серверная СУБД для промышленного применения.<br>● Oracle Database является неправильным ответом, так как это проприетарный продукт американской корпорации Oracle<br>**Задание 5.1**<br>Вопрос: Как называется свойство реальной среды, которое описывает, могут ли действия агента изменять саму среду и каким образом? Это свойство требует от системы планирования постоянно обновлять свою внутреннюю модель мира. Ответ состоит из одного слова.<br>Правильный ответ: Динамичность<br>Пояснение: Динамичность — это ключевое свойство реальных сред, отличающее их от статических моделей. Оно означает, что состояние мира меняется со временем независимо от действий агента (например, |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | погодные условия, движение других людей). Это напрямую влияет на архитектуру системы, требуя цикла "наблюдение-планирование-действие" и механизмов перепланирования.<br><br>**Задание 5.2**<br>Вопрос: В архитектуре автономных систем, работающих в реальном мире, как называется программный компонент, который напрямую взаимодействует с физическими датчиками, преобразуя сырые сигналы в структурированную информацию для модуля планирования? Ответ состоит из двух слов.<br>Правильный ответ: Модуль восприятия<br>Пояснение: Модуль восприятия (Perception Module) — это критически важный компонент, который служит "органами чувств" системы. Он отвечает за фильтрацию шума, распознавание объектов, построение карты и в конечном итоге — за создание непротиворечивой и актуальной модели мира, на основе которой уже работает планировщик. Без него планировщик будет работать с устаревшими или неверными данными. |
| **Тема 2.7** Рассуждения в условиях неопределенности | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 1.1**<br>Вопрос: Какие из перечисленных подходов и математических моделей используются для принятия решений и планирования в условиях неопределенности?<br>**1.** Детерминированные графы состояний с однозначными переходами<br>**2.** Марковские процессы принятия решений (Markov Decision Processes, MDP)<br>**3.** Байесовские сети доверия (Bayesian Belief Networks)<br>**4.** Таблицы случайных чисел для выбора действий<br>Правильные ответы: 2, 3<br>Пояснение:<br>● Марковские процессы принятия решений (MDP) являются правильным ответом, так как это классическая модель для последовательного принятия решений в стохастической среде, где последствия действий вероятностны. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | ● Байесовские сети доверия являются правильным ответом, так как они позволяют представлять и делать выводы о ненаблюдаемых переменных и причинно-следственных связях в условиях неполной информации, обновляя вероятности по мере поступления новых данных.<br>● Детерминированные графы состояний являются неправильным ответом, так как они по определению не учитывают неопределенность; переходы между состояниями в них строго детерминированы.<br>● Таблицы случайных чисел являются неправильным ответом, так как это примитивный генератор случайности, а не интеллектуальный метод для работы с неопределенностью; он не использует модель среды для принятия обоснованных решений.<br><br>**Задание 1.2**<br>Вопрос: Для реализации системы, способной функционировать в условиях неопределенности (например, для робототехники или анализа рисков), требуются следующие технологические компоненты и языки программирования:<br>**1.** Язык R со специализированными пакетами для статистического анализа и байесовских методов.<br>**2.** Язык Python с библиотеками для машинного обучения (Scikit-learn, Pyro, TensorFlow Probability).<br>**3.** Детерминированная экспертная система без механизма обновления вероятностей.<br>**4.** Система управления базами данных для хранения истории наблюдений и параметров вероятностных моделей.<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● Язык R является правильным ответом, так как это мощный инструмент для статистических вычислений, включая байесовский вывод, что критически важно для работы с вероятностными моделями.<br>● Язык Python является правильным ответом, так как его |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | экосистема включает современные библиотеки для вероятностного программирования (Pyro) и машинного обучения, которые позволяют строить и обучать сложные модели для uncertain environments.<br>● Система управления базами данных является правильным ответом, так как для обучения и верификации вероятностных моделей необходимы большие объемы данных, которые надежно хранятся в СУБД.<br>● Детерминированная экспертная система является неправильным ответом, так как в чистом виде, без механизма работы с вероятностями и нечеткой логикой, она не приспособлена для работы в условиях неопределенности.<br>**Задание 2.1**<br>Вопрос: Установите последовательность этапов в цикле байесовского обновления убеждений (belief update) для агента, действующего в условиях неопределенности.<br>1. Действие: Выполнить действие, влияющее на среду.<br>2. Наблюдение: Получить новые данные (evidence) от сенсоров.<br>3. Прогноз (Prediction): Предсказать априорное распределение вероятностей над состояниями мира на основе предыдущих убеждений и выполненного действия.<br>4. Коррекция (Update): Обновить убеждения (апостериорное распределение), используя теорему Байеса на основе априорного распределения и новых данных.<br>5. Принятие решения: Выбрать оптимальное действие на основе текущих убеждений и функции полезности.<br>Правильный порядок: 5, 1, 3, 2, 4<br>Пояснение: Цикл начинается с принятия решения на основе текущих знаний (5). Затем действие выполняется (1). Агент предсказывает, как действие изменит состояние мира (3). После этого он получает реальные данные от сенсоров (2) и корректирует свои убеждения, совмещая прогноз и наблюдение (4). Обновленные убеждения используются для следующего цикла принятия решения.<br>**Задание 2.2** |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Вопрос: Расположите в логическом порядке этапы построения и использования системы, основанной на байесовской сети доверия, для диагностики неисправностей в сложном оборудовании.<br>1. Кодирование экспертных знаний и статистических данных в виде условных вероятностей для узлов сети.<br>2. Ввод данных о наблюдаемых симптомах (evidence) в сеть.<br>3. Проведение вероятностного вывода для вычисления апостериорных вероятностей неисправностей.<br>4. Построение графовой структуры сети, определяющей причинно-следственные связи между переменными.<br>5. Принятие решения о ремонтных действиях на основе полученных вероятностей.<br>Правильный порядок: 4, 1, 2, 3, 5<br>Пояснение: Сначала создается структурная модель причинно-следственных связей (4). Затем модель наполняется количественными параметрами — вероятностями (1). При эксплуатации в сеть вводятся данные о конкретном случае (2). Запускается алгоритм вывода, который рассчитывает вероятности ненаблюдаемых неисправностей (3). На основе этого расчета принимается информированное решение (5)<br>**Задание 3.1**<br>Установите соответствие между концепцией работы в условиях неопределенности и ее описанием.<br><br>Список А (Концепция):<br>1. Априорная вероятность (Prior)<br>2. Апостериорная вероятность (Posterior)<br>3. Правдоподобие (Likelihood)<br>4. Теорема Байеса (Bayes' Theorem)<br><br>Список Б (Описание):<br>A. Вероятность гипотезы после учета новых наблюдаемых данных.<br>B. Вероятность наблюдаемых данных при условии, что гипотеза верна.<br>C. Исходная степень уверенности в гипотезе до учета новых данных.<br>D. Математическая формула для обновления убеждений: |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | $\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$.<br><br>Ответ: 1C, 2A, 3B, 4D<br><br>Пояснение: Prior — это начальное убеждение (1-C). Posterior — это конечное, обновленное убеждение (2-A). Likelihood измеряет адекватность гипотезы имеющимся данным (3-B). Теорема Байеса формализует процесс перехода от Prior к Posterior (4-D).<br><br>**Задание 3.2**<br>Установите соответствие между типом неопределенности в задаче управления и рекомендуемым технологическим подходом для ее решения.<br><br>Список А (Тип неопределенности):<br>1. Стохастичность (случайность outcomes действий)<br>2. Неполная наблюдаемость состояния<br>3. Неопределенность модели среды<br>4. Нечеткость, размытость понятий<br><br>Список Б (Технологический подход):<br>A. Обучение с подкреплением (Reinforcement Learning)<br>B. Марковские процессы принятия решений (MDP)<br>C. Частично наблюдаемые марковские процессы (POMDP)<br>D. Нечеткая логика (Fuzzy Logic) и нечеткие системы вывода<br><br>Ответ: 1B, 2C, 3A, 4D<br><br>Пояснение: MDP работают с вероятностными последствиями действий (1-B). POMDP расширяют MDP, вводя понятие "убеждения" о состоянии, так как состояние не наблюдается напрямую (2-C). Обучение с подкреплением позволяет агенту обучать оптимальную стратегию, даже не зная точной модели среды (3-A). Нечеткая логика оперирует понятиями вроде "высокий" или "низкий", а не точными числами (4-D). |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | **Задание 4.1**<br>Вопрос: Какая модель обобщает марковские процессы принятия решений (MDP) на случай, когда агент не может напрямую наблюдать истинное состояние среды, а вынужден поддерживать вероятностное убеждение (belief state) о нем?<br>1. Скрытая марковская модель (Hidden Markov Model, HMM)<br>2. Частично наблюдаемый марковский процесс принятия решений (Partially Observable MDP, POMDP)<br>3. Детерминированный граф состояний<br>4. Байесовская сеть статического вывода<br>Правильный ответ: 2<br>Пояснение:<br>● Частично наблюдаемый марковский процесс принятия решений (POMDP) является правильным ответом, так как это прямая и наиболее общая модель для последовательного принятия решений в условиях как стохастичности, так и неполной наблюдаемости. Belief state — это ключевое понятие в POMDP.<br>● Скрытая марковская модель (HMM) является неправильным ответом, так как это модель для *прогнозирования* и *анализа* последовательностей, но не для *активного принятия решений*.<br>● Детерминированный граф состояний является неправильным ответом, так как он не учитывает ни стохастичность, ни неполную наблюдаемость.<br>● Байесовская сеть статического вывода является неправильным ответом, так как она обычно используется для разовых выводов, а не для последовательного принятия решений во времени.<br><br>**Задание 4.2**<br>Вопрос: Какая из перечисленных российских систем управления базами данных, являющаяся развитием PostgreSQL, может эффективно использоваться для хранения и обработки больших объемов статистических данных, необходимых для обучения и работы вероятностных моделей в условиях неопределенности?<br>1. SQLite<br>2. MySQL |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | 3. Postgres Pro<br>4. Oracle Database<br>Правильный ответ: 3<br>Пояснение:<br>● Postgres Pro является правильным ответом, так как это российская СУБД, основанная на мощной и надежной PostgreSQL, которая поддерживает сложные типы данных, оконные функции и расширения для статистического анализа, что делает ее подходящей для работы с данными в условиях неопределенности.<br>● SQLite является неправильным ответом, так как это встраиваемая БД, не предназначенная для анализа больших данных.<br>● MySQL и Oracle Database являются неправильными ответами, так как это продукты американских компаний (Oracle Corporation), а не отечественные разработки.<br>**Задание 5.1**<br>Вопрос: Как называется вероятностная модель, представляющая набор случайных переменных и их условных зависимостей via направленный ациклический граф (DAG)? Она широко используется для рассуждений в условиях неопределенности. Ответ состоит из двух слов.<br>Правильный ответ: Байесовская сеть (Bayesian Network)<br>Пояснение: Байесовская сеть (Bayesian Network) — это компактное представление совместного распределения вероятностей над множеством переменных. Она позволяет эффективно вычислять апостериорные вероятности при наличии свидетельств, что является основой для диагностических и прогностических систем в условиях неполной информации.<br><br>**Задание 5.2**<br>Вопрос: В контексте частично наблюдаемых марковских процессов принятия решений (POMDP), как называется функция, которая отображает текущее вероятностное убеждение агента о состоянии среды (belief state) в выбираемое действие? Ответ состоит из двух слов.<br>Правильный ответ: Политика (Policy) |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Пояснение: В POMDP, где нет доступа к истинному состоянию, политика (Policy) определяется как отображение из пространства убеждений (belief space) в пространство действий. Нахождение оптимальной политики является конечной целью решения POMDP; она говорит агенту, что делать при любой возможной степени его неуверенности в состоянии мира<br>. |
| **Тема 3.1** Стандартизация в области интеллектуального анализа данных | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1 Знать:** Методы и средства проектирования программных интерфейсов | **Задание 1.1**<br>Вопрос: Какие из перечисленных стандартов и спецификаций имеют непосредственное отношение к процессу интеллектуального анализа данных (Data Mining) и организации этого процесса в рамках крупных проектов?<br>**1.** Спецификация CRISP-DM (Cross-Industry Standard Process for Data Mining)<br>**2.** Стандарт ISO/IEC 25012 (качество данных)<br>**3.** Стандарт IEEE 802.11 (Wi-Fi)<br>**4.** Язык разметки HTML5<br>Правильные ответы: 1, 2<br>Пояснение:<br>● CRISP-DM является правильным ответом, так как это наиболее распространенная и признанная в отрасли методология (де-факто стандарт), описывающая жизненный цикл проекта Data Mining.<br>● ISO/IEC 25012 является правильным ответом, так как этот международный стандарт определяет модель качества данных. Качество данных — критически важный аспект для успешного интеллектуального анализа.<br>● IEEE 802.11 является неправильным ответом, так как это стандарт для беспроводных локальных сетей (Wi-Fi), не имеющий отношения к процессам анализа данных.<br>● HTML5 является неправильным ответом, так как это язык разметки для веб-страниц, а не стандарт для организации процессов анализа данных. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | **Задание 1.2**<br>Вопрос: Для реализации промышленного проекта по интеллектуальному анализу данных в соответствии с современными стандартами и использованием отечественного ПО могут потребоваться следующие компоненты:<br>**1.** Язык Python с библиотеками (Pandas, Scikit-learn) для реализации этапов ETL и построения моделей.<br>**2.** Российская СУБД (например, Postgres Pro) для хранения и первичной обработки данных.<br>**3.** Графический редактор для визуализации итоговых отчетов.<br>**4.** Платформа управления контейнерами (например, Docker) для стандартизации окружения и воспроизводимости экспериментов.<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● Python с библиотеками является правильным ответом, так как это современный стандарт для прототипирования и реализации алгоритмов машинного обучения и анализа данных.<br>● Российская СУБД Postgres Pro является правильным ответом, так как она соответствует требованиям импортозамещения и обладает всеми необходимыми функциями для работы с данными, включая расширения для машинного обучения.<br>● Платформа управления контейнерами является правильным ответом, так как она обеспечивает воспроизводимость результатов и стандартизацию рабочего окружения, что является важным аспектом промышленного внедрения моделей.<br>● Графический редактор является неправильным ответом, так как это ручной инструмент для дизайна, а не часть стандартизированного технологического стека для автоматизированного анализа данных.<br>**Задание 2.1**<br>Вопрос: Установите правильную последовательность фаз согласно методологии CRISP-DM.<br>1. Моделирование (Modeling): Выбор и применение различных |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | алгоритмов моделирования и калибровка их параметров.<br>2. Понимание данных (Data Understanding): Сбор первичных данных и их первоначальное исследование.<br>3. Подготовка данных (Data Preparation): Формирование конечного набора данных для моделирования (очистка, преобразование, агрегация).<br>4. Постановка бизнес-задачи (Business Understanding): Определение целей и требований проекта с точки зрения бизнеса.<br>5. Внедрение (Deployment): Развертывание модели в производственной среде.<br>6. Оценка (Evaluation): Проверка того, что модель достигла целей проекта, и утверждение следующих шагов.<br>Правильный порядок: 4, 2, 3, 1, 6, 5<br>Пояснение: CRISP-DM начинается с понимания бизнес-целей (4). Затем следует сбор и изучение данных (2). После этого данные тщательно подготавливаются (3). Только на следующем этапе выбираются и настраиваются модели (1). Построенные модели всесторонне оцениваются (6), и если они успешны, внедряются в эксплуатацию (5).<br><br>**Задание 2.2**<br>Вопрос: Расположите в логическом порядке этапы развертывания отечественной платформы для анализа данных в защищенном контуре.<br>1. Разработка ETL-процедур и скриптов анализа на Python.<br>2. Установка и настройка российской ОС (например, Astra Linux) на сервере.<br>3. Наполнение базы тестовыми данными и проведение приемочных испытаний.<br>4. Установка и настройка российской СУБД (например, Postgres Pro).<br>5. Проектирование архитектуры решения и схемы базы данных.<br>Правильный порядок: 5, 2, 4, 1, 3<br>Пояснение: Сначала проектируется архитектура (5). Затем подготавливается инфраструктура — устанавливается ОС (2). После этого развертывается система хранения и обработки данных — СУБД |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | (4). Далее реализуется бизнес-логика анализа (1). Финальный этап — тестирование всего комплекса (3).<br><br>**Задание 3.1**<br>Установите соответствие между этапом методологии CRISP-DM и его основной задачей.<br><br>Список А (Этап CRISP-DM):<br>1. Business Understanding<br>2. Data Preparation<br>3. Modeling<br>4. Evaluation<br><br>Список Б (Основная задача):<br>A. Преобразование сырых данных в пригодный для моделирования формат (очистка, обогащение).<br>B. Интерпретация результатов моделирования с точки зрения бизнес-целей, утверждение перехода к внедрению.<br>C. Выявление в данных содержательных закономерностей с помощью алгоритмов машинного обучения.<br>D. Определение того, какие бизнес-проблемы должны быть решены с помощью анализа данных.<br><br>Ответ: 1D, 2A, 3C, 4B<br><br>Пояснение: Первый этап — это перевод бизнес-задачи на язык данных (1-D). Самый трудоемкий этап — подготовка данных (2-A). Моделирование — это ядро анализа (3-C). Оценка проверяет, решена ли исходная бизнес-задача (4-B).<br><br>**Задание 3.2**<br>Установите соответствие между компонентом технологического стека для анализа данных и его российской альтернативой. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | Список А (Компонент стека):<br>1. Операционная система<br>2. Система управления базами данных<br>3. Язык программирования для анализа<br>4. Платформа для управления контейнерами<br><br>Список Б (Российская альтернатива):<br>A. Postgres Pro<br>B. Astra Linux<br>C. Python (как мировой стандарт, активно используемый в РФ)<br>D. Russian Docker Registry / Tarantool Kubernetes<br><br>Ответ: 1B, 2A, 3C, 4D<br><br>Пояснение: Astra Linux — ведущая отечественная серверная ОС (1-B). Postgres Pro — российская СУБД для хранения данных (2-A). Python является де-факто стандартом в Data Science и широко используется в России (3-C). В России развиваются собственные экосистемы на базе Kubernetes для управления контейнерами (4-D).**Задание 4.1**<br>Вопрос: Какая из перечисленных методологий является наиболее распространенным отраслевым стандартом (de facto standard), описывающим жизненный цикл проекта по интеллектуальному анализу данных и состоящим из шести основных фаз?<br>1. Agile SCRUM<br>2. CRISP-DM (Cross-Industry Standard Process for Data Mining)<br>3. ITIL (Information Technology Infrastructure Library)<br>4. Waterfall (Каскадная модель)<br>Правильный ответ: 2<br>Пояснение:<br>● CRISP-DM является правильным ответом, так как это именно та методология, которая была разработана консорциумом компаний специально для проектов Data Mining и стала отраслевым стандартом благодаря своей универсальности и полноте. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | • Agile SCRUM является неправильным ответом, так как это гибкая методология управления проектами общего назначения, а не специализированный стандарт для анализа данных.<br>• ITIL является неправильным ответом, так как это библиотека лучших практик для управления ИТ-услугами, а не для реализации проектов по анализу данных.<br>• Waterfall является неправильным ответом, так как это общая модель жизненного цикла ПО, не адаптированная под специфические итеративные процессы анализа данных.<br><br>**Задание 4.2**<br>Вопрос: Какая из перечисленных российских операционных систем, соответствующая требованиям безопасности и входящая в реестр Минцифры, предпочтительна для развертывания серверной части аналитической платформы, работающей с данными ограниченного доступа?<br>1. Windows Server 2022<br>2. Ubuntu Server 22.04 LTS<br>3. Astra Linux<br>4. Red Hat Enterprise Linux<br>Правильный ответ: 3<br>Пояснение:<br>• Astra Linux является правильным ответом, так как это отечественная ОС, специально разработанная для создания защищенных информационных систем, работающих с конфиденциальной информацией и гостайной. Она имеет все необходимые сертификаты ФСТЭК России.<br>• Windows Server 2022 является неправильным ответом, так как это продукт американской компании Microsoft.<br>• Ubuntu Server и Red Hat Enterprise Linux являются неправильными ответами, так как это продукты международных (фактически американских) компаний Canonical и Red Hat (IBM) соответственно.<br>• . |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Тест по теме |
|---|---|---|---|
| | | | • азработки<br>**Задание 5.1**<br>Вопрос: Как называется наиболее известная иерархическая модель стандартов, связанных с качеством данных, которая включает такие характеристики, как точность, полнота, актуальность и согласованность? Ответ должен состоять из аббревиатуры и номера стандарта (например, ISO XXXX).<br>Правильный ответ: ISO 25012<br>Пояснение: ISO/IEC 25012 — это международный стандарт, который определяет модель качества данных. Он описывает характеристики качества данных, которые могут быть использованы для оценки пригодности данных для анализа, что является критически важным первым шагом в любом проекте интеллектуального анализа данных.<br><br>**Задание 5.2**<br>Вопрос: Какой открытый язык программирования с богатой экосистемой библиотек (таких как Pandas, Scikit-learn, TensorFlow) является де-факто отраслевым стандартом для реализации задач интеллектуального анализа данных и машинного обучения? Ответ состоит из одного слова.<br>Правильный ответ: Python<br>Пояснение: Python является доминирующим языком в области Data Science и Machine Learning благодаря своему простому синтаксису, активному сообществу и огромному количеству специализированных библиотек для обработки данных, визуализации и построения моделей. Его использование стало стандартом для быстрого прототипирования и промышленной реализации аналитических решений |

***Критерии оценивания тестового задания:***

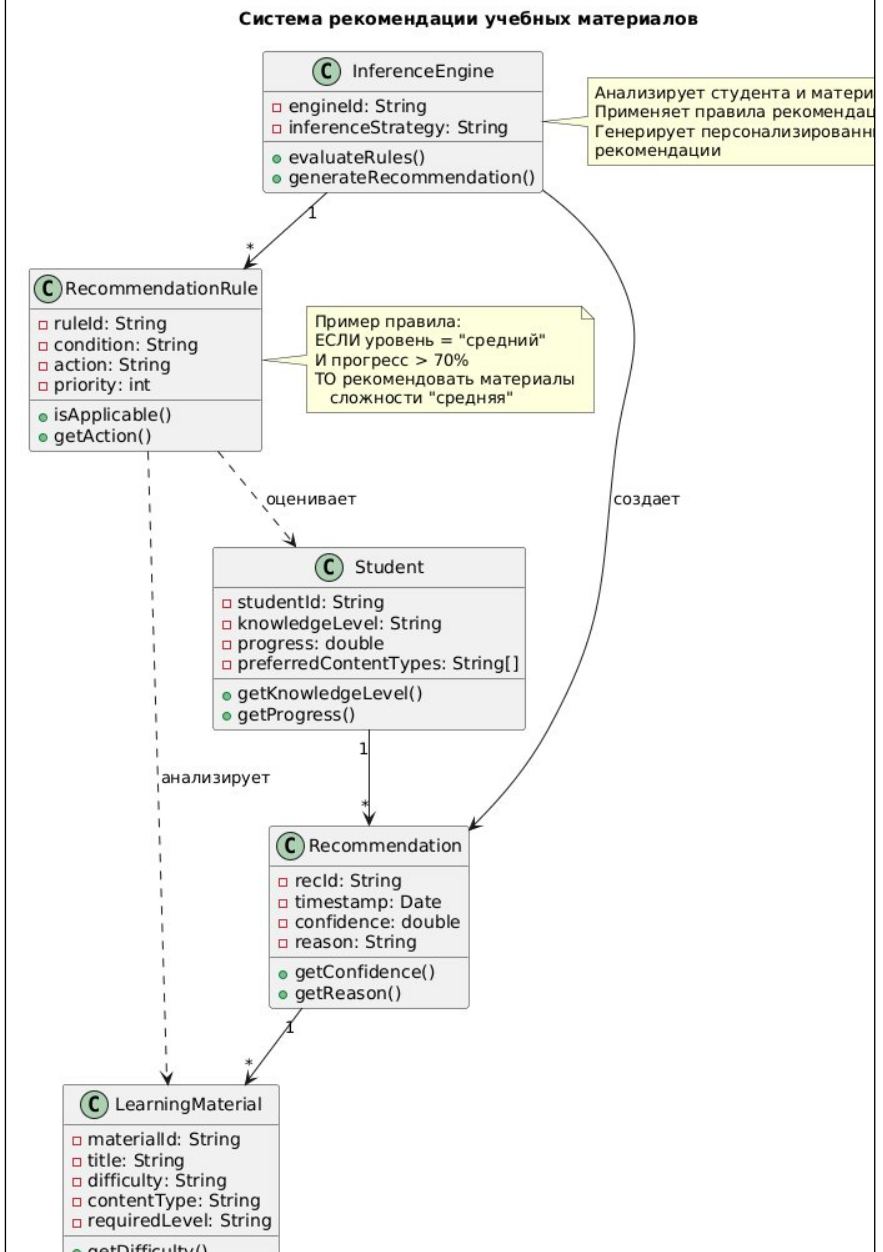
| Оценка | Критерии оценивания |
|---|---|
| **отлично** | от 90 до 100 % правильно выполненных заданий |

| | |
|---|---|
| **хорошо** | от 70 до 89 % правильно выполненных заданий |
| **удовлетворительно** | от 50 до 69 % правильно выполненных заданий |
| **неудовлетворительно** | менее 50 % правильно выполненных заданий |

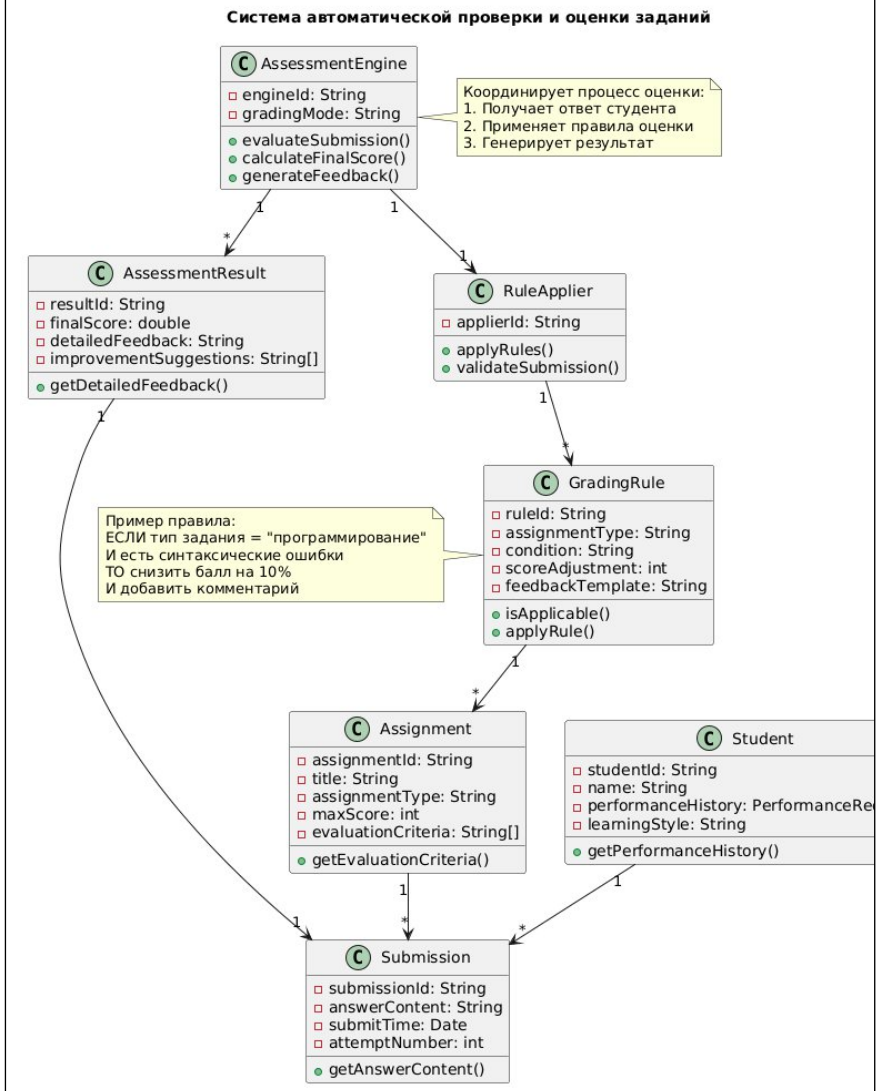
**4.2 ТИПОВЫЕ ПРАКТИЧЕСКИЕ ЗАДАНИЯ**

Практические задания должны отражать умение обучающегося применять осваеваемую компетенцию в практических ситуациях и при решении производственных задач.

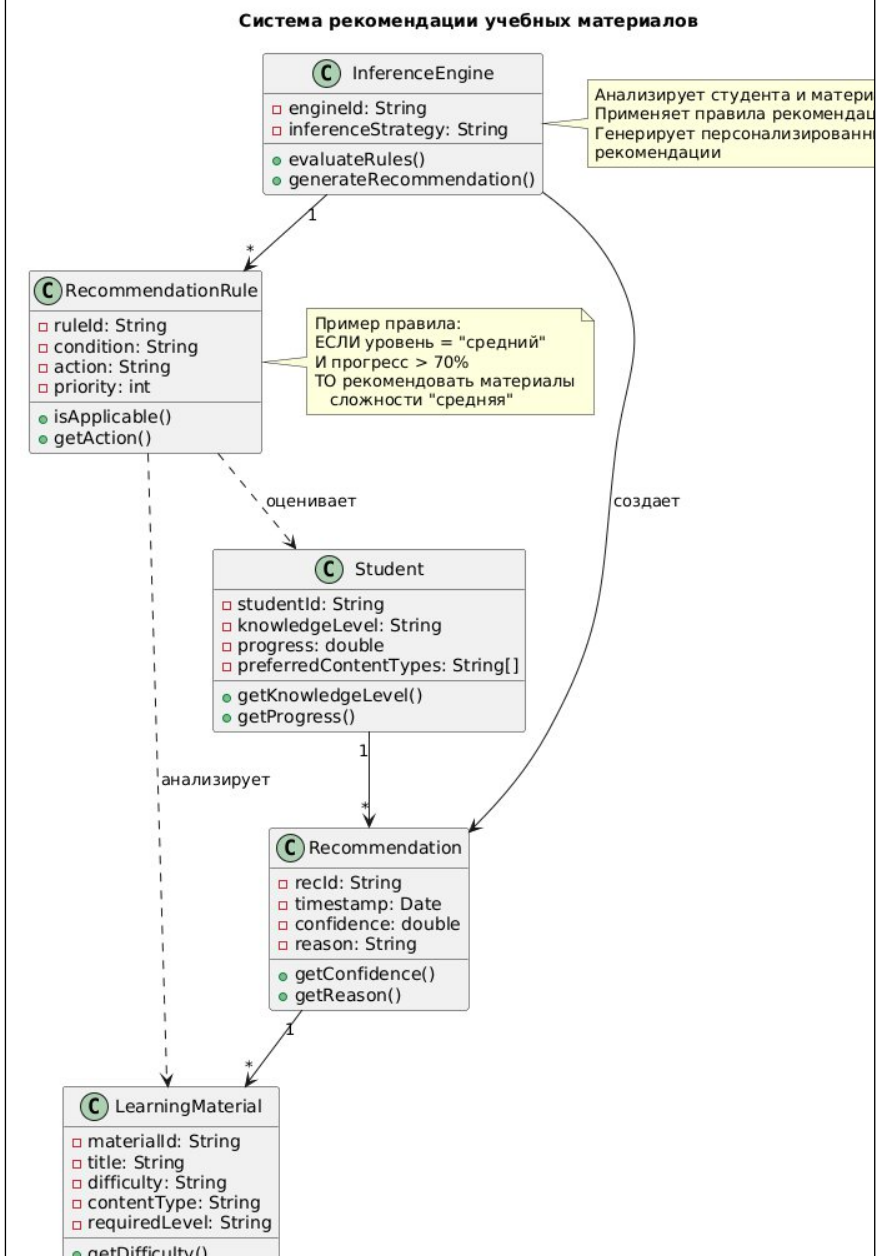
| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| **Раздел №1 «Интеллектуальные информационные системы»**<br>Тема 1.1 Искусственный интеллект: предпосылки, развитие, современное состояние разработок<br>Тема 1.2 Данные и знания<br>Тема 1.3 Моделирование знаний о предметных областях как основа интеллектуальных систем<br>Тема 1.4 Основные компоненты интеллектуальной информационной системы | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.2 Уметь:** Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | **Задание №1**<br>Условие: Разработайте диаграмму классов в нотации UML для машины логического вывода интеллектуальной системы рекомендации учебных материалов. Система должна анализировать:<br>● Уровень знаний студента (начальный, средний, продвинутый)<br>● Прогресс обучения (процент выполненных заданий)<br>● Предпочтения по типам контента (видео, текст, интерактивные задания)<br>● Сложность материалов (легкая, средняя, сложная)<br>Требования к машине логического вывода:<br>● Хранить правила рекомендаций<br>● Оценивать соответствие материалов студенту<br>● Формировать персонализированные рекомендации<br>● Вести историю рекомендаций<br>Пример ответа: |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | |  |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | **Задание №2**<br>Условие: Разработайте диаграмму классов в нотации UML для машины логического вывода интеллектуальной системы автоматической проверки и оценки студенческих заданий.<br>Система должна:<br>● Анализировать ответы студентов по различным критериям<br>● Применять правила оценки в зависимости от типа задания<br>● Учитывать историю успеваемости студента<br>● Формировать обратную связь и рекомендации по улучшению<br><br>Требования к машине логического вывода:<br>● Хранить критерии оценки для разных типов заданий<br>● Применять правила проверки<br>● Адаптировать оценку на основе прогресса студента<br>● Генерировать персонализированные комментарии<br>Пример ответа |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | |  |

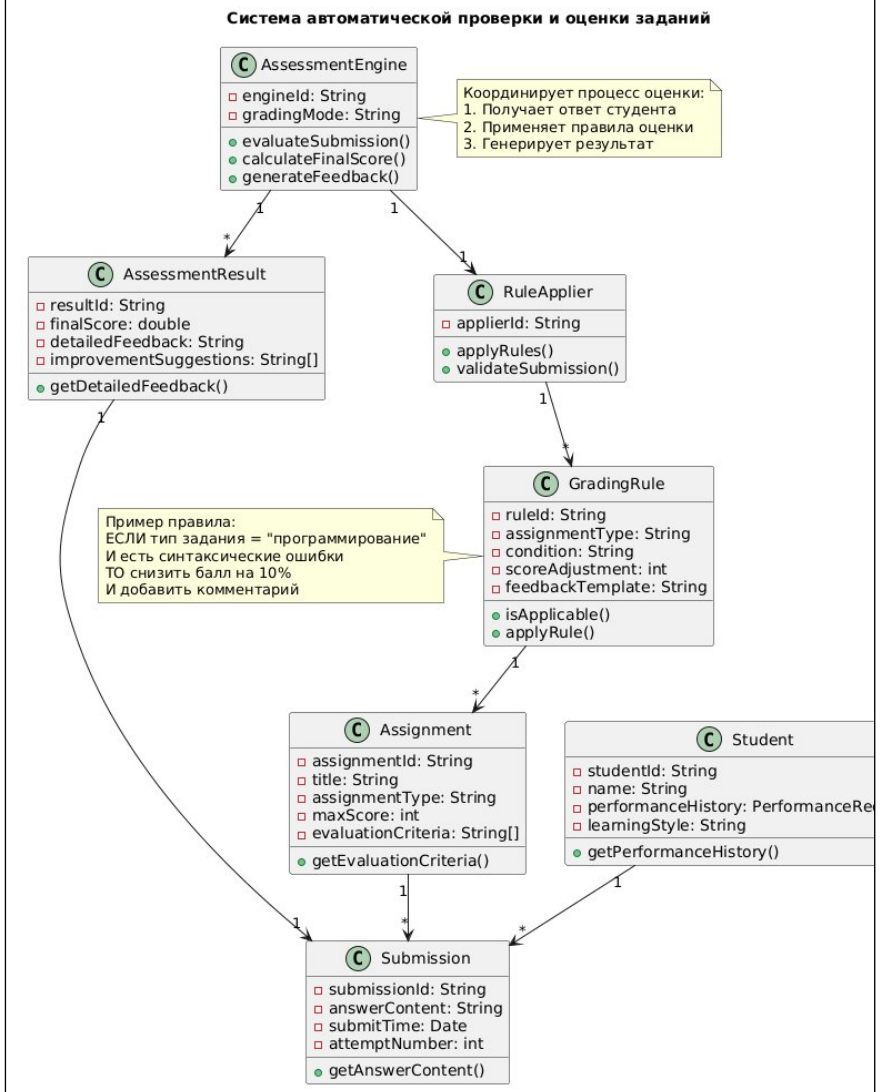
| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| **Раздел №2 «Решение проблем»**<br>Тема 2.1 Структуры и стратегии поиска в пространстве состояний<br>Тема 2.2 Эвристический поиск<br>Тема 2.3 Управление поиском и его реализация в пространстве состояний<br>Тема 2.4 Задача планирования<br>Тема 2.5 Планирование и осуществление действий в реальном мире<br>Тема 2.6 Интеллектуализация процедур поиска и управления<br>Тема 2.7 Рассуждения в условиях неопределенности | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.2 Уметь:** Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | **Задание №3**<br>Реализовать на выбранном языке программирования компоненты интеллектуальной системы рекомендации учебных материалов согласно предоставленной UML-диаграмме. Система должна автоматически подбирать учебные материалы для студентов на основе их уровня знаний, прогресса обучения и предпочтений по типам контента.<br>Требования к реализации:<br>● Создать все классы согласно UML-диаграмме<br>● Реализовать логику работы машины логического вывода<br>● Обеспечить взаимодействие между компонентами системы<br>● Реализовать систему правил для персонализированных рекомендаций<br>UML-диаграмма: |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | |  |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | Ответ (на языке python):<br>import datetime<br>class Student:<br>def __init__(self, studentId, knowledgeLevel, progress, preferredContentTypes):<br>self.studentId = studentId<br>self.knowledgeLevel = knowledgeLevel<br>self.progress = progress<br>self.preferredContentTypes = preferredContentTypes<br>def getKnowledgeLevel(self):<br>return self.knowledgeLevel<br>def getProgress(self):<br>return self.progress<br>def __str__(self):<br>return f"Student {self.studentId}: {self.knowledgeLevel} level, {self.progress}% progress"<br>class LearningMaterial:<br>def __init__(self, materialId, title, difficulty, contentType, requiredLevel):<br>self.materialId = materialId<br>self.title = title<br>self.difficulty = difficulty<br>self.contentType = contentType<br>self.requiredLevel = requiredLevel<br>def getDifficulty(self):<br>return self.difficulty<br>def isSuitableFor(self, studentLevel):<br>levels = ["beginner", "intermediate", "advanced"]<br>return levels.index(studentLevel) >= levels.index(self.requiredLevel)<br>def __str__(self):<br>return f"Material {self.materialId}: {self.title} ({self.difficulty})"<br>class RecommendationRule:<br>def __init__(self, ruleId, condition, action, priority):<br>self.ruleId = ruleId<br>self.condition = condition<br>self.action = action<br>self.priority = priority<br>def isApplicable(self, student, material):<br>try:<br>conditions = {<br>"student.knowledgeLevel == material.requiredLevel": |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | student.knowledgeLevel == material.requiredLevel,<br>"student.progress > 70 and material.difficulty == 'intermediate'":<br>student.progress > 70 and material.difficulty == "intermediate",<br>"material.contentType in student.preferredContentTypes":<br>material.contentType in student.preferredContentTypes<br>}<br>return conditions.get(self.condition, False)<br>except:<br>return False<br>def getAction(self):<br>return self.action<br>def __str__(self):<br>return f"Rule {self.ruleId}: {self.condition}"<br>class Recommendation:<br>def __init__(self, recId, confidence, reason):<br>self.recId = recId<br>self.timestamp = datetime.datetime.now()<br>self.confidence = confidence<br>self.reason = reason<br>self.materials = []<br>def getConfidence(self):<br>return self.confidence<br>def getReason(self):<br>return self.reason<br>def addMaterial(self, material):<br>self.materials.append(material)<br>def __str__(self):<br>materials_str = ", ".join([str(m) for m in self.materials])<br>return f"Recommendation {self.recId}: {self.confidence:.1%} confidence - {self.reason}\nMaterials: {materials_str}"<br>class InferenceEngine:<br>def __init__(self, engineId, inferenceStrategy="rule_based"):<br>self.engineId = engineId<br>self.inferenceStrategy = inferenceStrategy<br>self.rules = []<br>def addRule(self, rule):<br>self.rules.append(rule)<br>self.rules.sort(key=lambda x: x.priority)<br>def evaluateRules(self, student, material):<br>applicable_rules = []<br>for rule in self.rules: |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | if rule.isApplicable(student, material):<br>applicable_rules.append(rule)<br>return applicable_rules<br>def generateRecommendation(self, student, materials):<br>recommendations = []<br>for material in materials:<br>applicable_rules = self.evaluateRules(student, material)<br>if applicable_rules:<br>confidence = len(applicable_rules) / len(self.rules)<br>reason = f"Applied rules: {', '.join([r.ruleId for r in applicable_rules])}"<br>recommendation = Recommendation(<br>f"REC_{student.studentId}_{material.materialId}",<br>confidence,<br>reason<br>)<br>recommendation.addMaterial(material)<br>recommendations.append(recommendation)<br>recommendations.sort(key=lambda x: x.confidence, reverse=True)<br>return recommendations<br>def __str__(self):<br>return f"InferenceEngine {self.engineId} with {len(self.rules)} rules"<br><br>**Задание №4**<br>Реализовать на языке Python компоненты интеллектуальной системы автоматической проверки и оценки студенческих заданий согласно предоставленной UML-диаграмме. Система должна автоматически оценивать ответы студентов, применять правила оценки и генерировать персонализированную обратную связь.<br>Требования к реализации:<br>● Создать все классы согласно UML-диаграмме<br>● Реализовать логику автоматической оценки заданий<br>● Обеспечить применение правил оценки в зависимости от типа задания<br>● Реализовать генерацию детализированной обратной связи<br>● Учесть историю успеваемости студентов при оценке<br>UML-диаграмма: |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | |  |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | Ответ:<br>import datetime<br>class Student:<br>def __init__(self, studentId, name, performanceHistory, learningStyle):<br>self.studentId = studentId<br>self.name = name<br>self.performanceHistory = performanceHistory<br>self.learningStyle = learningStyle<br>def getPerformanceHistory(self):<br>return self.performanceHistory<br>def getAverageScore(self):<br>return sum(self.performanceHistory) / len(self.performanceHistory) if self.performanceHistory else 0<br>def __str__(self):<br>return f"Student {self.studentId}: {self.name} (avg: {self.getAverageScore():.1f})"<br>class Assignment:<br>def __init__(self, assignmentId, title, assignmentType, maxScore, evaluationCriteria):<br>self.assignmentId = assignmentId<br>self.title = title<br>self.assignmentType = assignmentType<br>self.maxScore = maxScore<br>self.evaluationCriteria = evaluationCriteria<br>def getEvaluationCriteria(self):<br>return self.evaluationCriteria<br>def __str__(self):<br>return f"Assignment {self.assignmentId}: {self.title} ({self.assignmentType})"<br>class Submission:<br>def __init__(self, submissionId, student, assignment, answerContent, attemptNumber):<br>self.submissionId = submissionId<br>self.student = student<br>self.assignment = assignment<br>self.answerContent = answerContent<br>self.submitTime = datetime.datetime.now()<br>self.attemptNumber = attemptNumber<br>def getAnswerContent(self):<br>return self.answerContent<br>def __str__(self):<br>return f"Submission {self.submissionId} by {self.student.name} for |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | {self.assignment.title}"<br>class GradingRule:<br>  def __init__(self, ruleId, assignmentType, condition, scoreAdjustment, feedbackTemplate):<br>  self.ruleId = ruleId<br>  self.assignmentType = assignmentType<br>  self.condition = condition<br>  self.scoreAdjustment = scoreAdjustment<br>  self.feedbackTemplate = feedbackTemplate<br>  def isApplicable(self, submission, student):<br>  if self.assignmentType != "all" and self.assignmentType != submission.assignment.assignmentType:<br>  return False<br>  if self.condition == "missing_return":<br>  return "return" not in submission.answerContent and "def " in submission.answerContent<br>  elif self.condition == "good_naming":<br>  good_names = ["calculate", "process", "handle", "validate"]<br>  return any(name in submission.answerContent for name in good_names)<br>  elif self.condition == "syntax_error":<br>  syntax_errors = ["print(", "input("]<br>  return any(error in submission.answerContent for error in syntax_errors)<br>  elif self.condition == "high_performer":<br>  return student.getAverageScore() > 80<br>  return False<br>  def applyRule(self):<br>  return {<br>  "scoreAdjustment": self.scoreAdjustment,<br>  "feedback": self.feedbackTemplate<br>  }<br>  def __str__(self):<br>  return f"Rule {self.ruleId}: {self.condition} ({self.scoreAdjustment} points)"<br>class RuleApplier:<br>  def __init__(self, applierId):<br>  self.applierId = applierId<br>  self.rules = []<br>  def addRule(self, rule):<br>  self.rules.append(rule)<br>  def applyRules(self, submission, student):<br>  applied_rules = []<br>  total_adjustment = 0 |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | feedback_items = []<br>for rule in self.rules:<br>if rule.isApplicable(submission, student):<br>result = rule.applyRule()<br>total_adjustment += result["scoreAdjustment"]<br>feedback_items.append(result["feedback"])<br>applied_rules.append(rule)<br>return {<br>"applied_rules": applied_rules,<br>"total_adjustment": total_adjustment,<br>"feedback_items": feedback_items<br>}<br>def validateSubmission(self, submission):<br>if not submission.answerContent or len(submission.answerContent.strip()) == 0:<br>return False, "Ответ не может быть пустым"<br>if submission.attemptNumber <= 0:<br>return False, "Некорректный номер попытки"<br>return True, "Валидация пройдена"<br><br>def __str__(self):<br>return f"RuleApplier {self.applierId} with {len(self.rules)} rules"<br>class AssessmentResult:<br>def __init__(self, resultId, submission, finalScore, detailedFeedback, improvementSuggestions):<br>self.resultId = resultId<br>self.submission = submission<br>self.finalScore = finalScore<br>self.detailedFeedback = detailedFeedback<br>self.improvementSuggestions = improvementSuggestions<br>def getDetailedFeedback(self):<br>return self.detailedFeedback<br>def __str__(self):<br>suggestions = "\n".join([f" - {suggestion}" for suggestion in self.improvementSuggestions])<br>return f"Assessment Result {self.resultId}:\nScore: {self.finalScore}/100\nFeedback: {self.detailedFeedback}\nSuggestions:\n{suggestions}<br>class AssessmentEngine:<br>def __init__(self, engineId, gradingMode="auto"):<br>self.engineId = engineId<br>self.gradingMode = gradingMode |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | self.ruleApplier = RuleApplier("RULER_001")<br>def evaluateSubmission(self, submission, student):<br>is_valid, validation_message = self.ruleApplier.validateSubmission(submission)<br>if not is_valid:<br>return AssessmentResult(<br>f"RES_{submission.submissionId}",<br>submission,<br>0,<br>f"Ошибка валидации: {validation_message}",<br>["Исправьте ошибки валидации и попробуйте снова"]<br>)<br>rule_results = self.ruleApplier.applyRules(submission, student)<br>final_score = self.calculateFinalScore(submission,<br>rule_results["total_adjustment"])<br>detailed_feedback = self.generateFeedback(rule_results["feedback_items"])<br>improvement_suggestions =<br>self.generateImprovementSuggestions(rule_results["applied_rules"])<br>return AssessmentResult(<br>f"RES_{submission.submissionId}",<br>submission,<br>final_score,<br>detailed_feedback,<br>improvement_suggestions<br>)<br>def calculateFinalScore(self, submission, total_adjustment):<br>base_score = 80<br>final_score = base_score + total_adjustment<br>return max(0, min(100, final_score))<br>def generateFeedback(self, feedback_items):<br>if not feedback_items:<br>return "Работа соответствует базовым требованиям. Хорошая работа!"<br>return ". ".join(feedback_items) + "."<br>def generateImprovementSuggestions(self, applied_rules):<br>suggestions = []<br>for rule in applied_rules:<br>if rule.scoreAdjustment < 0:<br>suggestions.append(f"Обратите внимание на: {rule.feedbackTemplate}")<br>if not suggestions:<br>suggestions.append("Продолжайте в том же духе! Ваш код соответствует<br>лучшим практикам.")<br>return suggestions |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | def __str__(self):<br>return f"AssessmentEngine {self.engineId} ({self.gradingMode} mode)" |
| **Раздел №3 «Интеллектуальный анализ данных»**<br>Тема 3.1<br>Стандартизация в области интеллектуального анализа данных | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.2 Уметь:** Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | **Задание №5**<br>Разработка плана тестирования и авто-тестов для системы рекомендации учебных материалов<br>Цель: Проверить корректность работы системы рекомендации учебных материалов путем разработки комплексного плана тестирования и создания модульных авто-тестов.<br>Задачи:<br>● Разработать стратегию тестирования системы<br>● Создать план тестирования с указанием тестовых сценариев<br>● Реализовать авто-тест для класса LearningMaterial с проверкой всех методов<br>● Обеспечить покрытие тестами boundary conditions и edge cases<br>Требования:<br>● Использовать фреймворк unittest<br>● Покрыть тестами не менее 80% функциональности класса<br>● Включить позитивные и негативные тестовые сценарии<br>● Проверить обработку граничных условий<br>Система: UML-диаграмма и код программы из практических заданий №1 и №3.<br>Вариант правильного ответа:<br>Примерный план проведения тестирования интеллектуальной системы<br>Модульное тестирование (Unit Testing)<br>Класс Student:<br>● Создание объекта с корректными данными<br>● Проверка методов getKnowledgeLevel() и getProgress()<br>● Проверка строкового представления<br>● Обработка некорректных данных прогресса (>100%, <0%)<br>Класс LearningMaterial:<br>● Создание объекта с разными уровнями сложности |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | ● Проверка метода isSuitableFor() для всех комбинаций уровней<br>● Проверка обработки неизвестных уровней<br>● Проверка строкового представления<br>Класс RecommendationRule:<br>● Проверка применения правил для разных условий<br>● Проверка приоритетов правил<br>● Тестирование невалидных условий<br>● Проверка возвращаемого действия<br>Класс Recommendation:<br>● Создание рекомендации с разной уверенностью<br>● Добавление материалов в рекомендацию<br>● Проверка временной метки<br>● Проверка строкового представления<br>Класс InferenceEngine:<br>● Добавление и сортировка правил по приоритету<br>● Генерация рекомендаций для разных студентов<br>● Расчет уверенности рекомендаций<br>● Обработка случаев без подходящих правил<br>Функциональное тестирование<br>1. Тест-кейс 1: Рекомендации для начинающего студента<br>a. Вход: Студент beginner, прогресс 30%, предпочтения ["video"]<br>b. Ожидаемый результат: Рекомендации только beginner материалов типа "video"<br>2. Тест-кейс 2: Рекомендации для успешного студента<br>a. Вход: Студент intermediate, прогресс 85%, предпочтения ["text", "quiz"]<br>b. Ожидаемый результат: Рекомендации intermediate материалов с высокой уверенностью<br>3. Тест-кейс 3: Отсутствие рекомендаций<br>a. Вход: Студент beginner, только advanced материалы |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | b. Ожидаемый результат: Пустой список рекомендаций<br>Итоговый авто-тест для класса LearningMaterial:<br>import unittest<br>import sys<br>import os<br>sys.path.append(os.path.dirname(os.path.abspath(__file__)))<br>from recommendation_system import LearningMaterial<br>class TestLearningMaterial(unittest.TestCase):<br>def setUp(self):<br>"""Подготовка тестовых данных перед каждым тестом"""<br>self.beginner_material = LearningMaterial(<br>"M001", "Основы Python", "beginner", "video", "beginner"<br>)<br>self.intermediate_material = LearningMaterial(<br>"M002", "ООП в Python", "intermediate", "text", "intermediate"<br>)<br>self.advanced_material = LearningMaterial(<br>"M003", "Паттерны проектирования", "advanced", "project", "advanced"<br>)<br>def test_constructor(self):<br>"""Тестирование корректного создания объекта"""<br>self.assertEqual(self.beginner_material.materialId, "M001")<br>self.assertEqual(self.beginner_material.title, "Основы Python")<br>self.assertEqual(self.beginner_material.difficulty, "beginner")<br>self.assertEqual(self.beginner_material.contentType, "video")<br>self.assertEqual(self.beginner_material.requiredLevel, "beginner")<br>def test_get_difficulty(self):<br>"""Тестирование метода getDifficulty"""<br>self.assertEqual(self.beginner_material.getDifficulty(), "beginner")<br>self.assertEqual(self.intermediate_material.getDifficulty(), "intermediate")<br>self.assertEqual(self.advanced_material.getDifficulty(), "advanced")<br>def test_is_suitable_for_beginner_student(self):<br>"""Тестирование подходящих материалов для начинающего студента"""<br>self.assertTrue(self.beginner_material.isSuitableFor("beginner"))<br>self.assertFalse(self.intermediate_material.isSuitableFor("beginner"))<br>self.assertFalse(self.advanced_material.isSuitableFor("beginner"))<br>def test_is_suitable_for_intermediate_student(self):<br>"""Тестирование подходящих материалов для студента среднего уровня"""<br>self.assertTrue(self.beginner_material.isSuitableFor("intermediate")) |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | self.assertTrue(self.intermediate_material.isSuitableFor("intermediate"))<br>self.assertFalse(self.advanced_material.isSuitableFor("intermediate"))<br>def test_is_suitable_for_advanced_student(self):<br>"""Тестирование подходящих материалов для продвинутого студента"""<br>self.assertTrue(self.beginner_material.isSuitableFor("advanced"))<br>self.assertTrue(self.intermediate_material.isSuitableFor("advanced"))<br>self.assertTrue(self.advanced_material.isSuitableFor("advanced"))<br>def test_is_suitable_for_invalid_level(self):<br>"""Тестирование обработки невалидных уровней"""<br>with self.assertRaises(ValueError):<br>self.beginner_material.isSuitableFor("expert")<br>with self.assertRaises(ValueError):<br>self.beginner_material.isSuitableFor("")<br>def test_string_representation(self):<br>"""Тестирование строкового представления объекта"""<br>expected_str = "Material M001: Основы Python (beginner)"<br>self.assertEqual(str(self.beginner_material), expected_str)<br>if __name__ == '__main__':<br>unittest.main(verbosity=2)<br><br>**Задание №6**<br>Разработка плана тестирования и авто-тестов для системы автоматической проверки заданий<br>Цель: Проверить корректность работы системы автоматической оценки студенческих заданий путем разработки комплексного плана тестирования и создания модульных авто-тестов.<br>Задачи:<br>● Разработать стратегию тестирования системы автоматической оценки<br>● Создать детальный план тестирования с тестовыми сценариями<br>● Реализовать авто-тест для класса GradingRule с проверкой всех методов<br>● Обеспечить покрытие тестами boundary conditions и edge cases<br>Требования:<br>● Использовать фреймворк unittest |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | ● Покрыть тестами не менее 85% функциональности класса<br>● Включить позитивные, негативные и граничные тестовые сценарии<br>● Проверить обработку различных условий применения правил<br>Система: UML-диаграмма и код программы из практических заданий №2 и №4.<br>Вариант правильного ответа:<br>Примерный план проведения тестирования интеллектуальной системы<br>Модульное тестирование (Unit Testing)<br>Класс Student:<br>● Создание объекта с различными данными<br>● Расчет среднего балла (с историей, без истории, пустая история)<br>● Проверка строкового представления<br>● Обработка некорректных данных<br>Класс Assignment:<br>● Создание заданий разных типов<br>● Проверка критериев оценки<br>● Строковое представление<br>Класс Submission:<br>● Создание отправки с различными данными<br>● Проверка временной метки<br>● Валидация номера попытки<br>● Строковое представление<br>Класс GradingRule:<br>● Все условия применения правил<br>● Разные типы заданий<br>● Граничные случаи студентов<br>● Корректность применения правил<br>Класс RuleApplier:<br>● Добавление и применение множества правил |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | ● Расчет общего adjustment<br>● Генерация feedback<br>● Валидация отправок<br>Класс AssessmentResult:<br>● Создание результатов с разными данными<br>● Форматирование обратной связи<br>● Генерация предложений по улучшению<br>Класс AssessmentEngine:<br>● Полный цикл оценки<br>● Обработка невалидных отправок<br>● Расчет итогового балла<br>● Генерация персонализированной обратной связи<br>Функциональное тестирование<br>1. Тест-кейс 1: Оценка корректного кода<br>a. Вход: Код с хорошими именами, с return, без ошибок<br>b. Ожидаемый результат: Высокий балл, позитивная обратная связь<br>2. Тест-кейс 2: Оценка кода с ошибками<br>a. Вход: Код без return, с print, плохие имена<br>b. Ожидаемый результат: Сниженный балл, конструктивная критика<br>3. Тест-кейс 3: Оценка для отличника<br>a. Вход: Студент с высокой успеваемостью<br>b. Ожидаемый результат: Бонусные баллы, упоминание успехов<br>4. Тест-кейс 4: Невалидная отправка<br>a. Вход: Пустой ответ<br>b. Ожидаемый результат: Ошибка валидации, 0 баллов<br>Итоговый авто-тест для класса GradingRule:<br>import unittest<br>import sys<br>import os |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | sys.path.append(os.path.dirname(os.path.abspath(__file__)))<br>from grading_system import GradingRule, Student, Assignment, Submission<br>class TestGradingRule(unittest.TestCase):<br>  def setUp(self):<br>  """Подготовка тестовых данных перед каждым тестом"""<br>  self.high_performer = Student("STU001", "Отличник", [90, 95, 92], "visual")<br>  self.low_performer = Student("STU002", "Студент", [65, 70, 60], "auditory")<br>  self.programming_assignment = Assignment(<br>  "ASS001", "Программирование", "programming", 100, ["syntax", "logic"]<br>  )<br>  self.quiz_assignment = Assignment(<br>  "ASS002", "Тест", "quiz", 50, ["accuracy"]<br>  )<br>  self.submission_with_return = Submission(<br>  "SUB001", self.high_performer, self.programming_assignment,<br>  "def calculate_sum(a, b):\n  return a + b", 1<br>  )<br>  self.submission_without_return = Submission(<br>  "SUB002", self.low_performer, self.programming_assignment,<br>  "def sum(a, b):\n  result = a + b", 1<br>  )<br>  def test_missing_return_condition(self):<br>  """Тестирование условия отсутствия return"""<br>  rule = GradingRule("R001", "programming", "missing_return", -20,<br>  "Добавьте return")<br>  self.assertTrue(rule.isApplicable(self.submission_without_return,<br>self.low_performer))<br>  self.assertFalse(rule.isApplicable(self.submission_with_return,<br>self.high_performer))<br><br>  def test_high_performer_condition(self):<br>  """Тестирование условия для успешных студентов"""<br>  rule = GradingRule("R002", "all", "high_performer", 5, "Бонус")<br>  self.assertTrue(rule.isApplicable(self.submission_with_return,<br>self.high_performer))<br>  self.assertFalse(rule.isApplicable(self.submission_without_return,<br>self.low_performer))<br>  def test_apply_rule_returns_correct_structure(self):<br>  """Тестирование структуры возвращаемых данных"""<br>  rule = GradingRule("R003", "programming", "missing_return", -15, "Фикс")<br>  result = rule.applyRule() |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Задания к практическому занятию |
|---|---|---|---|
| | | | self.assertIn("scoreAdjustment", result)<br>self.assertIn("feedback", result)<br>self.assertEqual(result["scoreAdjustment"], -15)<br>self.assertEqual(result["feedback"], "Фикс")<br>if __name__ == '__main__':<br>unittest.main(verbosity=2) |

***Критерии оценивания практических занятий:***

| Оценка | Критерии оценивания |
|---|---|
| **отлично** | Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи, формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя |
| **хорошо** | Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, но содержание ответов имеют некоторые неточности и требуют уточнения и комментария со стороны преподавателя |
| **удовлетворительно** | Выставляется, если обучающийся знает и понимает материал по заданной теме, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы преподавателя |
| **неудовлетворительно** | Выставляется, если обучающийся допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал. Делает ошибки в ответах на уточняющие вопросы преподавателя |

### 4.3. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ КОНТРОЛЬНЫХ РАБОТ

Контрольные работы содержат несколько практических заданий по индивидуальным вариантам, в полном объеме охватывающих изученный материал по указанной теме. Выполнение контрольных работ позволяет определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций.

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| **Тема 1.3** Моделирование знаний о предметных областях как основа интеллектуальных систем | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.2 Уметь:** Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | **Задание 1.1**<br>Вопрос: Какие из перечисленных технологий и языков программирования являются наиболее релевантными для разработки компонентов интеллектуальных систем, связанных с хранением, извлечением и логическим выводом на основе знаний?<br>**1.** Prolog<br>**2.** HTML<br>**3.** SQL<br>**4.** SPARQL<br>Правильные ответы: 1, 3, 4<br>Пояснение:<br>● Prolog является правильным ответом, так как это язык |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | логического программирования, созданный specifically для работы с символическими знаниями и правилами логического вывода, что является основой многих экспертных систем.<br>● SQL является правильным ответом, так как это основной язык для работы с реляционными базами данных, которые часто используются для хранения структурированных фактов и онтологий в информационных системах.<br>● SPARQL является правильным ответом, так как это язык запросов к семантическим базам данных (RDF), который является ключевым технологическим стеком для Веба Данных и сложных онтологий.<br>● HTML является неправильным ответом, так как это язык разметки для создания веб-страниц, а не для программирования логики или работы со знаниями.<br><br>**Задание 1.2**<br>Вопрос: Для разработки какой компонентной архитектуры современной информационной системы можно эффективно использовать следующие отечественные программные продукты?<br>**1.** PostgreSQL<br>**2.** 1С:Предприятие<br>**3.** Adobe Photoshop<br>**4.** Astra Linux<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● PostgreSQL является правильным ответом, так как это мощная СУБД (российская версия — Postgres Pro), которая может служить backend'ом для хранения данных и моделей знаний интеллектуальной системы.<br>● 1С:Предприятие является правильным ответом, так как эта платформа позволяет моделировать сложные бизнес-процессы и предметные области, что является формой представления знаний, и может интегрироваться с другими компонентами ИС. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | ● Astra Linux является правильным ответом, так как это сертифицированная отечественная ОС, на которой может быть развернута вся инфраструктура информационной системы, включая интеллектуальные модули.<br>● Adobe Photoshop является неправильным ответом, так как это графический редактор, не имеющий отношения к разработке архитектуры информационных систем.<br>**Задание 2.1**<br>Вопрос: Установите логическую последовательность этапов создания базы знаний для интеллектуальной системы.<br>1. Формализация знаний (создание онтологии, продукционных правил).<br>2. Извлечение знаний у экспертов предметной области.<br>3. Верификация и тестирование базы знаний на реальных задачах.<br>4. Выбор способа представления знаний (фреймы, онтологии, правила).<br>5. Реализация базы знаний в выбранной программной среде.<br>Правильный порядок: 2, 4, 1, 5, 3<br>Пояснение: Процесс начинается со сбора информации у экспертов (2). На основе этой информации выбирается подходящий метод ее структурирования (4). После выбора метода знания формализуются согласно этому методу (1). Затем формализованная модель реализуется в виде программного кода или конфигурации (5). Финальный этап — проверка работоспособности созданной системы (3).<br>**Задание 2.2**<br>Вопрос: Установите последовательность уровней абстракции при моделировании предметной области в архитектуре информационной системы, от физического хранения до пользовательского представления.<br>1. Уровень логической схемы (таблицы, сущности, атрибуты в СУБД).<br>2. Уровень физической схемы (файлы данных, индексы на диске).<br>3. Уровень представления (пользовательский интерфейс, отчеты, |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | API).<br>4. Уровень концептуальной схемы (ER-диаграмма, онтология).<br>Правильный порядок: 2, 1, 4, 3<br>Пояснение: Классическая трехуровневая архитектура СУБД (ANSI/SPARC) и систем моделирования знаний начинается с физического хранения (2). Над ним надстраивается уровень логического представления данных для системы (1). Концептуальная схема описывает предметную область в целом, абстрагируясь от деталей реализации (4). Уровень представления адаптирует данные и логику для конечного пользователя или внешней системы (3)<br><br>**Задание** 3.1<br>Установите соответствие между концепцией моделирования знаний и её описанием или примером технологии.<br><br>Список А (Концепция):<br>1. Онтология<br>2. Фрейм<br>3. Семантическая сеть<br>4. Логический вывод<br><br>Список Б (Описание / Технология):<br>A. Продукционное правило "ЕСЛИ... ТО..."<br>B. Язык Web Ontology Language (OWL)<br>C. Структура данных со слотами для описания стереотипного объекта<br>D. Граф, узлы которого — понятия, а дуги — отношения между ними<br>E. Процесс получения новых фактов на основе существующих с помощью механизма, такого как RacerPro<br><br>Ответ: 1B, 2C, 3D, 4E<br><br>Пояснение: Онтология формально описывает предметную область, и |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | OWL является стандартным языком для этого. Фрейм — это концепция из ИИ для представления объекта через его атрибуты (слоты). Семантическая сеть — это модель знаний в виде графа. Логический вывод — это процесс применения правил к существующим знаниям для генерации новых. Вариант А описывает форму правила, но не является исчерпывающим описанием вывода.<br><br>**Задание 3.2**<br>Вопрос: Установите соответствие между типами программного обеспечения и их ролью в контексте разработки информационных систем, основанных на знаниях.<br>Список А (Тип ПО / Язык):<br>1. SQL<br>2. Prolog<br>3. Git<br>4.Docker<br><br>Список Б (Роль в разработке ИС):<br>A. Язык логического программирования для создания ядра экспертной системы<br>B. Инструмент для управления версиями кода и совместной разработки<br>C. Язык запросов для извлечения структурированных данных из реляционной БД<br>D. Среда выполнения для изоляции и развертывания компонентов системы<br><br>Ответ: 1C, 2A, 3B, 4D<br><br>Пояснение: Соответствие строится на основной функции каждого инструмента. SQL работает с данными в БД. Prolog предназначен для символических вычислений и логики. Git необходим для командной работы над любым ПО. Docker стандартизирует процесс развертывания. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | **Задание 4.1**<br>Вопрос: Какой язык программирования, основанный на формальной логике и исчислении предикатов, традиционно считается одним из основных для создания "символического" ядра интеллектуальных систем и прототипов экспертных систем?<br>1. Java<br>2. Prolog<br>3. C++<br>4. Python<br>Правильный ответ: 2<br>Пояснение:<br>● Prolog является правильным ответом, так как его парадигма — логическое программирование — напрямую соответствует задачам представления знаний (в виде фактов и правил) и автоматического логического вывода на их основе.<br>● Java, C++, Python являются неправильными ответами. Это языки общего назначения (императивные или объектно-ориентированные). Хотя на них *можно* реализовать интеллектуальную систему (и на Python это часто делается для ML), они не являются специализированными языками для символического ИИ и логического вывода, в отличие от Prolog.<br><br>**Задание 4.2**<br>Вопрос: Какая из перечисленных открытых объектно-реляционных СУБД, имеющая российскую версию (Postgres Pro), является мощным инструментом для хранения сложных структурированных данных и часто используется как компонент backend'а для информационных систем, в том числе с элементами ИИ?<br>1. SQLite<br>2. MySQL |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | 3. PostgreSQL<br>4. MongoDB<br>Правильный ответ: 3<br>Пояснение:<br>● PostgreSQL является правильным ответом, так как это именно та СУБД, которая известна своей надежностью, поддержкой сложных типов данных и продвинутых функций SQL, и на основе которой создана российская Postgres Pro.<br>● SQLite является неправильным ответом, так как это встраиваемая легковесная БД, не предназначенная для высоконагруженных информационных систем.<br>● MySQL является неправильным ответом. Хотя это популярная СУБД, исторически она была менее функциональной в части сложных типов данных и стандартов SQL по сравнению с PostgreSQL.<br>● MongoDB является неправильным ответом, так как это документоориентированная NoSQL база данных, которая использует принципиально другую, нереляционную, модель данных.<br>**Задание 5.1**<br>Вопрос: Как называется стандартный язык World Wide Web Consortium (W3C), предназначенный для декларативного описания онтологий в рамках семантической паутины (Semantic Web) и моделирования знаний о предметной области? Ответ состоит из аббревиатуры из трех букв.<br>Правильный ответ: OWL<br>Пояснение: OWL (Web Ontology Language) — это именно тот стандарт W3C, который позволяет формально описывать онтологии, включая классы, свойства, индивидов и отношения между ними, делая знания машиночитаемыми и пригодными для логического вывода.<br><br>**Задание 5.2**<br>Вопрос: Какая популярная среда разработки от компании JetBrains, имеющая российские корни, предоставляет полнофункциональную |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | поддержку таких языков, как Python, Java и SQL, и широко используется профессионалами для создания информационных систем? Ответ состоит из одного слова.<br>Правильный ответ: PyCharm (или IntelliJ IDEA)<br>Пояснение: Оба продукта являются корректными ответами. PyCharm — специализированная IDE для Python, который часто используется в ML и Data Science. IntelliJ IDEA — IDE для JVM-языков (Java, Kotlin), широко используемая в enterprise-разработке. JetBrains — компания с российскими основателями, и ее продукты являются отраслевым стандартом. |
| **Тема 2.3**<br>Управление поиском и его реализация в пространстве состояний | **ПК-7**<br>Способен осуществлять проектирование программных интерфейсов | **ИПК-7.2 Уметь:** Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | **Задание 1.1**<br>Вопрос: Какие из перечисленных стратегий управления поиском в пространстве состояний являются систематическими (полными), то есть гарантируют нахождение решения, если оно существует?<br>**1.** Случайный поиск<br>**2.** Поиск в ширину (BFS)<br>**3.** Поиск с возвратом (Backtracking)<br>**4.** Поиск по первому наилучшему соответствию (жадный)<br>Правильные ответы: 2, 3<br>Пояснение:<br>● Поиск в ширину (BFS) является правильным ответом, так как он систематически исследует все узлы уровня за уровнем, гарантируя нахождение пути.<br>● Поиск с возвратом (Backtracking) является правильным ответом, так как это метод полного перебора, который последовательно проверяет все возможные варианты, возвращаясь при тупиках.<br>● Случайный поиск является неправильным ответом, так как он не систематичен и может бесконечно блуждать в пространстве состояний, не находя решение.<br>● Поиск по первому наилучшему соответствию (жадный) является неправильным ответом, так как он руководствуется локально оптимальным выбором и может зациклиться или попасть в |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | локальный оптимум, не гарантируя нахождение решения.<br><br>**Задание 1.2**<br>Вопрос: Какие из перечисленных технологий и языков программирования эффективно подходят для реализации систем, осуществляющих поиск в больших пространствах состояний (например, для задач планирования или анализа графов)?<br>**1.** Язык Prolog с его встроенным механизмом унификации и поиска с возвратами.<br>**2.** Язык Python с библиотеками для работы с графами (например, networkx) и структурами данных.<br>3. Язык разметки HTML.<br>4. Системы управления базами данных (например, PostgreSQL) с поддержкой рекурсивных запросов (WITH RECURSIVE).<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● Prolog является правильным ответом, так как его вычислительная модель изначально предназначена для логического вывода и поиска с возвратами в пространстве, определяемом фактами и правилами.<br>● Python является правильным ответом, так как он позволяет гибко реализовывать различные алгоритмы поиска и имеет богатый набор библиотек для работы с графами и данными.<br>● Системы управления базами данных являются правильным ответом, так как они позволяют хранить огромные графы состояний на диске и эффективно обходить их с помощью рекурсивных запросов.<br>● HTML является неправильным ответом, так как это язык разметки веб-страниц и он не предназначен для программирования логики поиска.<br>**Задание 2.1**<br>Вопрос: Установите правильную последовательность этапов в обобщенном алгоритме поиска в пространстве состояний. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | 1. Проверка цели: Проверить, является ли текущее состояние целевым.<br>2. Выбор состояния: Выбрать следующее состояние для анализа из множества пограничных.<br>3. Инициализация: Определить начальное состояние и добавить его в множество пограничных.<br>4. Расширение: Применить все возможные операторы к текущему состоянию, чтобы получить новые состояния, и добавить их в множество пограничных.<br>5. Проверка на неудачу: Если множество пограничных пусто, завершить с неудачей.<br>Правильный порядок: 3, 2, 1, 4, 5<br>Пояснение: Алгоритм начинается с инициализации (3). Затем в цикле выбирается состояние из пограничных (2) и проверяется на соответствие цели (1). Если цель не достигнута, состояние расширяется (4). Цикл продолжается, пока не будет найдено решение или не будет исчерпано пространство поиска (5).<br><br>**Задание 2.2**<br>Вопрос: Расположите в логическом порядке этапы настройки серверной среды для развертывания высокопроизводительной системы поиска, использующей реляционную базу данных.<br>1. Установка и настройка СУБД (например, настройка индексов для ускорения запросов).<br>2. Разработка и размещение кода приложения, реализующего алгоритм поиска.<br>3. Выбор и установка операционной системы на сервере.<br>4. Наполнение базы данных тестовыми данными (графом состояний).<br>5. Создание схемы базы данных (таблицы для узлов и ребер).<br>Правильный порядок: 3, 1, 5, 4, 2<br>Пояснение: Сначала развертывается базовое программное обеспечение — ОС (3). Затем устанавливается и базово настраивается СУБД (1). После |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | этого создается структура для хранения данных (5), которая затем наполняется информацией (4). Финальный шаг — размещение прикладного кода, который будет использовать эту базу (2)<br>**Задание 3.1**<br>Установите соответствие между стратегией поиска и ключевой структурой данных, которая для нее используется.<br><br>Список А (Стратегия поиска):<br>1. Поиск в ширину (BFS)<br>2. Поиск в глубину (DFS)<br>3. Поиск с возвратом (Backtracking)<br>4. Поиск по A*<br><br>Список Б (Структура данных):<br>A. Стек (Stack)<br>B. Очередь (Queue)<br>C. Множество (Set) для отслеживания посещенных состояний<br>D. Приоритетная очередь (Priority Queue)<br><br>Ответ: 1B, 2A, 3A, 4D<br><br>Пояснение: BFS обрабатывает состояния в порядке поступления (очередь). DFS и Backtracking идут "вглубь", обрабатывая последнее добавленное состояние (стек). A* извлекает состояние с наивысшим приоритетом (приоритетная очередь). Структура C (множество) является вспомогательной для многих стратегий, но не основной управляющей структурой.<br><br>**Задание 3.2**<br>Установите соответствие между задачей управления поиском и технологическим решением для ее реализации. |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | Список А (Задача):<br>1. Хранение очень большого графа состояний, не помещающегося в оперативную память<br>2. Быстрое прототипирование и визуализация алгоритма поиска на графе<br>3. Обеспечение максимальной скорости вычислений для сложного поиска<br>4. Создание отказоустойчивого сервера для системы поиска в госсекторе<br><br>Список Б (Технологическое решение):<br>A. Реализация алгоритма на C++<br>B. Использование СУБД (напр., PostgreSQL)<br>C. Язык Python с библиотеками (NumPy, NetworkX)<br>D. Развертывание на отечественной ОС (Astra Linux)<br><br>Ответ: 1B, 2C, 3A, 4D<br><br>Пояснение: Для больших данных, не помещающихся в RAM, используют СУБД. Python идеален для быстрого создания прототипов. C++ обеспечивает высокую производительность. Astra Linux является стандартом для защищенных серверов в госсекторе РФ<br>**Задание 4.1**<br>Вопрос: Какая стратегия поиска характеризуется тем, что всегда выбирает для расширения самый глубокий узел в текущем направлении, используя структуру данных "стек"?<br>1. Поиск в ширину (Breadth-First Search)<br>2. Поиск в глубину (Depth-First Search)<br>3. Поиск по A*<br>4. Двунаправленный поиск (Bidirectional Search)<br>Правильный ответ: 2<br>Пояснение:<br>● Поиск в глубину (Depth-First Search) является правильным ответом, так как его стратегия — идти "вглубь" графа, пока это возможно, а при достижении тупика возвращаться (backtrack), что |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | реализуется с помощью стека.<br>● Поиск в ширину (Breadth-First Search) является неправильным ответом, так как он использует очередь и исследует граф "слой за слоем".<br>● Поиск по A* является неправильным ответом, так как он использует приоритетную очередь и эвристическую функцию для выбора направления.<br>● Двунаправленный поиск (Bidirectional Search) является неправильным ответом, так как это общая стратегия, которая может использовать либо BFS, либо DFS внутри себя.<br><br>**Задание 4.2**<br>Вопрос: Какая из перечисленных российских систем управления базами данных, основанная на открытом коде PostgreSQL, является предпочтительным выбором для хранения и обработки данных в государственных информационных системах, требующих высокой надежности и производительности?<br>1. MySQL<br>2. SQLite<br>3. Postgres Pro<br>4. Oracle Database<br>Правильный ответ: 3<br>Пояснение:<br>● Postgres Pro является правильным ответом, так как это российская коммерческая СУБД, созданная на основе PostgreSQL, обладающая улучшенной производительностью и входящая в реестр отечественного ПО.<br>● MySQL является неправильным ответом, так как это СУБД, принадлежащая американской компании Oracle Corporation.<br>● SQLite является неправильным ответом, так как это встраиваемая легковесная БД, не предназначенная для высоконагруженных серверных систем.<br>● Oracle Database является неправильным ответом, так как это |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | проприетарная СУБД американской корпорации Oracle.<br>**Задание 5.1**<br>Вопрос: Как называется стратегия поиска, при которой два параллельных поисковых процесса запускаются одновременно: один от начального состояния, а другой от целевого, с целью обнаружить совпадение на их границах? Ответ состоит из двух слов.<br>Правильный ответ: Двунаправленный поиск<br>Пояснение: Двунаправленный поиск (Bidirectional Search) — это стратегия, предназначенная для уменьшения сложности по времени и памяти. Вместо исследования всего пространства от начала до конца, она ведет два поиска навстречу друг другу, что может значительно сократить общее количество рассмотренных состояний.<br><br>**Задание 5.2**<br>Вопрос: Какая структура данных, работающая по принципу "первым пришел — первым ушел" (FIFO), является фундаментальной для реализации алгоритма поиска в ширину (BFS)? Ответ состоит из одного слова.<br>Правильный ответ: Очередь<br>Пояснение: Очередь (Queue) — это абстрактный тип данных, который строго следует дисциплине FIFO. Это свойство критически важно для BFS, который должен обрабатывать состояния в том порядке, в котором они были обнаружены, что и обеспечивает нахождение кратчайшего пути в ненагруженном графе |
| **Тема 3.1** Стандартизация в области интеллектуального анализа данных | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.2 Уметь:** Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | **Задание 1.1**<br>Вопрос: Какие из перечисленных стандартов и спецификаций имеют непосредственное отношение к процессу интеллектуального анализа данных (Data Mining) и организации этого процесса в рамках крупных проектов?<br>**1.** Спецификация CRISP-DM (Cross-Industry Standard Process for Data Mining)<br>**2.** Стандарт ISO/IEC 25012 (качество данных)<br>**3.** Стандарт IEEE 802.11 (Wi-Fi) |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | **4.** Язык разметки HTML5<br>Правильные ответы: 1, 2<br>Пояснение:<br>● CRISP-DM является правильным ответом, так как это наиболее распространенная и признанная в отрасли методология (де-факто стандарт), описывающая жизненный цикл проекта Data Mining.<br>● ISO/IEC 25012 является правильным ответом, так как этот международный стандарт определяет модель качества данных. Качество данных — критически важный аспект для успешного интеллектуального анализа.<br>● IEEE 802.11 является неправильным ответом, так как это стандарт для беспроводных локальных сетей (Wi-Fi), не имеющий отношения к процессам анализа данных.<br>● HTML5 является неправильным ответом, так как это язык разметки для веб-страниц, а не стандарт для организации процессов анализа данных.<br><br>**Задание 1.2**<br>Вопрос: Для реализации промышленного проекта по интеллектуальному анализу данных в соответствии с современными стандартами и использованием отечественного ПО могут потребоваться следующие компоненты:<br>**1.** Язык Python с библиотеками (Pandas, Scikit-learn) для реализации этапов ETL и построения моделей.<br>**2.** Российская СУБД (например, Postgres Pro) для хранения и первичной обработки данных.<br>**3.** Графический редактор для визуализации итоговых отчетов.<br>**4.** Платформа управления контейнерами (например, Docker) для стандартизации окружения и воспроизводимости экспериментов.<br>Правильные ответы: 1, 2, 4<br>Пояснение:<br>● Python с библиотеками является правильным ответом, так как это современный стандарт для прототипирования и реализации |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | алгоритмов машинного обучения и анализа данных.<br>● Российская СУБД Postgres Pro является правильным ответом, так как она соответствует требованиям импортозамещения и обладает всеми необходимыми функциями для работы с данными, включая расширения для машинного обучения.<br>● Платформа управления контейнерами является правильным ответом, так как она обеспечивает воспроизводимость результатов и стандартизацию рабочего окружения, что является важным аспектом промышленного внедрения моделей.<br>● Графический редактор является неправильным ответом, так как это ручной инструмент для дизайна, а не часть стандартизированного технологического стека для автоматизированного анализа данных.<br>**Задание 2.1**<br>Вопрос: Установите правильную последовательность фаз согласно методологии CRISP-DM.<br>1. Моделирование (Modeling): Выбор и применение различных алгоритмов моделирования и калибровка их параметров.<br>2. Понимание данных (Data Understanding): Сбор первичных данных и их первоначальное исследование.<br>3. Подготовка данных (Data Preparation): Формирование конечного набора данных для моделирования (очистка, преобразование, агрегация).<br>4. Постановка бизнес-задачи (Business Understanding): Определение целей и требований проекта с точки зрения бизнеса.<br>5. Внедрение (Deployment): Развертывание модели в производственной среде.<br>6. Оценка (Evaluation): Проверка того, что модель достигла целей проекта, и утверждение следующих шагов.<br>Правильный порядок: 4, 2, 3, 1, 6, 5<br>Пояснение: CRISP-DM начинается с понимания бизнес-целей (4). Затем следует сбор и изучение данных (2). После этого данные тщательно |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | подготавливаются (3). Только на следующем этапе выбираются и настраиваются модели (1). Построенные модели всесторонне оцениваются (6), и если они успешны, внедряются в эксплуатацию (5).<br><br>**Задание 2.2**<br>Вопрос: Расположите в логическом порядке этапы развертывания отечественной платформы для анализа данных в защищенном контуре.<br>1. Разработка ETL-процедур и скриптов анализа на Python.<br>2. Установка и настройка российской ОС (например, Astra Linux) на сервере.<br>3. Наполнение базы тестовыми данными и проведение приемочных испытаний.<br>4. Установка и настройка российской СУБД (например, Postgres Pro).<br>5. Проектирование архитектуры решения и схемы базы данных.<br>Правильный порядок: 5, 2, 4, 1, 3<br>Пояснение: Сначала проектируется архитектура (5). Затем подготавливается инфраструктура — устанавливается ОС (2). После этого развертывается система хранения и обработки данных — СУБД (4). Далее реализуется бизнес-логика анализа (1). Финальный этап — тестирование всего комплекса (3).<br><br>**Задание 3.1**<br>Установите соответствие между этапом методологии CRISP-DM и его основной задачей.<br><br>Список А (Этап CRISP-DM):<br>1. Business Understanding<br>2. Data Preparation<br>3. Modeling<br>4. Evaluation<br><br>Список Б (Основная задача): |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | A. Преобразование сырых данных в пригодный для моделирования формат (очистка, обогащение).<br>B. Интерпретация результатов моделирования с точки зрения бизнес-целей, утверждение перехода к внедрению.<br>C. Выявление в данных содержательных закономерностей с помощью алгоритмов машинного обучения.<br>D. Определение того, какие бизнес-проблемы должны быть решены с помощью анализа данных.<br><br>Ответ: 1D, 2A, 3C, 4B<br><br>Пояснение: Первый этап — это перевод бизнес-задачи на язык данных (1-D). Самый трудоемкий этап — подготовка данных (2-A). Моделирование — это ядро анализа (3-C). Оценка проверяет, решена ли исходная бизнес-задача (4-B).<br><br>**Задание 3.2**<br>Установите соответствие между компонентом технологического стека для анализа данных и его российской альтернативой.<br><br>Список А (Компонент стека):<br>1. Операционная система<br>2. Система управления базами данных<br>3. Язык программирования для анализа<br>4. Платформа для управления контейнерами<br><br>Список Б (Российская альтернатива):<br>A. Postgres Pro<br>B. Astra Linux<br>C. Python (как мировой стандарт, активно используемый в РФ)<br>D. Russian Docker Registry / Tarantool Kubernetes |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | Ответ: 1B, 2A, 3C, 4D<br><br>Пояснение: Astra Linux — ведущая отечественная серверная ОС (1-B). Postgres Pro — российская СУБД для хранения данных (2-A). Python является де-факто стандартом в Data Science и широко используется в России (3-C). В России развиваются собственные экосистемы на базе Kubernetes для управления контейнерами (4-D).**Задание 4.1**<br>Вопрос: Какая из перечисленных методологий является наиболее распространенным отраслевым стандартом (de facto standard), описывающим жизненный цикл проекта по интеллектуальному анализу данных и состоящим из шести основных фаз?<br>5. Agile SCRUM<br>6. CRISP-DM (Cross-Industry Standard Process for Data Mining)<br>7. ITIL (Information Technology Infrastructure Library)<br>8. Waterfall (Каскадная модель)<br>Правильный ответ: 2<br>Пояснение:<br>● CRISP-DM является правильным ответом, так как это именно та методология, которая была разработана консорциумом компаний специально для проектов Data Mining и стала отраслевым стандартом благодаря своей универсальности и полноте.<br>● Agile SCRUM является неправильным ответом, так как это гибкая методология управления проектами общего назначения, а не специализированный стандарт для анализа данных.<br>● ITIL является неправильным ответом, так как это библиотека лучших практик для управления ИТ-услугами, а не для реализации проектов по анализу данных.<br>● Waterfall является неправильным ответом, так как это общая модель жизненного цикла ПО, не адаптированная под специфические итеративные процессы анализа данных.<br><br>**Задание 4.2**<br>Вопрос: Какая из перечисленных российских операционных систем, |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | соответствующая требованиям безопасности и входящая в реестр Минцифры, предпочтительна для развертывания серверной части аналитической платформы, работающей с данными ограниченного доступа?<br>1. Windows Server 2022<br>2. Ubuntu Server 22.04 LTS<br>3. Astra Linux<br>4. Red Hat Enterprise Linux<br>Правильный ответ: 3<br>Пояснение:<br>● Astra Linux является правильным ответом, так как это отечественная ОС, специально разработанная для создания защищенных информационных систем, работающих с конфиденциальной информацией и гостайной. Она имеет все необходимые сертификаты ФСТЭК России.<br>● Windows Server 2022 является неправильным ответом, так как это продукт американской компании Microsoft.<br>● Ubuntu Server и Red Hat Enterprise Linux являются неправильными ответами, так как это продукты международных (фактически американских) компаний Canonical и Red Hat (IBM) соответственно.<br><br>**Задание 5.1**<br>Вопрос: Как называется наиболее известная иерархическая модель стандартов, связанных с качеством данных, которая включает такие характеристики, как точность, полнота, актуальность и согласованность? Ответ должен состоять из аббревиатуры и номера стандарта (например, ISO XXXX).<br>Правильный ответ: ISO 25012<br><br>Пояснение: ISO/IEC 25012 — это международный стандарт, который |

| Тема | Код и формулировка компетенции | Индикаторы достижения компетенции | Типовой вариант контрольной работы |
|---|---|---|---|
| | | | определяет модель качества данных. Он описывает характеристики качества данных, которые могут быть использованы для оценки пригодности данных для анализа, что является критически важным первым шагом в любом проекте интеллектуального анализа данных.<br><br>**Задание 5.2**<br>Вопрос: Какой открытый язык программирования с богатой экосистемой библиотек (таких как Pandas, Scikit-learn, TensorFlow) является де-факто отраслевым стандартом для реализации задач интеллектуального анализа данных и машинного обучения? Ответ состоит из одного слова.<br>Правильный ответ: Python<br>Пояснение: Python является доминирующим языком в области Data Science и Machine Learning благодаря своему простому синтаксису, активному сообществу и огромному количеству специализированных библиотек для обработки данных, визуализации и построения моделей. Его использование стало стандартом для быстрого прототипирования и промышленной реализации аналитических решений |

***Критерии оценивания контрольной работы:***

| Оценка | Критерии оценивания |
|---|---|
| **отлично** | Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи, формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, обосновывает свои суждения и даёт<br>правильные ответы на вопросы преподавателя |
| **хорошо** | Выставляется, если обучающийся умеет увязывать теорию с практикой (решает задачи и формулирует выводы, умеет пояснить полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом по заданной теме, но содержание ответов имеют некоторые неточности и требуют уточнения и комментария со стороны преподавателя. |
| **удовлетворительно** | Выставляется, если обучающийся знает и понимает материал по заданной теме, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы преподавателя |

| | |
|---|---|
| **неудовлетворительно** | Выставляется, если обучающийся допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал. Делает ошибки в ответах на уточняющие вопросы преподавателя |

### 4.4. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа включает в себя проработку теоретического материала, изучение рекомендуемой литературы, выполнение практико-ориентированных заданий (заполнение таблиц, проведение сравнительного анализа, составление схем и др.), решение практических задач, создание презентаций, написание рефератов, подборка нормативного и иного материала и выполнение других заданий.

| **Тема** | **Код и формулировка компетенции** | **Индикаторы достижения компетенции** | **Типовое задание для самостоятельной работы** |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **1.1** Искусственный интеллект: предпосылки, развитие, современное состояние разработок | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов<br>**ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | Подготовьте реферат на тему<br>**Вариант 1**<br>“Зарождение искусственного интеллекта”<br>Что нужно отразить:<br>Период (1950-е – 1960-е): Появление термина (Дартмутская конференция, 1956), работы Тьюринга, Ньюэлла, Шоу, Саймона. Эйфория, создание первых программ для игр и доказательства теорем.<br>**Вариант 2**<br>“Развитие Экспертных систем”<br>Что нужно отразить:<br>Период (1970-е – 1980-е): Сдвиг от общих алгоритмов к системам, основанным на знаниях (MYCIN, DENDRAL). Коммерческий успех и последующий спад ("зима ИИ") из-за завышенных ожиданий.<br>**Вариант 3**<br>“Возрождение и статистический подход к ии”<br>Что нужно отразить:<br>Период (1990-е – 2000-е): Широкое использование вероятностных методов, машинного обучения, агентных подходов. Победа Deep Blue над Каспаровым.<br>**Вариант 4**<br>“Глубокое обучение ИИ и Big Data”<br>Что нужно отразить:<br>Период (2010-е – по н.в.): Прорыв в областях компьютерного зрения, обработки естественного языка (NLP) благодаря глубоким нейронным сетям и доступности больших данных.<br>**Вариант 5**<br>“Области применения ИИ”:<br>Что нужно отразить:<br>Медицина (диагностика), финансы (кредитный скоринг, алготрейдинг), робототехника, беспилотные транспортные средства, рекомендательные системы, кибербезопасность. |

| | | | |
|---|---|---|---|
| **Тема 1.3** Моделирование знаний о предметных областях как основа интеллектуальных систем | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов<br>**ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | Подготовьте реферат на тему<br>**Вариант 1**<br>“Продукционная модель.”<br>Что нужно отразить:<br>Продукционная модель — одна из наиболее распространенных моделей, в которой знания представляются в виде набора правил (продукций).<br>Структура правила: ЕСЛИ (условие, антецедент) ТО (действие, консеквент).<br>Условие: Факт или совокупность фактов, которые должны быть истинны.<br>Действие: Добавление нового факта, запуск процедуры, изменение данных.<br>Достоинства:<br>● Модульность и наглядность.<br>● Простота модификации и дополнения БЗ.<br>● Естественность для представления эвристических знаний.<br>Недостатки:<br>● Низкая эффективность при большом количестве правил.<br>● Сложность анализа целостности БЗ (противоречия, тупиковые ситуации).<br>Подготовьте реферат на тему<br>**Вариант 2**<br>“Фреймы”<br>Что нужно отразить:<br>Фреймовая модель представляет знания в виде иерархической сети структур данных, называемых фреймами.<br>Фрейм — это структура для описания стереотипного объекта, ситуации или понятия. Состоит из:<br>1. Имя фрейма.<br>2. Слоты (атрибуты), которые характеризуют объект. Слоты могут иметь значения, типы данных, процедуры-демоны (выполняются при обращении к слоту) и указатели на другие фреймы.<br>Пример: Фрейм "СТУДЕНТ" имеет слоты: Имя, Группа, Средний_балл.<br>Достоинства:<br>● Наглядность и структурность.<br>● Эффективность представления иерархических знаний.<br>● Возможность использования наследования (дочерний фрейм наследует слоты родительского). |
| **Тема 2.7** Рассуждения в условиях неопределенности | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов<br>**ИПК-7.2** Уметь | Подготовьте реферат на тему<br><br>**Вариант 1**<br>“Логический метод рассуждения”<br>Что должно быть отражено:<br>Основан на применении формальных правил логики (например, исчисления предикатов). Вывод является строгим, детерминированным и формально доказуемым.<br>Пример: Modus Ponens: Если A → B и A истинно, то B истинно. |

| | | | |
|---|---|---|---|
| | | применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | Применение: В системах, где требуются строгие, гарантированные выводы (математические доказательства, верификация программ).<br><br>**Вариант 2**<br>“Эвристический метод рассуждения”:<br>Что должно быть отражено:<br>Основан на использовании эвристик — правдоподобных, но не строгих правил, основанных на опыте эксперта.<br>Вывод является вероятностным и не гарантирует абсолютной правильности.<br>Пример: "Если у пациента высокая температура и кашель, то, вероятно, это ОРВИ".<br>Применение: В экспертных системах для решения неформализованных задач в условиях неполной информации. |

***Критерии оценивания самостоятельной работы:***

| Оценка | Критерии оценивания |
|---|---|
| **отлично** | выставляется если работа носит научно-исследовательский характер, проанализирован и сделан сравнительный анализ нескольких литературных источников, приведены примеры |
| **хорошо** | выставляется если проанализирован и сделан сравнительный анализ нескольких литературных источников, приведены примеры |
| **удовлетворительно** | выставляется если проведен сравнительный анализ научно-методической литературы, приведены примеры |
| **неудовлетворительно** | выставляется если работа прошла проверку на антиплагиат и соответствует требованиям оформления |

## 4.5. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ ДЛЯ КУРСОВЫХ РАБОТ

**Курсовые работы не предусмотрены**

Курсовая работа позволяет выявить степень владения базовыми знаниями, умениями и навыками, необходимыми для обучения, и определить уровень владения новым материалом.

Примерные индивидуальные задания (темы) для курсовых работ:

***Критерии оценивания курсовой работы:***

| Оценка | Критерии оценивания |
|---|---|
| **отлично** | Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Представлены результаты структурированного и логически последовательного обзора литературных и иных источников по теме исследования. Структура курсовой работы логически и методически выдержана. Верно определены исходные данные для расчетов. Все аналитические расчеты выполнены верно, корректно применены методы экономического анализа, не нарушена методика анализа предмета исследования. Все выводы и предложения убедительно аргументированы. При защите курсовой работы обучающийся правильно и уверенно отвечает на вопросы преподавателя, демонстрирует глубокое знание теоретического материала, способен аргументировать собственные утверждения и выводы |

| | |
|---|---|
| **хорошо** | Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Представлены результаты структурированного и логически последовательного обзора литературных и иных источников по теме исследования. Структура курсовой работы логически и методически выдержана. Верно определены исходные данные для расчетов. В расчетах допускаются незначительные (не искажающие общего итога оценки) погрешности/ошибки. Большинство выводов и предложений аргументировано, корректно применены методы экономического анализа, не нарушена методика анализа предмета исследования. Оформление курсовой работы и полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две несущественные ошибки в использовании терминов, в построенных диаграммах и схемах, в оформлении таблиц. Наличествует незначительное количество грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся правильно и уверенно отвечает на большинство вопросов преподавателя, демонстрирует хорошее знание теоретического материала, но не всегда способен аргументировать собственные утверждения и выводы. При наводящих вопросах преподавателя исправляет ошибки в ответе |
| **удовлетворительно** | Содержание курсовой работы полностью соответствует заданию, содержащемуся в методических указаниях, и плану. Результаты обзора литературных и иных источников представлены недостаточно полно, недостаточно логично и последовательно. Верно определены исходные данные для расчетов, но имеются грубые ошибки в расчетах. Аргументация выводов и предложений слабая или отсутствует. Экономические выводы носят констатирующий (описательный) характер. Имеются одно-два существенных отклонений от требований в оформлении курсовой работы. Полученные результаты в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две существенных ошибки в использовании терминов, в построенных диаграммах и схемах. Много грамматических и/или стилистических ошибок. При защите курсовой работы обучающийся допускает грубые ошибки при ответах на вопросы преподавателя, демонстрирует слабое знание теоретического материала, в большинстве случаев не способен уверенно аргументировать собственные утверждения и выводы |
| **неудовлетворительно** | Содержание курсовой работы не соответствует заданию, содержащемуся в методических указаниях, и плану. Неверно определены исходные данные для расчетов, неверно и не корректно применены методы экономического анализа. Экономические выводы содержат неверную экономическую оценку. Имеются более двух существенных отклонений от требований в оформлении курсовой работы. Большое количество существенных ошибок по сути работы, много грамматических и стилистических ошибок и др. Полученные результаты не отвечают требованиям, изложенным в методических указаниях. При защите курсовой работы обучающийся демонстрирует слабое понимание программного материала, студент не может защитить свои решения, допускает грубые фактические ошибки при ответах на поставленные вопросы или вовсе не отвечает на них. Курсовая работа не представлена преподавателю. Обучающийся не явился на защиту курсовой работы |

## 5. МАТЕРИАЛЫ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Промежуточная аттестация проводится в форме зачета/зачета с оценкой/экзамена.

**5.1. Вопросы к экзамену:**

1. Понятие интеллектуальной информационной системы (ИИС), основные свойства.

2. Классификация ИИС.
3. Развитие исследований в области искусственного интеллекта (этапы; области применения).
4. Развитие исследований в области искусственного интеллекта (направления исследований; проблемы и перспективы).
5. Экспертные системы – основная разновидность прикладных интеллектуальных систем. Инженерия знаний.
6. Характеристика ЭС.
7. Архитектура ИИС: база знаний, механизм вывода, механизмы приобретения и объяснения знаний, интеллектуальный интерфейс.
8. Модели представления знаний. Продукционная модель.
9. Модели представления знаний. Фреймы.
10. Модели представления знаний. Семантическая модель.
11. Интеллектуализация поисковых процедур.
12. Организация базы знаний.
13. Логический и эвристический методы рассуждения в ИИС.
14. Нечеткий вывод знаний.
15. Стохастический подход к описанию неопределенности.
16. Технология проектирования и разработки интеллектуальных систем.
17. Стандартизация в области интеллектуального анализа данных.

***Критерии оценивания зачета с оценкой/экзамена:***

| Оценка | Критерии оценивания |
|---|---|
| **зачтено** | Обучающийся должен: уметь строить ответ в соответствии со структурой излагаемого вопроса; продемонстрировать прочное, достаточно полное усвоение знаний программного материала; продемонстрировать знание основных теоретических понятий; правильно формулировать определения; последовательно, грамотно и логически стройно изложить теоретический материал; продемонстрировать умения самостоятельной работы с литературой; уметь сделать достаточно обоснованные выводы по излагаемому материалу. |
| **не зачтено** | Обучающийся демонстрирует: незнание значительной части программного материала; не владение понятийным аппаратом дисциплины; существенные ошибки при изложении учебного материала; неумение строить ответ в соответствии со структурой излагаемого вопроса; неумение делать выводы по излагаемому материалу. |

***Критерии оценивания зачета с оценкой/экзамена:***

| Оценка | Критерии оценивания |
|---|---|
| **отлично** | Выставляется, если обучающимся правильно и полностью раскрыто содержание материала в пределах программы, чётко и правильно даны определения и раскрыто содержание понятий, точно использованы научные и технические термины, в ответе использованы ранее приобретённые теоретические знания, сделаны необходимые выводы и обобщения |
| **хорошо** | Выставляется, если обучающимся раскрыто основное содержание материала в пределах программы, даны определения и раскрыто содержание понятий, в ответе использованы ранее приобретённые теоретические знания, сделаны необходимые выводы и обобщения, но присутствуют незначительные нарушения в последовательности изложения, имеются одна-две |

| | |
|---|---|
| | неточности в содержании ответа |
| **удовлетворительно** | Выставляется, если обучающимся содержание учебного материала изложено фрагментарно, не всегда последовательно, не даны определения, не раскрыто содержание понятий, или они изложены с ошибками, допускаются ошибки и неточности в использовании научной терминологии, отсутствуют выводы и обобщения из предыдущего материала, или возможны ошибки в их изложении |
| **неудовлетворительно** | Выставляется, если обучающимся основное содержание учебного материала не раскрыто, не даются ответы на основные вопросы, допускаются грубые ошибки в определении понятий, в использовании терминологии, отсутствуют выводы и обобщения |

## 6. МАТЕРИАЛЫ ДЛЯ ДИАГНОСТИЧЕСКОЙ РАБОТЫ

Задания для диагностической работы должны обеспечивать оценку полностью или частично сформированных компетенций. Каждое задание должно быть привязано к тому или иному индикатору сформированности компетенций.

***При формировании заданий для диагностической работы необходимо использовать тестовые задания следующих типов***:

Тип задания 1. Задания закрытого типа на установление соответствия.

Тип задания 2. Задания закрытого типа на установление последовательности.

Тип задания 3. Задания комбинированного типа, предполагающие выбор одного правильного ответа из предложенных

Тип задания 4. Задания комбинированного типа, предполагающие выбор нескольких ответов из предложенных

Тип задания 5. Задания открытого типа с развернутым ответом.

**Все типы заданий должны быть представлены не менее одного раза.**

| № п/п | Тема занятия | Код компетенции | Индикатор | Тип задания | Задание | | Уровень освоения компетенции |
|---|---|---|---|---|---|---|---|
| | | | | | Вариант 1 | Вариант 2 | |
| 1. | **Тема 1.1** Искусственный интеллект: предпосылки, развитие, современное состояние разработок | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов **ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | 2,2 | **1.** Вопрос: Установите правильную последовательность этапов внедрения отечественного программного обеспечения в организации в соответствии с принципами импортозамещения.<br>1. Анализ и аудит существующей ИТ-инфраструктуры на наличие зарубежного ПО.<br>2. Пилотная эксплуатация выбранного российского ПО в одном из департаментов.<br>3. Массовый переход и полное развертывание отечественного решения по всей организации.<br>4. Тестирование и выбор подходящего российского аналога из реестра Минцифры.<br>5. Обучение сотрудников работе с новым программным обеспечением.<br>Правильный порядок: 1, 4, 2, 5, 3<br>Пояснение: Логика начинается с аудита (1), чтобы понять объем предстоящей работы. Далее следует выбор конкретного решения из одобренного перечня | **1.** Вопрос: Расположите в хронологическом порядке появление следующих ключевых технологий и сервисов в области искусственного интеллекта от российских компаний.<br>1. Запуск фреймворка SberVision для компьютерного зрения.<br>2. Создание нейросети «Крибрум» для анализа интернет-трафика.<br>3. Появление голосового помощника «Алиса» от Яндекса.<br>4. Разработка языковой модели GigaChat от Сбера.<br>Правильный порядок: 2, 3, 1, 4<br>Пояснение: Нейросеть «Крибрум» одна из первых получила известность в середине 2010-х. Голосовой помощник «Алиса» был представлен Яндексом в 2017 году. Фреймворк SberVision был анонсирован в рамках развития экосистемы SberAI в начале 2020-х. Модель GigaChat, как прямой аналог ChatGPT, появилась одной из последних, в 2023 году. | **Базовый (1-3 минуты)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | (4). Перед полным внедрением обязателен пилотный запуск (2) для выявления скрытых проблем. Параллельно или сразу после пилота проводится обучение пользователей (5), и только затем — полномасштабное развертывание (3). | | |
| 2. | **Тема 1.2**Данные и знания | **ПК-7** Способен осуществлять проектировани е программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов **ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | 1,1 | **2.** Какие из перечисленных технологий и языков используются для работы со структурированными данными и знаниями в системах искусственного интеллекта?<br>1. SQL для выполнения запросов к реляционным базам данных, хранящим факты и онтологии.<br>2. Язык OWL (Web Ontology Language) для формального описания знаний и онтологий.<br>3. Язык разметки HTML для создания веб-интерфейсов.<br>4. Фреймворк TensorFlow для работы с векторными представлениями слов (Word2Vec, GloVe).<br><br>Ответ: 1, 2, 4<br><br>Пояснение: SQL является правильным ответом, так это | **2.** Какие из перечисленных компонентов и технологий являются ключевыми для построения систем, основанных на знаниях (Knowledge-Based Systems)?<br>1. База данных PostgreSQL для хранения сырых транзакционных данных.<br>2. База знаний, содержащая правила и факты о предметной области.<br>3. Механизм логического вывода (Inference Engine).<br>4. Графический процессор (GPU) для ускорения матричных вычислений.<br><br>Ответ: 2, 3<br><br>Пояснение: База знаний является правильным ответом, так как это ядро любой системы, основанной на знаниях; она содержит символическое представление экспертных знаний. Механизм | **Повышенный (3-5 минут)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | основной язык для взаимодействия с реляционными БД, которые часто используются для хранения структурированных фактов в экспертных системах и других ИИ-приложениях. OWL является правильным ответом, так как это стандартный язык семантической паутины для создания сложных онтологий, позволяющих машинам "понимать" смысл данных. TensorFlow является правильным ответом, так как он используется для создания моделей, которые преобразуют текстовые данные (неструктурированные знания) в числовые векторные представления, с которыми может работать ИИ. | логического вывода является правильным ответом, так как это "мозг" системы, который применяет логические правила к базе знаний для получения новых фактов и ответов. | |
| 3. | **Тема 1.3** Моделирование знаний о предметных областях как основа интеллектуальных систем | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов **ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, | 3,3 | **3.** Установите соответствие между концепцией моделирования знаний и её описанием или примером технологии.<br><br>Список А (Концепция):<br>1. Онтология<br>2. Фрейм<br>3. Семантическая сеть<br>4. Логический вывод<br><br>Список Б (Описание / Технология):<br>А. Продукционное правило | **3.** Вопрос: Установите соответствие между типами программного обеспечения и их ролью в контексте разработки информационных систем, основанных на знаниях.<br>Список А (Тип ПО / Язык):<br>1. SQL<br>2. Prolog<br>3. Git<br>4.Docker<br><br>Список Б (Роль в разработке ИС):<br>А. Язык логического программирования для создания | **Базовый (1-3 минуты)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | программных интерфейсов | | "ЕСЛИ... ТО..."<br>B. Язык Web Ontology Language (OWL)<br>C. Структура данных со слотами для описания стереотипного объекта<br>D. Граф, узлы которого — понятия, а дуги — отношения между ними<br>E. Процесс получения новых фактов на основе существующих с помощью механизма, такого как RacerPro<br><br>Ответ: 1B, 2C, 3D, 4E<br><br>Пояснение: Онтология формально описывает предметную область, и OWL является стандартным языком для этого. Фрейм — это концепция из ИИ для представления объекта через его атрибуты (слоты). Семантическая сеть — это модель знаний в виде графа. Логический вывод — это процесс применения правил к существующим знаниям для генерации новых. Вариант A описывает форму правила, но не является исчерпывающим описанием вывода. | ядра экспертной системы<br>B. Инструмент для управления версиями кода и совместной разработки<br>C. Язык запросов для извлечения структурированных данных из реляционной БД<br>D. Среда выполнения для изоляции и развертывания компонентов системы<br><br>Ответ: 1C, 2A, 3B, 4D<br><br>Пояснение: Соответствие строится на основной функции каждого инструмента. SQL работает с данными в БД. Prolog предназначен для символических вычислений и логики. Git необходим для командной работы над любым ПО. Docker стандартизирует процесс развертывания | |
| 4. | **Тема 1.4** Основные компоненты | **ПК-7** Способен осуществлять проектировани | **ИПК-7.1** Знать методы и средства проектирования | 4,4 | **4.** Вопрос: Какой компонент интеллектуальной системы отвечает за применение продукционных правил (типа | **4.** Вопрос: Какая из перечисленных СУБД, имеющая российскую версию Postgres Pro, является надежным решением для | **Повышенный (3-5 минут)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | интеллектуальной информационной системы | е программных интерфейсов | программных интерфейсов<br>**ИПК-7.2**<br>Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | | "ЕСЛИ-ТО") к фактам из базы знаний для формирования новых фактов и принятия решений?<br>1. База знаний<br>2. Решатель (механизм вывода)<br>3. Пользовательский интерфейс<br>4. База данных<br>Правильный ответ: 2<br>Пояснение:<br>● Решатель (механизм вывода) является правильным ответом, так как это именно тот компонент, который содержит алгоритмы (прямая или обратная цепочка рассуждений) для манипуляции знаниями и получения новых результатов.<br>● База знаний является неправильным ответом, так как это пассивное хранилище информации, а не активный обработчик.<br>● Пользовательский интерфейс является неправильным ответом, так как он только обеспечивает взаимодействие с пользователем.<br>● База данных является | хранения структурированных данных, которые могут использоваться в качестве основы или компонента базы знаний интеллектуальной системы?<br>1. SQLite<br>2. PostgreSQL<br>3. Microsoft Access<br>4. MongoDB<br>Правильный ответ: 2<br>Пояснение:<br>● PostgreSQL является правильным ответом, так как это мощная, надежная и полнофункциональная объектно-реляционная СУБД, на основе которой создан отечественный продукт Postgres Pro. Она подходит для хранения сложных структур данных.<br>● SQLite является неправильным ответом, так как это встраиваемая БД для легковесных приложений, не предназначенная для высоконагруженных систем.<br>● Microsoft Access является неправильным ответом, так как это настольная СУБД, не рассчитанная на масштабирование и использование в качестве бэкенда для серьезных | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | неправильным ответом, так как это система для хранения структурированных данных, но не для выполнения логического вывода. | ИИС.<br>● MongoDB является неправильным ответом, так как это документоориентированная NoSQL база, которая использует другую модель данных и не является реляционной, что может быть критично для ряда задач логического вывода. | |
| 5. | **Тема 2.1** Структуры и стратегии поиска в пространстве состояний | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов<br>**ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | 5,5 | 5. Вопрос: Какая структура данных, работающая по принципу "первый пришел — первый ушел" (FIFO), является основной для реализации алгоритма поиска в ширину (BFS)? Ответ состоит из одного слова.<br>Правильный ответ: Очередь<br>Пояснение: Очередь (Queue) — это абстрактный тип данных, который точно следует дисциплине FIFO. Это свойство критически важно для BFS, который должен обрабатывать состояния в том порядке, в котором они были обнаружены, что и обеспечивает нахождение кратчайшего пути. | 5. Вопрос: Какой язык программирования, использующий механизм автоматического перебора с возвратами (backtracking) для поиска решений, основанных на заданных фактах и правилах, является классическим выбором для задач, формулируемых в терминах логики? Ответ состоит из одного слова.<br>Правильный ответ: Prolog<br>Пояснение: Prolog — это язык, чья вычислительная модель основана на автоматическом поиске с возвратами и унификации. Программа на Prolog описывает проблему в виде логических утверждений, а интерпретатор автоматически осуществляет поиск в пространстве возможных решений, что напрямую соотносится с поиском в пространстве состояний | Высокий (5-10 минут) |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6. | **Тема 2.2** Эвристический поиск | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов **ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | 3,3 | **6.** Установите соответствие между алгоритмом поиска и его ключевым свойством.<br><br>Список А (Алгоритм):<br>1. A*<br>2. Жадный поиск по первому наилучшему соответствию<br>3. Алгоритм Дейкстры<br>4. Поиск в ширину<br><br>Список Б (Свойство):<br>A. Оптимален и эффективен при использовании допустимой и монотонной эвристики<br>B. Использует только эвристическую функцию h(n), игнорируя стоимость достижения узла g(n).<br>C. Оптимален для поиска кратчайшего пути без использования эвристики (фактически, A* с h(n) = 0).<br>D. Является частным случаем алгоритма Дейкстры для графов с единичной стоимостью ребер.<br><br>Ответ: 1A, 2B, 3C, 4D<br><br>Пояснение: A* сочетает стоимость пройденного пути g(n) и эвристику h(n). Жадный поиск использует только h(n). Алгоритм Дейкстры можно рассматривать как A*, где эвристика равна нулю. BFS — | **6.** Установите соответствие между компонентом/концепцией в разработке ПО для эвристического поиска и примером технологии или инструмента.<br><br>Список А (Компонент/Концепция):<br>1. Язык для высокопроизводительной реализации вычислительно сложных алгоритмов<br>2. Инструмент для быстрого прототипирования и визуализации графов<br>3. Система для хранения большого графа и выполнения запросов к нему<br>4. Отечественная ОС для развертывания серверной части системы<br><br>Список Б (Технология/Инструмент):<br>A. Python с библиотекой heapq<br>B. C++ со STL (priority_queue)<br>C. СУБД PostgreSQL (с рекурсивными CTE)<br>D. Astra Linux<br><br>Ответ: 1B, 2A, 3C, 4D<br><br>Пояснение: C++ и его стандартная библиотека предоставляют эффективные структуры данных для высоконагруженных систем. | **Базовый (1-3 минуты)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | это Дейкстра, примененная к графу, где все ребра имеют одинаковую стоимость | Python известен скоростью разработки и богатыми библиотеками. PostgreSQL позволяет хранить и обрабатывать огромные объемы структурированных данных, включая графы. Astra Linux — это сертифицированная российская ОС для защищенных серверов. | |
| 7. | **Тема 2.3** Управление поиском и его реализация в пространстве состояний | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов **ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | 1,1 | **7.** Вопрос: Какие из перечисленных стратегий управления поиском в пространстве состояний являются систематическими (полными), то есть гарантируют нахождение решения, если оно существует?<br>1. Случайный поиск<br>2. Поиск в ширину (BFS)<br>3. Поиск с возвратом (Backtracking)<br>4. Поиск по первому наилучшему соответствию (жадный)<br>Правильные ответы: 2, 3<br>Пояснение:<br>● Поиск в ширину (BFS) является правильным ответом, так как он систематически исследует все узлы уровня за уровнем, гарантируя нахождение пути.<br>● Поиск с возвратом | **7.** Вопрос: Какие из перечисленных технологий и языков программирования эффективно подходят для реализации систем, осуществляющих поиск в больших пространствах состояний (например, для задач планирования или анализа графов)?<br>1. Язык Prolog с его встроенным механизмом унификации и поиска с возвратами.<br>2. Язык Python с библиотеками для работы с графами (например, networkx) и структурами данных.<br>3. Язык разметки HTML.<br>4. Системы управления базами данных (например, PostgreSQL) с поддержкой рекурсивных запросов (WITH RECURSIVE).<br>Правильные ответы: 1, 2, 4 | **Базовый (1-3 минуты)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | (Backtracking) является правильным ответом, так как это метод полного перебора, который последовательно проверяет все возможные варианты, возвращаясь при тупиках.<br>● Случайный поиск является неправильным ответом, так как он не систематичен и может бесконечно блуждать в пространстве состояний, не находя решение.<br>● Поиск по первому наилучшему соответствию (жадный) является неправильным ответом, так как он руководствуется локально оптимальным выбором и может зациклиться или попасть в локальный оптимум, не гарантируя нахождение решения. | Пояснение:<br>● Prolog является правильным ответом, так как его вычислительная модель изначально предназначена для логического вывода и поиска с возвратами в пространстве, определяемом фактами и правилами.<br>● Python является правильным ответом, так как он позволяет гибко реализовывать различные алгоритмы поиска и имеет богатый набор библиотек для работы с графами и данными.<br>● Системы управления базами данных являются правильным ответом, так как они позволяют хранить огромные графы состояний на диске и эффективно обходить их с помощью рекурсивных запросов.<br>● HTML является неправильным ответом, так как это язык разметки веб-страниц и он не предназначен для программирования логики поиска. | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8. | **Тема 2.4** Задача планирования | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов<br>**ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | 2,2 | **8.** Вопрос: Установите последовательность шагов в алгоритме прямого (forward) поиска планирования от начального состояния к целевому.<br>1. Проверить, удовлетворяет ли текущее состояние целевому условию.<br>2. Выбрать применимое действие (оператор), чьи предусловия выполнены в текущем состоянии.<br>3. Применить эффекты выбранного действия к текущему состоянию, получив новое состояние.<br>4. Начать с исходного состояния.<br>5. Добавить новое состояние в пространство поиска.<br>Правильный порядок: 4, 2, 3, 5, 1<br>Пояснение: Алгоритм начинается с исходного состояния (4). В цикле выбирается действие, применимое в текущей ситуации (2). Это действие применяется, изменяя состояние мира (3). Новое состояние фиксируется (5). Затем проверяется, является ли это состояние целевым (1). Процесс повторяется для новых состояний. | **8.** Вопрос: Расположите в логическом порядке этапы разработки программного комплекса для автоматического планирования с использованием современных технологий.<br>1. Реализация алгоритма поиска плана (напр., на Python или Java).<br>2. Формализация домена планирования и конкретной задачи (напр., на языке PDDL).<br>3. Визуализация найденного плана и его верификация.<br>4. Выбор или разработка синтаксического анализатора (parser) для входного языка описания домена.<br>5. Интеграция решателя планирования в более крупную информационную систему.<br>Правильный порядок: 2, 4, 1, 3, 5<br>Пояснение: Сначала необходимо формально описать предметную область и задачу (2). Затем создать инструмент, который сможет прочитать это описание (4). После этого реализуется ядро системы — алгоритм, работающий с внутренним представлением домена (1). Далее следует этап визуализации и проверки результатов для пользователя (3). | **Базовый (1-3 минуты)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | Финальный этап — встраивание модуля планирования в реальную систему (5). | |
| 9. | **Тема 2.5** Планирование и осуществление действий в реальном мире | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов **ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | 5,5 | **9.** Как называется свойство реальной среды, которое описывает, могут ли действия агента изменять саму среду и каким образом? Это свойство требует от системы планирования постоянно обновлять свою внутреннюю модель мира.<br><br>Ответ: Динамичность<br><br>Пояснение: Динамичность — это ключевое свойство реальных сред, отличающее их от статических моделей. Оно означает, что состояние мира меняется со временем независимо от действий агента | **9.** В архитектуре автономных систем, работающих в реальном мире, как называется программный компонент, который напрямую взаимодействует с физическими датчиками, преобразуя сырые сигналы в структурированную информацию для модуля планирования?<br><br>Правильный ответ: Модуль восприятия<br><br>Пояснение: Модуль восприятия (Perception Module) — это критически важный компонент, который служит "органами чувств" системы. Он отвечает за фильтрацию шума, распознавание | **Высокий (5-10 минут)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | (например, погодные условия, движение других людей). Это напрямую влияет на архитектуру системы, требуя цикла "наблюдение-планирование-действие" и механизмов перепланирования. | объектов, построение карты и в конечном итоге — за создание непротиворечивой и актуальной модели мира, на основе которой уже работает планировщик. Без него планировщик будет работать с устаревшими или неверными данными | |
| 10. | **Тема 2.6** Интеллектуализация процедур поиска и управления | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов **ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | 5,5 | **10.** Вопрос: Как называется свойство реальной среды, которое описывает, могут ли действия агента изменять саму среду и каким образом? Это свойство требует от системы планирования постоянно обновлять свою внутреннюю модель мира. Ответ состоит из одного слова.<br>Правильный ответ: Динамичность<br>Пояснение: Динамичность — это ключевое свойство реальных сред, отличающее их от статических моделей. Оно означает, что состояние мира меняется со временем независимо от действий агента (например, погодные условия, движение других людей). Это напрямую влияет на архитектуру системы, требуя цикла "наблюдение-планирование-действие" и механизмов перепланирования. | **10.** Вопрос: В архитектуре автономных систем, работающих в реальном мире, как называется программный компонент, который напрямую взаимодействует с физическими датчиками, преобразуя сырые сигналы в структурированную информацию для модуля планирования? Ответ состоит из двух слов.<br>Правильный ответ: Модуль восприятия<br>Пояснение: Модуль восприятия (Perception Module) — это критически важный компонент, который служит "органами чувств" системы. Он отвечает за фильтрацию шума, распознавание объектов, построение карты и в конечном итоге — за создание непротиворечивой и актуальной модели мира, на основе которой уже работает планировщик. Без него планировщик будет работать с устаревшими или неверными данными. | **Высокий (5-10 минут)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 11. | **Тема 2.7** Рассуждения в условиях неопределенности | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов<br>**ИПК-7.2** Уметь применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | 4,4 | **11.** Вопрос: Какая модель обобщает марковские процессы принятия решений (MDP) на случай, когда агент не может напрямую наблюдать истинное состояние среды, а вынужден поддерживать вероятностное убеждение (belief state) о нем?<br>1. Скрытая марковская модель (Hidden Markov Model, HMM)<br>2. Частично наблюдаемый марковский процесс принятия решений (Partially Observable MDP, POMDP)<br>3. Детерминированный граф состояний<br>4. Байесовская сеть статического вывода<br>Правильный ответ: 2<br>Пояснение:<br>● Частично наблюдаемый марковский процесс принятия решений (POMDP) является правильным ответом, так как это прямая и наиболее общая модель для последовательного принятия решений в условиях как стохастичности, так и неполной наблюдаемости. Belief state — это ключевое | **11.** Вопрос: Какая из перечисленных российских систем управления базами данных, являющаяся развитием PostgreSQL, может эффективно использоваться для хранения и обработки больших объемов статистических данных, необходимых для обучения и работы вероятностных моделей в условиях неопределенности?<br>1. SQLite<br>2. MySQL<br>3. Postgres Pro<br>4. Oracle Database<br>Правильный ответ: 3<br>Пояснение:<br>● Postgres Pro является правильным ответом, так как это российская СУБД, основанная на мощной и надежной PostgreSQL, которая поддерживает сложные типы данных, оконные функции и расширения для статистического анализа, что делает ее подходящей для работы с данными в условиях неопределенности.<br>● SQLite является неправильным ответом, так как это встраиваемая БД, не предназначенная для анализа больших данных. | **Базовый (1-3 минуты)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | понятие в POMDP.<br>● Скрытая марковская модель (HMM) является неправильным ответом, так как это модель для *прогнозирования* и *анализа* последовательностей, но не для *активного принятия решений*.<br>● Детерминированный граф состояний является неправильным ответом, так как он не учитывает ни стохастичность, ни неполную наблюдаемость.<br>● Байесовская сеть статического вывода является неправильным ответом, так как она обычно используется для разовых выводов, а не для последовательного принятия решений во времени. | ● MySQL и Oracle Database являются неправильными ответами, так как это продукты американских компаний (Oracle Corporation), а не отечественные разработки. | |
| 12. | **Тема 3.1** Стандартизация в области интеллектуального анализа данных | **ПК-7** Способен осуществлять проектирование программных интерфейсов | **ИПК-7.1** Знать методы и средства проектирования программных интерфейсов<br>**ИПК-7.2** Уметь применять методы и средства | | **12.** Вопрос: Установите правильную последовательность фаз согласно методологии CRISP-DM.<br>1. Моделирование (Modeling): Выбор и применение различных алгоритмов моделирования и | **12.** Вопрос: Расположите в логическом порядке этапы развертывания отечественной платформы для анализа данных в защищенном контуре.<br>1. Разработка ETL-процедур и скриптов анализа на Python.<br>2. Установка и настройка | **Базовый (1-3 минуты)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов | | калибровка их параметров.<br>2. Понимание данных (Data Understanding): Сбор первичных данных и их первоначальное исследование.<br>3. Подготовка данных (Data Preparation): Формирование конечного набора данных для моделирования (очистка, преобразование, агрегация).<br>4. Постановка бизнес-задачи (Business Understanding): Определение целей и требований проекта с точки зрения бизнеса.<br>5. Внедрение (Deployment): Развертывание модели в производственной среде.<br>7. Оценка (Evaluation): Проверка того, что модель достигла целей проекта, и утверждение следующих шагов.<br>Правильный порядок: 4, 2, 3, 1, 6, 5<br>Пояснение: CRISP-DM начинается с понимания бизнес-целей (4). Затем следует сбор и изучение данных (2). После этого данные тщательно подготавливаются (3). Только на | российской ОС (например, Astra Linux) на сервере.<br>3. Наполнение базы тестовыми данными и проведение приемочных испытаний.<br>4. Установка и настройка российской СУБД (например, Postgres Pro)<br>.5. Проектирование архитектуры решения и схемы базы данных.<br>Правильный порядок: 5, 2, 4, 1, 3<br>Пояснение: Сначала проектируется архитектура (5). Затем подготавливается инфраструктура — устанавливается ОС (2). После этого развертывается система хранения и обработки данных — СУБД (4). Далее реализуется бизнес-логика анализа (1). Финальный этап — тестирование всего комплекса (3). | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | следующем этапе выбираются и настраиваются модели (1). Построенные модели всесторонне оцениваются (6), и если они успешны, внедряются в эксплуатацию (5). | | |

***Критерии оценивания диагностической работы:***

| Оценка | Критерии оценивания |
|---|---|
| **отлично** | от 90 до 100 % правильно выполненных заданий |
| **хорошо** | от 70 до 89 % правильно выполненных заданий |
| **удовлетворительно** | от 50 до 69 % правильно выполненных заданий |
| **неудовлетворительно** | менее 50 % правильно выполненных заданий |