

РАССМОТРЕНО

на заседании методической комиссии
гуманитарных и программно-
вычислительных дисциплин

Председатель МК Строде Т.Н.
Протокол № 10 от « 14 » 05 2025

УТВЕРЖДАЮ

Зам.директора по учебной работе

Овчинникова И.А.
« 14 » 05 2025 г.

СОГЛАСОВАНО

Начальник отдела эксплуатации и
внедрения информационных систем
СОГАУ «Центр информационных
технологий »

Я. А. Комиссаров
« 14 » 05 2025 г.

КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ

для промежуточной аттестации

по МДК 01.01. Разработка программных модулей в системах

искусственного интеллекта

для специальности

09.02.13 Интеграция решений с применением технологий искусственного интеллекта

Составитель: Котяткина А.Н. – преподаватель СКТ(ф)СПбГУТ

Дифференцированный зачет является промежуточной формой контроля, подводит итог освоения МДК 01.01. Разработка программных модулей в системах искусственного интеллекта в 4 семестре.

К сдаче зачета допускаются студенты, успешно выполнившие все практические и лабораторные работы.

В результате освоения МДК студент должен освоить следующие профессиональные компетенции:

- ВД 1. Разработка кода для обучения искусственного интеллекта.
- ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
- ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.
- ПК 1.3. Оформлять программный код в соответствии с техническим заданием.
- ПК 1.4. Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки.

В ходе проведения дифференцированного зачета проверяется сформированность следующих навыков:

- разработка, оптимизация и тестирование алгоритмов для ИИ-программ;
- использование библиотеки инструментов для работы с алгоритмами и данными

(Pandas, NumPy, Scikit-learn.);

- применение структур данных (деревья, графы, списки для реализации алгоритмов);
- разработка модульных ИИ-систем, соответствующих требованиям производительности и безопасности;
- внедрение разработанных ИИ-модулей в комплексные программные системы

умений:

- анализ технических заданий и выявление требования к алгоритмам;
- применение методов алгоритмизации для решения задач программирования;
- разработка оптимальных алгоритмов для решения задач в области ИИ;

знаний:

- основных методов и подходов к построению алгоритмов (жадные алгоритмы, динамическое программирование, рекурсивные подходы);
- принципов эффективной обработки данных;
- языков программирования, применяемых для разработки алгоритмов (Python, C#, Java.);
- принципов модульного программирования.

Дифференцированный зачет проводится в форме тестирования. Тест содержит 25 из 120 вопросов (15 заданий закрытого типа и 10 заданий открытого типа с кратким ответом), выбираемых случайным образом программой из блоков заданий.

Время тестирования – 45 минут (по одной минуте на каждый вопрос с выбором ответа и по три минуты на вопрос с кратким ответом).

Критерии оценивания:

- «5» - соответствует работа, содержащая 100-90% правильных ответов;
- «4» - соответствует работа, содержащая 76-89% правильных ответов;
- «3» - соответствует работа, содержащая 60-75% правильных ответов;
- «2» - соответствует работа, содержащая менее 60% правильных ответов.

Шкала оценивания образовательных результатов:

Оценка	Критерии
5 «отлично»	Студент набрал 5 баллов согласно критериям
4 «хорошо»	Студент набрал 4 балла согласно критериям
3 «удовлетворительно»	Студент набрал 3 балла согласно критериям
2 «неудовлетворительно»	Студент набрал 2 балла согласно критериям

Первый блок заданий

Проверяемая компетенция - ПК 1.1.

1) Прочитайте текст и выберите один правильный ответ.

Что является первым шагом при формировании алгоритма разработки программного модуля по техническому заданию?

1. Реализация кода
2. Анализ требований и постановка задач
3. Тестирование модуля
4. Внедрение модуля в систему

2) Прочитайте текст и выберите один правильный ответ.

Какой документ обычно содержит описание требований к разрабатываемому программному модулю?

1. Техническое задание (ТЗ)
2. Пользовательская документация
3. Отчет о тестировании
4. Руководство пользователя

3) Прочитайте текст и выберите один правильный ответ.

В процессе формирования алгоритма разработки, что необходимо сделать после анализа требований?

1. Написать код модуля
2. Спроектировать архитектуру и определить основные компоненты
3. Провести тестирование системы
4. Выполнить деплоймент системы

4) Прочитайте текст и выберите один правильный ответ.

Почему важно разбивать техническое задание на этапы при формировании алгоритма разработки?

1. Для упрощения планирования и контроля выполнения работ по каждому этапу
2. Чтобы уменьшить объем работы для разработчика
3. Для исключения необходимости тестирования на каждом этапе
4. Чтобы ускорить процесс внедрения без предварительного анализа

5) Прочитайте текст и выберите один правильный ответ.

Какой из методов помогает определить последовательность выполнения задач при разработке модуля?

1. Метод критического пути (critical PathMethod, CPM)
2. Метод случайного выбора задач
3. Метод "горячих точек" (Hotspots method)
4. Метод "всесразу" (all-at-once approach)

6) Прочитайте текст и выберите один правильный ответ.

Что включает этап проектирования программного модуля?

1. Определение структуры, интерфейсов, алгоритмов и данных, необходимых для реализации требований ТЗ
2. Только написание исходного кода без предварительной подготовки документации
3. Проведение пользовательского обучения по использованию модуля
4. Установка программного обеспечения на сервере клиента

7) Прочитайте текст и выберите один правильный ответ.

Для чего важно создавать блок-схемы или диаграммы при формировании алгоритма разработки?

1. Для визуализации логики работы модуля и облегчения понимания процесса разработки
2. Для автоматической генерации кода без участия разработчика
3. Для уменьшения объема документации
4. Для ускорения выполнения программы во время работы системы

8) Прочитайте текст и выберите один правильный ответ.

Какие критерии должны учитываться при выборе методов реализации программного модуля согласно ТЗ?

1. Соответствие требованиям, эффективность, удобство поддержки и расширяемость
2. Минимальный объем кода без учета требований
3. Максимальная сложность реализации для повышения уровня сложности системы
4. Использование только одного языка программирования без учета требований проекта

9) Прочитайте текст и выберите один правильный ответ.

Что такое "модульное проектирование" в контексте формирования алгоритма разработки?

1. Разделение системы на независимые компоненты с четко определенными интерфейсами для облегчения разработки и поддержки
2. Создание одного большого монолитного компонента без разделения функций
3. Проектирование только пользовательского интерфейса системы
4. Разработка системы без учета требований заказчика

10) Прочитайте текст и выберите один правильный ответ.

Какое действие следует выполнить после завершения этапа проектирования перед началом реализации?

1. Подготовка технической документации и утверждение проекта у заказчика
2. Немедленное написание кода без проверки проектных решений
3. Удаление всех комментариев из исходных кодов
4. Проведение финального тестирования всей системы сразу после проектирования

11) Прочитайте текст и выберите один правильный ответ.

Почему важно учитывать ограничения и риски при формировании алгоритма разработки?

1. Для минимизации возможных ошибок и своевременного реагирования на возможные проблемы в процессе реализации
2. Чтобы увеличить сложность проекта и усложнить работу команды
3. Для исключения необходимости тестирования после завершения разработки
4. Чтобы ускорить процесс внедрения без предварительного анализа рисков

12) Прочитайте текст и выберите один правильный ответ.

Какие инструменты помогают автоматизировать процесс формирования алгоритма разработки?

1. Средства моделирования (например: UML-диаграммы, системы управления проектами, например: Jira, средства контроля версий Git)
2. Только ручное написание текста без использования специальных средств
3. Использование только электронных таблиц Excel для планирования
4. Инструменты для редактирования изображений (например, Photoshop)

13) Прочитайте текст и выберите один правильный ответ.

В чем заключается роль тестового сценария в алгоритме разработки программного модуля?

1. Обеспечить проверку соответствия реализованного модуля требованиям ТЗ и выявление ошибок на ранних этапах
2. Автоматически генерировать исходный код модуля
3. Обеспечить только документацию по использованию модуля без проверки его работы
4. Ускорить внедрение модуля в систему без предварительного тестирования

14) Прочитайте текст и выберите один правильный ответ.

Какие виды тестирования следует включать в алгоритм разработки для обеспечения качества программного модуля?

1. Модульное, интеграционное, системное и приемочное тестирование
2. Только автоматическое тестирование без ручных проверок
3. Тестирование только после полного завершения всей системы
4. Исключительно нагрузочное тестирование для оценки производительности

15) Прочитайте текст и выберите один правильный ответ.

Почему важно документировать все этапы формирования алгоритма разработки?

1. Для обеспечения прозрачности процесса, возможности повторения и поддержки проекта в будущем
2. Чтобы скрыть детали реализации от команды разработчиков
3. Для уменьшения объема рабочей документации
4. Чтобы ускорить процесс внедрения без проверки этапов

16) Прочитайте текст и выберите один правильный ответ.

Какое действие рекомендуется выполнить после завершения реализации программного модуля согласно алгоритму?

1. Провести комплексное тестирование и подготовить документацию по эксплуатации и сопровождению
2. Немедленно запускать модуль в продуктивную среду без проверки
3. Удалить все комментарии из исходных кодов для оптимизации файла
4. Обновлять техническое задание без учета результатов тестирования

17) Прочитайте текст и выберите один правильный ответ.

В чем заключается важность обратной связи от пользователей или заказчика при формировании алгоритма разработки?

1. Для корректировки требований, улучшения качества продукта и своевременного устранения недочетов
2. Для исключения необходимости дальнейшей поддержки системы
3. Для автоматической генерации новых требований без участия заказчика
4. Обратная связь не важна — главное выполнить план

18) Прочитайте текст и выберите один правильный ответ.

Какие показатели эффективности следует учитывать при формировании алгоритма разработки программных модулей?

1. Время выполнения работ, качество результата, соответствие требованиям ТЗ, затраты ресурсов
2. Количество строк кода без учета качества
3. Максимальную сложность реализации
4. Минимальное использование комментариев

19) Прочитайте текст и выберите один правильный ответ.

Почему важно проводить ревью или код ревью на этапе формирования алгоритма разработки?

1. Для выявления ошибок, недочетов и повышения качества проектных решений до начала реализации
2. Чтобы ускорить процесс написания кода без проверки его качества
3. Для автоматической генерации документации
4. Ревью не обязательно — достаточно просто написать код

20) Прочитайте текст и выберите один правильный ответ.

Какие критерии должны использоваться при оценке готовности программы к следующему этапу (согласно сформированному алгоритму)?

1. Полное выполнение требований ТЗ, отсутствие критических ошибок, прохождение всех видов тестирования
2. Минимальный объем кода
3. Быстрое выполнение задачи независимо от качества
4. Отсутствие необходимости документировать результаты

21) Прочитайте текст и выберите один правильный ответ.

Что такое "итеративная разработка" в контексте формирования алгоритма создания программных модулей?

1. Постепенное создание и улучшение модуля через повторные циклы планирования, реализации и проверки
2. Разработка всего проекта за один цикл без повторений
3. Создание прототипа с последующим его полным удалением
4. Разработка только одного варианта решения без изменений

22) Прочитайте текст и выберите один правильный ответ.

Какое значение имеет контроль версий при формировании алгоритма разработки?

1. Позволяет отслеживать изменения, управлять версиями кода и обеспечивать совместную работу команды
2. Не важен — можно работать с одним файлом постоянно
3. Используется только для хранения документов, а не исходного кода
4. Предназначен только для автоматического исправления ошибок

23) Прочитайте текст и выберите один правильный ответ.

Почему необходимо учитывать требования безопасности при формировании алгоритма разработки программных модулей?

1. Для защиты данных и предотвращения несанкционированного доступа или повреждения системы
2. Безопасность не важна — главное быстро реализовать функциональность
3. Безопасность можно добавить только после завершения всей работы
4. Требования безопасности касаются только аппаратных средств

24) Прочитайте текст и выберите один правильный ответ.

Какие действия рекомендуется предпринять перед началом реализации каждого нового функционального блока согласно сформированному алгоритму?

1. Проверка соответствия блока требованиям ТЗ и подготовка необходимых ресурсов для его реализации
2. Немедленное написание кода без предварительных проверок
3. Удаление всех комментариев из предыдущего блока
4. Перенос задачи на следующий этап без проверки

25) Прочитайте текст и выберите один правильный ответ.

В чем заключается основная цель формирования четкого алгоритма разработки программных модулей по ТЗ?

1. Обеспечить систематический подход к созданию качественного продукта с учетом всех требований заказчика и минимизацией ошибок на всех этапах работы
2. Ускорить процесс разработки за счет исключения этапов планирования и анализа
3. Максимально сократить затраты времени и ресурсов без учета требований заказчика
4. Создать максимально сложную архитектуру системы для повышения уровня сложности проекта

Проверяемая компетенция - ПК 1.2.

1) Прочитайте текст и выберите один правильный ответ.

Какой основной этап следует выполнить перед началом разработки программного модуля в соответствии с ТЗ?

1. Тестирование модуля
2. Анализ требований и формулировка технического задания
3. Разработка пользовательской документации
4. Внедрение системы

2) Прочитайте текст и выберите один правильный ответ.

Что является ключевым при разработке программного модуля согласно ТЗ?

1. Максимальная сложность кода
2. Соответствие функциональности требованиям заказчика
3. Минимизация затрат времени на тестирование
4. Использование только новых технологий

3) Прочитайте текст и выберите один правильный ответ.

Какой документ служит основой для разработки программного модуля?

1. Руководство пользователя
2. Техническое задание
3. Отчет о тестировании
4. План проекта

4) Прочитайте текст и выберите один правильный ответ.

Почему важно разбивать ТЗ на отдельные задачи при разработке модуля?

1. Для увеличения объема документации
2. Для упрощения реализации и контроля выполнения
3. Для уменьшения времени разработки за счет пропуска этапов
4. Для снижения требований к качеству

5) Прочитайте текст и выберите один правильный ответ.

Какие методы лучше всего использовать для проверки соответствия разработанного модуля ТЗ?

1. Визуальный осмотр кода
2. Юнит-тестирование и ревью кода
3. Обсуждение с заказчиком без тестирования
4. Игнорирование требований ТЗ

6) Прочитайте текст и выберите один правильный ответ.

Что необходимо сделать после завершения разработки программного модуля?

1. Немедленно внедрить его в продуктивную среду без тестирования
2. Провести тестирование и проверить соответствие требованиям ТЗ

3. Удалить исходный код и оставить только документацию
4. Передать модуль заказчику без проверки

7) Прочитайте текст и выберите один правильный ответ.

Какие критерии должны быть учтены при реализации модуля согласно ТЗ?

1. Производительность, надежность, безопасность
2. Минимальный объем кода без учета требований безопасности
3. Максимальная сложность алгоритмов без учета ресурсов системы
4. Использование только открытых библиотек

8) Прочитайте текст и выберите один правильный ответ.

Почему важно вести документацию по разработке программных модулей?

1. Для повышения стоимости проекта
2. Для облегчения поддержки, тестирования и дальнейшей доработки
3. Чтобы скрыть особенности реализации от команды
4. Для уменьшения времени разработки

9) Прочитайте текст и выберите один правильный ответ.

Какое действие помогает обеспечить соответствие модуля требованиям ТЗ во время разработки?

1. Постоянное сравнение реализуемого функционала с требованиями из ТЗ
2. Игнорирование изменений в требованиях после начала разработки
3. Разработка без учета требований заказчика
4. Использование только автоматических средств разработки

10) Прочитайте текст и выберите один правильный ответ.

Что такое "контроль версий" и зачем он нужен при разработке по ТЗ?

1. Метод отслеживания изменений кода для обеспечения соответствия требованиям и возможности возврата к предыдущим версиям
2. Процесс автоматического тестирования модулей
3. Метод ускорения разработки за счет автоматической генерации кода
4. Способ документирования требований заказчика

11) Прочитайте текст и выберите один правильный ответ.

Какие подходы помогают обеспечить качество разрабатываемого программного модуля?

1. Модульное тестирование, ревью кода, интеграционное тестирование
2. Только автоматизированное тестирование без ручных проверок
3. Разработка без проверки требований ТЗ
4. Игнорирование ошибок после их обнаружения

12) Прочитайте текст и выберите один правильный ответ.

Почему важно учитывать ограничения ресурсов (время, память, вычислительная мощность)?

1. Чтобы снизить требования к качеству системы
2. Чтобы обеспечить выполнение проекта в рамках заданных условий и требований ТЗ
3. Чтобы уменьшить объем документации
4. Чтобы ускорить внедрение системы без проверки

13) Прочитайте текст и выберите один правильный ответ.

Что такое "модульная архитектура" и как она связана с выполнением ТЗ?

1. Структура системы, разделенная на независимые компоненты, что облегчает соответствие требованиям и поддержку

2. Архитектура, основанная на едином монолитном блоке без разделения функций
3. Архитектура, которая не учитывает требования безопасности или надежности
4. Структура системы, полностью основанная на внешних библиотеках

14) Прочитайте текст и выберите один правильный ответ.

Какие виды тестирования наиболее важны при разработке по ТЗ?

1. Юнит-тесты, интеграционные тесты, системные тесты
2. Только автоматизированные тесты без ручных проверок
3. Только нагрузочные тесты для оценки производительности
4. Не требуется тестирование при соблюдении сроков

15) Прочитайте текст и выберите один правильный ответ.

Как правильно управлять изменениями в требованиях во время разработки модуля?

1. Вести журнал изменений и согласовывать их с заказчиком
2. Игнорировать изменения после утверждения ТЗ
3. Вносить изменения без согласования для ускорения процесса
4. Объявлять изменения недопустимыми после начала разработки

16) Прочитайте текст и выберите один правильный ответ.

Почему важно проводить ревью кода в процессе разработки по ТЗ?

1. Для выявления несоответствий требованиям и повышения качества продукта
2. Чтобы снизить затраты на проект
3. Для ускорения процесса внедрения
4. Чтобы уменьшить объем документации

17) Прочитайте текст и выберите один правильный ответ.

Какие инструменты помогают автоматизировать контроль выполнения требований из ТЗ?

1. Системы управления проектами (например, Jira, системы отслеживания ошибок и автоматические тесты)
2. Только текстовые редакторы
3. Электронная почта и личные заметки
4. Ручной контроль без использования специальных средств

18) Прочитайте текст и выберите один правильный ответ.

Что такое "интеграционное тестирование" и зачем оно нужно при разработке по ТЗ?

1. Проверка взаимодействия различных модулей системы для обеспечения их совместной работы согласно требованиям
2. Тестирование отдельных функций внутри одного модуля
3. Проверка производительности системы под нагрузкой
4. Тестирование пользовательского интерфейса

19) Прочитайте текст и выберите один правильный ответ.

Как обеспечить безопасность программного модуля в соответствии с требованиями ТЗ?

1. Реализовать механизмы аутентификации, авторизации и защиты данных согласно спецификациям
2. Игнорировать вопросы безопасности для ускорения разработки
3. Использовать только открытые библиотеки без проверки их безопасности
4. Обеспечивать безопасность только после внедрения системы

20) Прочитайте текст и выберите один правильный ответ.

Почему важно проводить регрессионное тестирование после внесения изменений в модуль?

1. Чтобы убедиться, что новые изменения не нарушили работу ранее функционирующих частей системы
2. Чтобы проверить производительность системы под нагрузкой
3. Для оценки удобства использования интерфейса
4. Для поиска новых ошибок в коде

21) Прочитайте текст и выберите один правильный ответ.

Какие показатели эффективности следует учитывать при разработке программных модулей по ТЗ?

1. Соответствие функциональности требованиям, качество кода, производительность, надежность
2. Количество строк кода без учета качества или требований
3. Минимальное время разработки независимо от качества результата
4. Стоимость проекта без учета качества или сроков выполнения

22) Прочитайте текст и выберите один правильный ответ.

Как правильно документировать результаты выполнения требований из ТЗ во время разработки?

1. Создавать отчеты о выполнении задач, вести журнал изменений и писать комментарии в коде
2. Объявлять выполнение требований устно без фиксации документов
3. Игнорировать необходимость документации
4. Документировать только финальную версию продукта

23) Прочитайте текст и выберите один правильный ответ.

В чем заключается роль автоматизированных средств при разработке модулей по ТЗ?

1. Обеспечивают быструю проверку соответствия функционала требованиям и автоматизацию повторяющихся задач
2. Полностью заменяют команду разработчиков
3. Используются только для хранения данных
4. Не имеют существенного значения для соблюдения требований

24) Прочитайте текст и выберите один правильный ответ.

Почему важно соблюдать принципы модульности при разработке систем искусственного интеллекта по ТЗ?

1. Облегчают поддержку, расширение и повторное использование компонентов
2. Усложняют структуру системы для повышения уровня сложности
3. Позволяют объединять все функции в один большой модуль
4. Исключают необходимость проведения тестирования

25) Прочитайте текст и выберите один правильный ответ.

Как обеспечить выполнение всех требований из технического задания при ограниченных сроках проекта?

1. Планировать работу по этапам, использовать итеративную модель разработки и регулярно контролировать прогресс
2. Игнорировать часть требований ради скорости
3. Разрабатывать все сразу без планирования
4. Полностью полагаться на автоматические средства без контроля человека

Проверяемая компетенция - ПК 1.3

- 1) Прочитайте текст и выберите один правильный ответ.

Почему важно соблюдать стандарты оформления кода при разработке модулей ИИ?

1. Для повышения читаемости и поддержки кода командой разработчиков
2. Чтобы уменьшить объем исходного кода
3. Для скрытия сложности алгоритмов
4. Чтобы ускорить выполнение программы

2) Прочитайте текст и выберите один правильный ответ.

Какие элементы должны обязательно присутствовать в комментариях к функции согласно ТЗ?

1. Только название функции
2. Описание назначения функции, входных параметров и возвращаемого значения
3. Личные заметки разработчика о реализации
4. Инструкции по сборке проекта

3) Прочитайте текст и выберите один правильный ответ.

Какое правило оформления имен переменных рекомендуется соблюдать?

1. Использовать короткие имена без смысла для экономии места
2. Использовать осмысленные имена, отражающие содержимое или назначение переменной
3. Использовать только заглавные буквы для всех имен
4. Не использовать комментарии для объяснения имен

4) Прочитайте текст и выберите один правильный ответ.

Что является хорошей практикой при структурировании кода в соответствии с ТЗ?

1. Разделять код на логические блоки, использовать отступы и комментарии для ясности
2. Писать весь код без разделений и отступов для компактности
3. Объединять все функции в один файл без документации
4. Исключать комментарии, чтобы не отвлекать внимание

5) Прочитайте текст и выберите один правильный ответ.

Какие стандарты оформления рекомендуется соблюдать при разработке модулей ИИ?

1. Следовать внутренним стандартам компании или общепринятым стандартам языка программирования (например, PEP8 для Python).
2. Создавать собственные правила оформления без учета стандартов языка или команды
3. Не обращать внимания на оформление, главное — функциональность
4. Использовать минимальное количество комментариев и форматирования

6.) Прочитайте текст и выберите один правильный ответ.

Как правильно оформлять длинные выражения или строки кода?

1. Разбивать их на несколько строк с помощью переносов или специальных символов (например, обратный слеш в Python) для повышения читаемости
2. Оставлять длинные строки без переноса ради компактности
3. Удалять все пробелы внутри длинных выражений для экономии места
4. Объединять все выражения в одну строку без переносов

7) Прочитайте текст и выберите один правильный ответ.

Что необходимо делать при документировании сложных алгоритмов в соответствии с ТЗ?

1. Добавлять подробные комментарии и документацию по этапам реализации и логике алгоритма
2. Оставлять их без комментариев, чтобы не мешать чтению кода
3. Разделять алгоритм на несколько файлов без объяснений внутри кода
4. Исключать документацию, чтобы ускорить разработку

8) Прочитайте текст и выберите один правильный ответ.

Какие элементы должны быть задокументированы в файле исходного кода?

1. Только публичные функции и классы с описанием назначения и параметров
2. Все функции, переменные и блоки кода без исключения
3. Только заголовки файлов без описания содержимого
4. Документацию писать только после завершения всей разработки

9) Прочитайте текст и выберите один правильный ответ.

Почему важно использовать соглашения о стиле именования переменных и функций?

1. Для обеспечения единообразия, читаемости и поддержки кода командой разработки
2. Чтобы сделать код менее понятным другим участникам команды
3. Для скрытия сложности реализации за красивым стилем оформления
4. Чтобы уменьшить количество комментариев внутри кода

10) Прочитайте текст и выберите один правильный ответ.

Какие рекомендации по оформлению функций соответствуют лучшим практикам?

1. Давать функциям осмысленные имена, описывать параметры и возвращаемое значение через комментарии или документацию
2. Назначать короткие имена типа $f(.)$, $g(.)$ без описаний
3. Не писать документацию внутри функций, чтобы не мешать чтению кода
4. Объявлять все функции как глобальные без использования модулей или классов

11) Прочитайте текст и выберите один правильный ответ.

Как оформить условные конструкции для повышения читаемости?

1. Расписывать каждое условие на отдельной строке с отступами для вложенности и использования понятных условий
2. Объединять все условия в одну строку ради краткости
3. Не использовать условные конструкции вообще
4. Писать условия как можно короче без объяснений

12) Прочитайте текст и выберите один правильный ответ.

Что рекомендуется делать при оформлении циклов?

1. Использовать понятные имена переменных счетчиков, соблюдать отступы внутри цикла и добавлять комментарии при необходимости
2. Писать циклы как можно короче, игнорируя читаемость
- 3- Не добавлять комментарии внутри циклов даже при необходимости объяснить логику
- 4- Объединять все циклы в один большой блок без разделения

13) Прочитайте текст и выберите один правильный ответ.

Почему важно соблюдать единообразие оформления кода во всем проекте?

1. Для облегчения поддержки, понимания и развития проекта командой разработчиков
2. Чтобы сделать код менее понятным другим участникам команды
3. Для скрытия недостатков реализации за красивым стилем оформления
4. Чтобы уменьшить количество комментариев в коде

14) Прочитайте текст и выберите один правильный ответ.

Какие элементы оформления помогают снизить вероятность ошибок при чтении кода?

1. Использование четких отступов, единых стилей именования и комментариев по необходимости
2. Игнорирование стандартов оформления ради скорости разработки

3. Удаление всех комментариев после написания кода
4. Писание всего кода одним блоком без разделений

15) Прочитайте текст и выберите один правильный ответ.

Как оформлять длинные функции или модули согласно ТЗ?

1. Разделять их на меньшие логические части с помощью функций или методов, добавляя описание каждого блока
2. Писать все подряд без разделений ради краткости
3. Объединять все части в один большой файл без документации
4. Исключать комментарии внутри функций для экономии места

16) Прочитайте текст и выберите один правильный ответ.

Как правильно оформлять импорт модулей и библиотек в соответствии с ТЗ?

1. Расположить все импорты в начале файла, группируя стандартные, сторонние и локальные библиотеки, с комментариями при необходимости
2. Распределять импорты по всему файлу для удобства чтения
3. Не использовать комментарии или группировки при импортах
4. Импортировать только те модули, которые используются в текущем файле, без комментариев

17) Прочитайте текст и выберите один правильный ответ.

Какие рекомендации по оформлению документации к классам и методам?

1. Использовать стандартизированные форматы (например, docstrings в Python), описывать назначение, параметры и возвращаемое значение
2. Оставлять документацию только для публичных методов, игнорируя приватные
3. Не писать документацию, чтобы не мешать чтению кода
4. Документацию писать только после завершения всей разработки

18) Прочитайте текст и выберите один правильный ответ.

Как следует оформлять обработку ошибок и исключений в соответствии с ТЗ?

1. Использовать блоки try-except с поясняющими комментариями и логированием ошибок для повышения надежности кода
2. Игнорировать обработку ошибок для ускорения разработки
3. Обрабатывать ошибки только в самом конце проекта после тестирования
4. Не использовать обработку исключений, чтобы не усложнять код

19) Прочитайте текст и выберите один правильный ответ.

Какие правила оформления переменных и функций помогают обеспечить поддержку кода?

1. Использовать понятные имена, придерживаться единого стиля именования и структурировать код логически
2. Назначать переменным короткие имена без смысла для экономии места
3. Использовать случайные имена без описания назначения
4. Не писать комментарии или документацию внутри кода

20) Прочитайте текст и выберите один правильный ответ.

Как правильно оформлять тестовые сценарии и вспомогательный код в соответствии с ТЗ?

1. Размещать их в отдельных файлах или разделах с ясной маркировкой и комментариями для понимания их назначения
2. Вписывать тестовые сценарии прямо внутри основной логики программы без разделения

3. Удалять все тестовые сценарии после завершения разработки
4. Не документировать тестовые сценарии вовсе

21) Прочитайте текст и выберите один правильный ответ.

Почему важно соблюдать единый стиль оформления кода при разработке модулей ИИ?

1. Для облегчения командной работы, поддержки и дальнейшего развития проекта
2. Чтобы усложнить понимание кода другими разработчиками
3. Для скрытия недостатков реализации за красивым стилем оформления
4. Чтобы уменьшить количество комментариев внутри кода

22) Прочитайте текст и выберите один правильный ответ.

Какие элементы оформления помогают сделать код более понятным для других участников проекта?

1. Использование осмысленных имен переменных, структурирование кода, добавление комментариев и документации
2. Писать весь код одним блоком без разделений или комментариев
3. Использовать минимальное количество отступов и форматирования ради краткости
4. Исключать все комментарии после написания кода

23) Прочитайте текст и выберите один правильный ответ.

Как следует оформлять сложные условия или вложенные конструкции?

1. Расписывать каждое условие на отдельной строке с отступами для повышения читаемости и добавлять комментарии при необходимости
2. Объединять все условия в одну строку ради краткости
3. Избегать использования вложенных условий полностью
4. Писать условия как можно короче без объяснений

24) Прочитайте текст и выберите один правильный ответ.

Что рекомендуется делать после завершения написания модуля перед сдачей его на проверку?

1. Провести ревизию оформления кода, проверить наличие комментариев, соответствие стандартам и выполнить тестирование
2. Удалить все комментарии для уменьшения объема файла
3. Оставить код без изменений до следующего релиза без проверки оформления
4. Переписать весь код заново без учета предыдущих правил оформления

25) Прочитайте текст и выберите один правильный ответ.

Как правильно оформлять документацию к сложным алгоритмам в соответствии с ТЗ?

1. Добавлять подробные комментарии внутри кода, описывающие логику и этапы выполнения алгоритма, а также создавать внешнюю документацию при необходимости
2. Оставлять все комментарии только в конце разработки, после тестирования
3. Не писать комментарии, чтобы не мешать чтению кода
4. Использовать только короткие однострочные комментарии без объяснений

Проверяемая компетенция - ПК 1.4.

1) Прочитайте текст и выберите один правильный ответ.

Какова основная цель использования системы контроля версий в командной разработке?

1. Обеспечить возможность отслеживания изменений и совместной работы над кодом
2. Ускорить выполнение программы во время работы
3. Уменьшить объем исходного кода
4. Автоматически исправлять ошибки в коде

2) Прочитайте текст и выберите один правильный ответ.

Какой из следующих инструментов является системой контроля версий?

1. Git
2. Visual Studio
3. Docker
4. Jenkins

3) Прочитайте текст и выберите один правильный ответ.

Что такое "ветвление" (branching) в системе контроля версий?

1. Создание отдельной копии проекта для разработки новых функций или исправлений без влияния на основную ветку
2. Объединение нескольких проектов в один репозиторий
3. Удаление старых версий кода
4. Автоматическая компиляция проекта

4) Прочитайте текст и выберите один правильный ответ.

Почему важно использовать систему контроля версий при групповой разработке?

1. Для координации изменений, предотвращения конфликтов и сохранения истории изменений
2. Чтобы автоматически писать документацию к проекту
3. Для ускорения выполнения программы во время работы
4. Чтобы исключить необходимость тестирования

5) Прочитайте текст и выберите один правильный ответ.

Что такое "слияние" merge в системе контроля версий?

1. Объединение изменений из одной ветки в другую, обычно из ветки разработки в основную ветку
2. Создание новой ветки для экспериментов
3. Удаление старых версий кода
4. Разделение проекта на части

6) Прочитайте текст и выберите один правильный ответ.

Какие преимущества дает использование системы контроля версий для групповой разработки?

1. Совместное редактирование, отслеживание истории изменений, разрешение конфликтов и восстановление предыдущих версий
2. Автоматическую генерацию тестов для кода
3. Ускорение выполнения программы за счет оптимизации кода
4. Исключение необходимости писать документацию

7) Прочитайте текст и выберите один правильный ответ.

Что означает термин "конфликт" при работе с системой контроля версий?

1. Ситуация, когда два разработчика внесли несовместимые изменения в один и тот же участок кода одновременно, и их нужно разрешить вручную
2. Ошибка компиляции программы при сборке проекта
3. Вирус, заражающий репозиторий проекта
4. Процесс автоматического объединения изменений без вмешательства пользователя

8) Прочитайте текст и выберите один правильный ответ.

Как можно предотвратить конфликты при работе с системой контроля версий?

1. Регулярно обновлять локальную копию репозитория и своевременно сливать изменения из основной ветки

2. Не использовать ветвления и работать только в главной ветке
3. Не коммитить изменения до завершения всей работы
4. Отключать автоматические проверки конфликтов

9) Прочитайте текст и выберите один правильный ответ.

Что такое "коммит"?

1. Зафиксировать изменения в локальном репозитории с описанием выполненных работ
2. Объединить две ветки проекта
3. Удалить старую версию файла
4. Запустить автоматический тест проекта

10) Прочитайте текст и выберите один правильный ответ.

Почему важно писать осмысленные сообщения при коммите?

1. Для ясности истории изменений и облегчения поиска нужных исправлений или функций
2. Чтобы увеличить размер файла репозитория
3. Для автоматической генерации документации
4. Чтобы скрыть детали внесенных изменений

11) Прочитайте текст и выберите один правильный ответ.

Что такое "репозиторий"?

1. Хранилище всех файлов проекта и истории изменений
2. Инструмент для автоматической сборки проекта
3. Средство для тестирования программного обеспечения
4. Виртуальная машина для запуска приложений

12) Прочитайте текст и выберите один правильный ответ.

Какие виды ветвлений обычно используются в командной разработке?

1. Основная (main/master), разработка (develop), функциональные ветки (feature), исправления ошибок (hotfix/bugfix), релизы (release)
2. Только одна главная ветка без дополнительных вариантов
3. Ветки создаются только для экспериментов без последующего объединения
4. Ветки создаются автоматически без участия разработчика

13) Прочитайте текст и выберите один правильный ответ.

Какой из следующих процессов обеспечивает интеграцию изменений из разных веток?

1. Слияние (merge или rebase)
2. Клонирование (clone)
3. Выгрузка (push)
4. Проверка на вирусы

14) Прочитайте текст и выберите один правильный ответ.

Что такое "pullrequest" или "запрос на слияние"?

1. Предложение внести изменения из одной ветки в другую с целью их проверки и одобрения командой перед объединением
2. Автоматическая команда по удалению старых веток
3. Запрос на получение доступа к репозиторию
4. Процесс автоматической сборки проекта

15) Прочитайте текст и выберите один правильный ответ.

Почему рекомендуется регулярно делать коммиты при групповой разработке?

1. Для минимизации конфликтов и сохранения актуальной истории изменений

2. Чтобы увеличить размер репозитория
3. Для автоматического тестирования всего проекта
4. Чтобы скрыть ошибки от команды

16) Прочитайте текст и выберите один правильный ответ.

Какие инструменты позволяют управлять несколькими репозиториями одновременно?

1. Git submodules или субмодули
2. docker
3. Jenkins
4. VisualStudioCode

17) Прочитайте текст и выберите один правильный ответ.

Что такое "ветка разработки"?

1. Ветка, где ведется активная работа над новыми функциями перед их интеграцией в основную ветку
2. Основная стабильная версия продукта
3. Ветка только для хранения архивных данных
4. Ветка, которая автоматически удаляется после завершения работы

18) Прочитайте текст и выберите один правильный ответ.

Как обеспечить контроль доступа к репозиторию в командной работе?

1. Использовать системы аутентификации и разграничения прав доступа, например, через платформы GitHub или GitLab
2. Не ограничивать доступ никому
3. Делать все изменения только локально без синхронизации
4. Использовать только одну учетную запись для всех участников

19) Прочитайте текст и выберите один правильный ответ.

Какие практики помогают снизить риск потери данных при использовании системы контроля версий?

1. Регулярные коммиты, создание резервных копий репозитория и использование облачных сервисов
2. Работа только локально без синхронизации
3. Отключение системы контроля версий во время работы
4. Удаление старых веток сразу после создания

20) Прочитайте текст и выберите один правильный ответ.

Чем отличается команда "push" от команды "pull"?

1. Push — отправка локальных изменений на сервер; pull — получение последних изменений с сервера
2. Push — получение обновлений; pull — отправка своих изменений
3. Они выполняют одинаковые функции
4. Push используется только для создания новых репозиториев

21) Прочитайте текст и выберите один правильный ответ.

Почему важно регулярно обновлять локальную копию репозитория перед началом работы?

1. Чтобы иметь актуальную версию кода и избегать конфликтов при слиянии
2. Чтобы увеличить размер локального репозитория
3. Для автоматической генерации документации
4. Это не важно

22) Прочитайте текст и выберите один правильный ответ.

Что такое "redase" и чем он отличается от "merge"?

- 1.Redase переносит ваши изменения поверх последних коммитов основной ветки; merge объединяет две ветки без переписывания истории.
- 2.Redase удаляет все старые коммиты; merge их сохраняет.
- 3.Redase используется только для удаления конфликтов; merge — для их разрешения.
4. Они выполняют одинаковую функцию

23) Прочитайте текст и выберите один правильный ответ.

Какие преимущества дает использование систем контроля версий при работе над проектом ИИ?

1. Отслеживание прогресса, совместное редактирование, управление версиями моделей и данных, предотвращение потери информации
2. Автоматическое обучение моделей ИИ
3. Ускорение вычислений
4. Исключает необходимость тестирования

24) Прочитайте текст и выберите один правильный ответ.

Как правильно организовать работу над несколькими функциями или модулями в рамках одного репозитория?

1. Создавать отдельные ветки для каждой функции или модуля, затем объединять их после завершения работы
2. Работать все сразу в главной ветке
3. Создавать отдельные репозитории для каждого модуля
4. Не использовать систему контроля версий

25) Прочитайте текст и выберите один правильный ответ.

Какие действия необходимо выполнить после завершения работы над новой функцией перед её объединением с основной веткой?

1. Провести ревью кода, протестировать изменения, решить возможные конфликты и сделать запрос на слияние.
2. Немедленно слить изменения без проверки.
3. Удалить все временные файлы.
4. Откатить все изменения.

Второй блок заданий

Проверяемая компетенция - ПК 1.1.

1. Прочитайте текст. Дайте краткий ответ.
Опишите основные этапы формирования алгоритма разработки программного модуля в соответствии с техническим заданием.
2. Прочитайте текст. Дайте краткий ответ.
Какие ключевые требования должны быть учтены при анализе технического задания для разработки программного модуля?
3. Прочитайте текст. Дайте краткий ответ.
Объясните роль предварительного анализа требований в процессе формирования алгоритма разработки.
4. Прочитайте текст. Дайте краткий ответ.
Какие методы и инструменты можно использовать для моделирования логики работы программного модуля?
5. Прочитайте текст. Дайте краткий ответ.

- Как определить последовательность выполнения задач при разработке программного модуля согласно ТЗ?
6. Прочитайте текст. Дайте краткий ответ.
В чем заключается важность разбивки технического задания на подзадачи и этапы при формировании алгоритма?
 7. Прочитайте текст. Дайте краткий ответ.
Опишите процесс проектирования архитектуры программного модуля на основе требований ТЗ.
 8. Прочитайте текст. Дайте краткий ответ.
Какие критерии следует учитывать при выборе методов реализации программного модуля?
 9. Прочитайте текст. Дайте краткий ответ.
Объясните, как можно использовать диаграммы и блок-схемы для визуализации алгоритма разработки.
 10. Прочитайте текст. Дайте краткий ответ.
Какие риски могут возникнуть при неправильном формировании алгоритма разработки, и как их избежать?
 11. Прочитайте текст. Дайте краткий ответ.
Как обеспечить соответствие разрабатываемого модуля требованиям безопасности и надежности согласно ТЗ?
 12. Прочитайте текст. Дайте краткий ответ.
В чем заключается роль тестирования в процессе формирования алгоритма разработки программных модулей?
 13. Какие виды тестирования необходимо предусмотреть при разработке системы в соответствии с ТЗ?
 14. Прочитайте текст. Дайте краткий ответ.
Объясните, как документация помогает в процессе формирования и реализации алгоритма разработки.
 15. Прочитайте текст. Дайте краткий ответ.
Какие подходы к управлению проектом позволяют эффективно реализовать алгоритм разработки программных модулей?
 16. Прочитайте текст. Дайте краткий ответ.
Как учитывать ограничения ресурсов (время, бюджет, оборудование) при формировании алгоритма?
 17. В чем заключается значение обратной связи от заказчика или пользователя на этапах формирования алгоритма?
 18. Прочитайте текст. Дайте краткий ответ.
Опишите основные критерии оценки готовности программного модуля к внедрению после его разработки.
 19. Как можно автоматизировать процесс формирования и проверки алгоритмов разработки программных модулей?
 20. Прочитайте текст. Дайте краткий ответ.
Объясните значение итеративного подхода в формировании алгоритмов разработки систем искусственного интеллекта.
 21. Прочитайте текст. Дайте краткий ответ.
Какие особенности необходимо учитывать при разработке модулей для систем искусственного интеллекта по ТЗ?
 22. Прочитайте текст. Дайте краткий ответ.
Как обеспечить масштабируемость и расширяемость программных модулей при их проектировании согласно ТЗ?
 23. Прочитайте текст. Дайте краткий ответ.

В чем заключается роль ревью и код-ревью в процессе формирования алгоритмов разработки?

24. Прочитайте текст. Дайте краткий ответ.

Какие показатели эффективности следует учитывать при планировании этапов разработки по ТЗ?

25. Прочитайте текст. Дайте краткий ответ.

Опишите основные шаги по подготовке к реализации каждого функционального блока системы согласно сформированному алгоритму.

Проверяемая компетенция - ПК 1.2.

1. Прочитайте текст. Дайте краткий ответ.

Какие основные этапы необходимо пройти при разработке программного модуля в соответствии с техническим заданием?

2. Прочитайте текст. Дайте краткий ответ.

Как правильно интерпретировать техническое задание для разработки программного модуля в системах искусственного интеллекта?

3. Прочитайте текст. Дайте краткий ответ.

Какие ключевые параметры должны быть учтены при проектировании модуля согласно ТЗ?

4. Прочитайте текст. Дайте краткий ответ.

Как обеспечить соответствие реализуемого модуля требованиям, указанным в техническом задании?

5. Прочитайте текст. Дайте краткий ответ.

Какие методы проверки соответствия разработанного модуля техническому заданию вы знаете?

6. Прочитайте текст. Дайте краткий ответ.

Как документировать процесс разработки программного модуля для последующей проверки его соответствия ТЗ?

7. Прочитайте текст. Дайте краткий ответ.

Какие сложности могут возникнуть при реализации модуля по техническому заданию и как их преодолеть?

8. Прочитайте текст. Дайте краткий ответ.

Как организовать работу команды разработчиков для соблюдения требований ТЗ при создании модуля?

9. Прочитайте текст. Дайте краткий ответ.

Какие инструменты и средства помогают автоматизировать разработку и тестирование модулей согласно ТЗ?

10. Прочитайте текст. Дайте краткий ответ.

Как учитывать возможные изменения в техническом задании во время разработки программного модуля?

11. Прочитайте текст. Дайте краткий ответ.

Какие критерии оценки соответствия программного модуля требованиям ТЗ вы можете назвать?

12. Прочитайте текст. Дайте краткий ответ.

Как правильно разбивать техническое задание на задачи для разработки отдельных компонентов модуля?

13. Прочитайте текст. Дайте краткий ответ.

Какие особенности разработки модулей для систем искусственного интеллекта требуют особого внимания при соблюдении ТЗ?

14. Прочитайте текст. Дайте краткий ответ.

Как обеспечить масштабируемость и расширяемость программных модулей в рамках требований ТЗ?

15. Прочитайте текст. Дайте краткий ответ.
Какие подходы к тестированию позволяют убедиться, что модуль соответствует техническому заданию?
16. Прочитайте текст. Дайте краткий ответ.
Как взаимодействовать с заказчиком или аналитиком для уточнения требований из ТЗ при разработке модуля?
17. Прочитайте текст. Дайте краткий ответ.
Какие ошибки чаще всего допускают при реализации модулей по техническому заданию и как их избежать?
18. Прочитайте текст. Дайте краткий ответ.
Как использовать моделирование и прототипирование для проверки соответствия будущего модуля ТЗ?
19. Прочитайте текст. Дайте краткий ответ.
В чем заключается роль документации при разработке программных модулей согласно ТЗ?
20. Прочитайте текст. Дайте краткий ответ.
Какие особенности учета требований к безопасности и надежности при разработке модулей по ТЗ в системах ИИ?
21. Прочитайте текст. Дайте краткий ответ.
Как обеспечить совместимость разрабатываемого модуля с другими компонентами системы согласно ТЗ?
22. Прочитайте текст. Дайте краткий ответ.
Какие методы управления изменениями в техническом задании применимы при разработке модулей ИИ?
23. Прочитайте текст. Дайте краткий ответ.
Как оценить трудозатраты и сроки выполнения разработки, исходя из требований ТЗ?
24. Прочитайте текст. Дайте краткий ответ.
В чем заключается важность обратной связи от тестировщиков и заказчика для корректировки разработки по ТЗ?
25. Прочитайте текст. Дайте краткий ответ.
Какие лучшие практики существуют для обеспечения высокого качества программных модулей, разрабатываемых по техническому заданию?

Проверяемая компетенция - ПК 1.3.

1. Прочитайте текст. Дайте краткий ответ.
Какие основные требования предъявляются к оформлению программного кода в соответствии с техническим заданием?
2. Прочитайте текст. Дайте краткий ответ.
Почему важно соблюдать стандарты оформления кода при разработке модулей для систем искусственного интеллекта?
3. Прочитайте текст. Дайте краткий ответ.
Опишите основные элементы, которые должны присутствовать в комментариях к коду согласно техническому заданию.
4. Прочитайте текст. Дайте краткий ответ.
Какие правила именования переменных и функций рекомендуется использовать для повышения читаемости кода?
5. Прочитайте текст. Дайте краткий ответ.
Как оформление кода влияет на его поддержку и дальнейшее развитие?
6. Прочитайте текст. Дайте краткий ответ.
Какие стандарты оформления кода существуют в практике разработки программных модулей, и как их применять?
7. Прочитайте текст. Дайте краткий ответ.

Объясните роль документации и комментариев в соответствии с требованиями технического задания.

8. Прочитайте текст. Дайте краткий ответ.
Какие особенности оформления кода следует учитывать при работе с алгоритмами машинного обучения или нейронными сетями?
9. Прочитайте текст. Дайте краткий ответ.
Как правильно структурировать код, чтобы он соответствовал требованиям технического задания?
10. Прочитайте текст. Дайте краткий ответ.
Какие методы проверки соответствия оформленного кода требованиям документации и стандартам?
11. Прочитайте текст. Дайте краткий ответ.
В чем заключается важность использования отступов и форматирования при оформлении программного модуля?
12. Прочитайте текст. Дайте краткий ответ.
Как оформить сложные функции или классы для обеспечения их понятности и соответствия стандартам?
13. Прочитайте текст. Дайте краткий ответ.
Какие рекомендации по оформлению тестовых сценариев и вспомогательных скриптов, связанных с основным модулем?
14. Прочитайте текст. Дайте краткий ответ.
Объясните, каким образом оформление кода влияет на его тестируемость и отладку.
15. Прочитайте текст. Дайте краткий ответ.
Какие типичные ошибки в оформлении кода встречаются при разработке модулей для систем искусственного интеллекта?
16. Прочитайте текст. Дайте краткий ответ.
Как оформить обработку ошибок и исключений в соответствии с требованиями технического задания?
17. Прочитайте текст. Дайте краткий ответ.
Какие требования предъявляются к оформлению конфигурационных файлов или настроек, связанных с программным модулем?
18. Прочитайте текст. Дайте краткий ответ.
Как обеспечить единообразие оформления кода при работе в команде согласно стандартам проекта?
19. Прочитайте текст. Дайте краткий ответ.
В чем заключается важность соблюдения правил оформления при интеграции модуля в более крупную систему?
20. Прочитайте текст. Дайте краткий ответ.
Какие инструменты автоматической проверки качества оформления кода существуют, и как их использовать?
21. Прочитайте текст. Дайте краткий ответ.
Объясните, как оформление комментариев помогает другим разработчикам понять логику работы модуля.
22. Прочитайте текст. Дайте краткий ответ.
Какие особенности оформления кода необходимо учитывать при использовании специфических библиотек или фреймворков для ИИ?
23. Прочитайте текст. Дайте краткий ответ.
Как оформить документацию по использованию программного модуля согласно требованиям технического задания?
24. Прочитайте текст. Дайте краткий ответ.
Почему важно соблюдать единый стиль оформления кода во всем проекте, и как это реализовать?

25. Прочитайте текст. Дайте краткий ответ.
Какие шаги необходимо предпринять для приведения существующего кода в соответствие с установленными стандартами оформления согласно ТЗ?

Проверяемая компетенция - ПК 1.4.

1. Прочитайте текст. Дайте краткий ответ.
Почему важно использовать системы контроля версий при разработке программных модулей в системах искусственного интеллекта?
2. Прочитайте текст. Дайте краткий ответ.
Какие основные преимущества дает использование системы контроля версий для групповой разработки?
3. Прочитайте текст. Дайте краткий ответ.
Опишите основные функции системы контроля версий, которые способствуют организации командной работы.
4. Прочитайте текст. Дайте краткий ответ.
Какие требования к организации групповой разработки следует учитывать при выборе системы контроля версий?
5. Прочитайте текст. Дайте краткий ответ.
Как правильно структурировать репозиторий проекта для эффективной командной работы в системе контроля версий?
6. Прочитайте текст. Дайте краткий ответ.
Какие стратегии ветвления (branching) рекомендуется использовать при совместной разработке модулей ИИ?
7. Прочитайте текст. Дайте краткий ответ.
Объясните, как использование системы контроля версий помогает управлять конфликтами при одновременной работе нескольких разработчиков.
8. Прочитайте текст. Дайте краткий ответ.
Какие практики рекомендуется соблюдать для обеспечения согласованности кода в команде с помощью системы контроля версий?
9. Прочитайте текст. Дайте краткий ответ.
Как организовать процесс ревью и слияния изменений в системе контроля версий для повышения качества кода?
10. Прочитайте текст. Дайте краткий ответ.
Какие инструменты интеграции систем контроля версий с системами автоматического тестирования и CI/cd применимы в проектах ИИ?
11. Прочитайте текст. Дайте краткий ответ.
Объясните роль тегов и меток (tags, labels) в управлении версиями программных модулей.
12. Прочитайте текст. Дайте краткий ответ.
Как обеспечить безопасность и контроль доступа к различным частям репозитория при групповой разработке?
13. Прочитайте текст. Дайте краткий ответ.
Какие особенности использования системы контроля версий необходимо учитывать при работе с большими моделями и данными в проектах ИИ?
14. Прочитайте текст. Дайте краткий ответ.
Как организовать работу над несколькими версиями модуля или модели одновременно с помощью системы контроля версий?
15. Прочитайте текст. Дайте краткий ответ.
В чем заключается важность ведения истории изменений и как это помогает при отладке и развитии проекта?
16. Прочитайте текст. Дайте краткий ответ.

Какие типичные ошибки допускают при использовании систем контроля версий в командных проектах по разработке ИИ-модулей?

17. Прочитайте текст. Дайте краткий ответ.

Как автоматизировать процессы обновления, тестирования и деплоя программных модулей с помощью системы контроля версий?

18. Прочитайте текст. Дайте краткий ответ.

Объясните, каким образом система контроля версий способствует соблюдению стандартов кодирования и документации.

19. Прочитайте текст. Дайте краткий ответ.

Какие рекомендации по обучению команды работе с системой контроля версий вы можете дать для повышения эффективности совместной разработки?

20. Прочитайте текст. Дайте краткий ответ.

Как организовать работу над экспериментальными или прототипными версиями моделей внутри системы контроля версий?

21. Прочитайте текст. Дайте краткий ответ.

В чем заключается отличие между централизованными и распределенными системами контроля версий, и какую из них предпочтительнее использовать для проектов ИИ?

22. Прочитайте текст. Дайте краткий ответ.

Как обеспечить согласованность данных и моделей при использовании системы контроля версий в рамках групповой разработки?

23. Прочитайте текст. Дайте краткий ответ.

Какие подходы существуют для интеграции системы контроля версий с системами управления проектами и задачами?

24. Прочитайте текст. Дайте краткий ответ.

Объясните, как использование системы контроля версий помогает управлять релизами и обновлениями программных модулей.

25. Прочитайте текст. Дайте краткий ответ.

Какие шаги необходимо предпринять для внедрения эффективной системы контроля версий в существующий проект по разработке модулей ИИ с учетом командной работы?