

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ
(АКТ (ф) СПбГУТ)**

УТВЕРЖДАЮ

И.о. зам. директора по учебной работе
Семенцына
12 11 2025 г.

**КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ
по междисциплинарному курсу
МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ В
СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА
программы подготовки специалистов среднего звена**

по специальности СПО

**09.02.13 Интеграция решений с применением технологий
искусственного интеллекта**

**г. Архангельск
2025**

Организация-разработчик: АКТ (ф) СПбГУТ.

Разработчики:

М.Н. Нехлебаева, преподаватель высшей квалификационной категории
АКТ (ф) СПбГУТ

С.В. Лукина, преподаватель высшей квалификационной категории АКТ
(ф) СПбГУТ

Рассмотрено и одобрено цикловой комиссией Информационных
технологий и математических дисциплин

Протокол № 3 от 16 ноября 2025 г.

Председатель М.Н. Нехлебаева

КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ
для промежуточной аттестации (дифференцированный зачет)
по МДК.01.01 Разработка программных модулей в системах
искусственного интеллекта
для специальности

09.02.13 Интеграция решений с применением технологий искусственного интеллекта

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:

В результате освоения МДК студент должен освоить следующие профессиональные компетенции:

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.

В ходе проведения дифференцированного зачета проверяется сформированность следующих навыков:

- разработка программные модули для систем искусственного интеллекта;
- выполнение предобработку и анализ данных;
- обучение, оценивание и улучшение модели машинного и глубокого обучения;
- проектирование и развертывание масштабируемых ИИ-систем;
- знание основных направлений, истории и этических аспектов ИИ;
- методы сбора, предобработки и анализа данных;
- алгоритмы машинного обучения и глубокого обучения;
- методы оценки и валидации моделей;
- принципы проектирования и контейнеризации ИИ-систем.

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).

Шкала оценивания образовательных результатов:

Оценка	Критерии
«отлично»	90-100% правильно выполненных заданий
«хорошо»	76-89% правильно выполненных заданий
«удовлетворительно»	60-75% правильно выполненных заданий
«неудовлетворительно»	правильно выполненных заданий менее 60%

Проверяемая компетенция - ПК 1.1.

Блок заданий закрытого типа

1) Прочитайте текст и выберите один правильный ответ.

Что является главной целью формализации алгоритма разработки программного модуля?

1. Сокращение времени на написание кода
2. Систематизация процесса для минимизации ошибок и соблюдения требований ТЗ
3. Ускорение процесса тестирования
4. Автоматическая генерация пользовательской документации

2) Прочитайте текст и выберите один правильный ответ.

Какой документ является первоисточником для формирования алгоритма разработки?

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

3) Прочитайте текст и выберите один правильный ответ.

Какой этап следует сразу после анализа требований ТЗ?

1. Непосредственное кодирование
2. Проектирование архитектуры модуля
3. Разворачивание системы
4. Написание финальной документации

4) Прочитайте текст и выберите один правильный ответ.

Почему декомпозиция ТЗ на подзадачи критически важна?

1. Для увеличения объема работ
2. Для упрощения планирования, контроля и оценки каждого этапа
3. Чтобы исключить необходимость тестирования
4. Чтобы сразу приступить к программированию

5) Прочитайте текст и выберите один правильный ответ.

Какой метод помогает визуализировать последовательность и зависимость задач в алгоритме разработки?

1. Диаграмма Парето
2. Ментальная карта
3. Блок-схема или PERT-диаграмма
4. Таблица случайных чисел

6) Прочтите текст и выберите один правильный ответ.

Что входит в этап проектирования программного модуля?

1. Определение структурных компонентов, интерфейсов и алгоритмов
2. Только написание исходного кода
3. Прямое развертывание на продакшн-сервере
4. Проведение финального демо для заказчика

7) Прочтите текст и выберите один правильный ответ.

Для чего при формировании алгоритма используют UML-диаграммы?

1. Для автоматической компиляции кода
2. Для визуального моделирования структуры и поведения системы
3. Для замены текстового ТЗ
4. Для увеличения скорости выполнения программы

8) Прочтите текст и выберите один правильный ответ.

Какие критерии ключевые при выборе технологий для реализации модуля?

1. Соответствие ТЗ, производительность, поддерживаемость
2. Максимальная популярность технологии на рынке
3. Минимизация количества строк кода
4. Использование только самых новых языков программирования

9) Прочтите текст и выберите один правильный ответ.

Что подразумевает принцип модульности в проектировании?

1. Создание монолитной системы
2. Разделение системы на слабосвязанные компоненты с четкими интерфейсами
3. Проектирование без учета будущих изменений
4. Использование только одного языка программирования

10) Прочтите текст и выберите один правильный ответ.

Что следует сделать после этапа проектирования перед началом кодирования?

1. Утвердить проектную документацию у ответственных лиц
2. Немедленно начать реализацию
3. Пропустить этап тестирования
4. Закупить оборудование

11) Прочтите текст и выберите один правильный ответ.

Почему анализ рисков является частью формирования алгоритма разработки?

1. Для предотвращения проблем и создания планов по их mitigation
2. Чтобы усложнить процесс планирования
3. Для исключения этапа тестирования
4. Чтобы увеличить бюджет проекта

12) Прочтите текст и выберите один правильный ответ.

Какой инструмент НЕ используется для формализации алгоритма разработки?

1. Jira или Trello для управления задачами
2. UML-редакторы для моделирования
3. Git для контроля версий
4. Adobe Photoshop для создания графики

13) Прочтите текст и выберите один правильный ответ.

Какова основная роль тестового плана в алгоритме разработки?

1. Замена технического задания

2. Обеспечение проверки соответствия модуля требованиям
3. Автоматическое исправление ошибок в коде
4. Сокращение времени на код-ревью

14) Прочтите текст и выберите один правильный ответ.

Какие виды тестирования должны быть заложены в алгоритм?

1. Только unit-тесты
2. Модульное, интеграционное, системное, приемочное
3. Только ручное тестирование
4. Только нагружочное тестирование

15) Прочтите текст и выберите один правильный ответ.

Почему документирование каждого этапа разработки обязательно?

1. Для обеспечения прозрачности, воспроизводимости и облегчения поддержки
2. Чтобы увеличить объем документации
3. Для сокрытия реального прогресса
4. Чтобы избежать этапа тестирования

16) Прочтите текст и выберите один правильный ответ.

Что является ключевым действием после завершения кодирования модуля?

1. Немедленный вывод в продакшен
2. Всестороннее тестирование и подготовка документации
3. Удаление исходного кода
4. Назначение нового разработчика на проект

17) Прочтите текст и выберите один правильный ответ.

Какую роль играет обратная связь от заказчика в процессе разработки?

1. Позволяет оперативно корректировать разработку в соответствии с ожиданиями
2. Замедляет процесс разработки
3. Требуется только на финальном этапе
4. Не является важной

18) Прочтите текст и выберите один правильный ответ.

Какие метрики используются для оценки эффективности алгоритма разработки?

1. Соответствие срокам, качество кода, соблюдение требований ТЗ
2. Количество строк кода
3. Сложность алгоритмов
4. Количество использованных библиотек

19) Прочтите текст и выберите один правильный ответ.

Зачем проводить ревью проектных решений до начала кодирования?

1. Для выявления недочетов и улучшения архитектуры на ранней стадии
2. Чтобы отложить начало разработки
3. Для автоматической генерации кода
4. Это необязательная процедура

20) Прочтите текст и выберите один правильный ответ.

Какие критерии определяют готовность модуля к переходу на следующий этап?

1. Выполнение всех пунктов ТЗ для этапа, успешное прохождение тестов
2. Истечание запланированного времени
3. Желание разработчика
4. Отсутствие документации

21) Прочтите текст и выберите один правильный ответ.

Что характеризует итеративный подход к разработке?

1. Постепенная разработка через циклы "планирование-реализация-оценка"
2. Реализация всех требований за один непрерывный цикл
3. Отказ от планирования
4. Разработка без тестирования

22) Прочтите текст и выберите один правильный ответ.

Как система контроля версий помогает в реализации алгоритма разработки?

1. Позволяет отслеживать изменения, Collaborate и управлять версиями
2. Автоматически исправляет ошибки
3. Заменяет систему управления проектами
4. Увеличивает производительность сервера

23) Прочтите текст и выберите один правильный ответ.

Почему требования безопасности должны быть учтены в алгоритме разработки?

1. Для защиты системы и данных от угроз на этапе проектирования
2. Безопасность можно добавить после релиза
3. Это увеличивает стоимость проекта
4. Требуется только для государственных проектов

24) Прочтите текст и выберите один правильный ответ.

Что нужно сделать перед началом реализации каждого нового компонента?

1. Убедиться в понимании требований и наличии необходимых ресурсов
2. Начать кодировать без подготовки
3. Пропустить модульное тестирование
4. Слить все ветки в основную

25) Прочтите текст и выберите один правильный ответ.

Какова итоговая цель формирования четкого алгоритма разработки?

1. Создание качественного продукта, отвечающего ТЗ, с предсказуемым процессом
2. Максимальное усложнение архитектуры
3. Сокращение времени любыми способами
4. Демонстрация использования модных технологий

Проверяемая компетенция - ПК 1.2.

Блок заданий закрытого типа

1) Прочтите текст и выберите один правильный ответ.

Какой этап является первым в непосредственной разработке модуля по ТЗ?

1. Анализ требований и проектирование
2. Публикация релиза
3. Написание пользовательской документации
4. Рефакторинг кода

2) Прочтите текст и выберите один правильный ответ.

Что является основным ориентиром для разработчика при создании модуля?

1. Личные предпочтения в кодировании
2. Требования, зафиксированные в ТЗ
3. Примеры кода из открытых источников
4. Ревью соседа по команде

3) Прочтите текст и выберите один правильный ответ.

Какой документ является фундаментом для всех этапов разработки?

1. План тестирования
2. Техническое задание (ТЗ)
3. Руководство по стилю кода
4. Договор с заказчиком

4) Прочтите текст и выберите один правильный ответ.

Зачем разбивать ТЗ на мелкие задачи?

1. Для упрощения реализации, оценки и контроля
2. Для создания видимости большого объема работы
3. Чтобы увеличить количество коммитов
4. Это требование системы контроля версий

5) Прочтите текст и выберите один правильный ответ.

Какой метод является наиболее эффективным для проверки соответствия кода ТЗ на ранних этапах?

1. Нагрузочное тестирование
2. Модульное (юнит) тестирование и код-ревью
3. Пользовательское приемочное тестирование
4. Визуальный осмотр интерфейса

6) Прочтите текст и выберите один правильный ответ.

Что необходимо сделать после написания кода модуля?

1. Слить его в основную ветку без проверки
2. Провести тестирование и валидацию против требований ТЗ
3. Немедленно отправить заказчику
4. Закрыть задачу в трекере

7) Прочтите текст и выберите один правильный ответ.

Какие нефункциональные требования часто указаны в ТЗ?

1. Производительность, надежность, безопасность
2. Цвет кнопок интерфейса
3. Имена переменных в коде
4. Стиль комментариев

8) Прочтите текст и выберите один правильный ответ.

Почему ведение документации в процессе разработки критически важно?

1. Для облегчения поддержки, онбординга новых разработчиков и аудита
2. Чтобы увеличить время работы над проектом
3. Это формальность, которую можно выполнить в конце
4. Для сокрытия реальной логики кода

9) Прочтите текст и выберите один правильный ответ.

Как обеспечить постоянное соответствие кода ТЗ в ходе работы?

1. Регулярно сверяться с требованиями ТЗ в процессе реализации
2. Довериться памяти разработчика
3. Игнорировать изменения в ТЗ
4. Использовать только самые новые технологии

10) Прочтите текст и выберите один правильный ответ.

Как система контроля версий помогает в разработке по ТЗ?

1. Позволяет отслеживать, какие изменения и для каких требований ТЗ были сделаны
2. Автоматически правит код под требования ТЗ
3. Заменяет собой техническое задание
4. Ускоряет выполнение программы

11) Прочтите текст и выберите один правильный ответ.

Какие практики напрямую способствуют качеству модуля?

1. Пропуск этапа тестирования для экономии времени
2. Модульное тестирование, код-ревью, CI/CD
3. Использование устаревших библиотек
4. Написание максимально сложного кода

12) Прочтите текст и выберите один правильный ответ.

Почему важно учитывать ограничения (память, процессор) при разработке?

1. Чтобы обеспечить работу системы в рамках заданных условий ТЗ
2. Чтобы увеличить стоимость оборудования
3. Это не важно для современных серверов
4. Требуется только для мобильных приложений

13) Прочтите текст и выберите один правильный ответ.

Что такое "модульная архитектура" и как она помогает выполнить ТЗ?

1. Разделение на независимые компоненты упрощает реализацию и проверку требований
2. Это монолит, который быстрее работает
3. Архитектура, не учитывающая безопасность
4. Подход, при котором весь код пишется в одном файле

14) Прочтите текст и выберите один правильный ответ.

Какое тестирование проверяет взаимодействие нескольких модулей?

1. Модульное
2. Интеграционное
3. Нагрузочное
4. Юнит-тестирование

15) Прочтите текст и выберите один правильный ответ.

Как управлять изменениями в ТЗ после начала разработки?

1. Внести изменения без уведомления команды
2. Вести журнал изменений ТЗ и formal process их согласования
3. Игнорировать новые пожелания заказчика
4. Начать новый проект

16) Прочтите текст и выберите один правильный ответ.

Зачем проводить код-ревью в процессе разработки?

1. Для выявления ошибок и несоответствий ТЗ силами команды
2. Чтобы затянуть сроки проекта
3. Это альтернатива модульному тестированию
4. Для обучения стажеров

17) Прочтите текст и выберите один правильный ответ.

Какой инструмент помогает автоматически проверить покрытие кода тестами?

1. Графический редактор

2. CI-сервер (например, Jenkins, GitLab CI) и инструменты покрытия (coverage.py)
3. Текстовый процессор
4. Система контроля версий

18) Прочтите текст и выберите один правильный ответ.

Что проверяет интеграционное тестирование?

1. Отдельные функции модуля
2. Взаимодействие и совместную работу нескольких модулей
3. Только пользовательский интерфейс
4. Устойчивость к хакерским атакам

19) Прочтите текст и выберите один правильный ответ.

Как реализовать требования безопасности из ТЗ в коде?

- Проигнорировать их до этапа тестирования
Внедрить валидацию данных, аутентификацию, шифрование
Использовать только сложные пароли
Это задача сисадмина, а не разработчика

20) Прочтите текст и выберите один правильный ответ.

Зачем нужно регрессионное тестирование?

1. Чтобы убедиться, что новые правки не сломали старый функционал
2. Для проверки нового функционала
3. Для оценки производительности
4. Это синоним модульного тестирования

21) Прочтите текст и выберите один правильный ответ.

Какие показатели характеризуют качество разработанного модуля?

1. Соответствие ТЗ, надежность, производительность, читаемость кода
2. Количество строк кода
3. Количество использованных паттернов
4. Сложность алгоритмов

22) Прочтите текст и выберите один правильный ответ.

Как документировать выполнение требований ТЗ?

1. Вести трекер задач, писать комментарии в коммитах, создавать тесты
2. Устно докладывать менеджеру
3. Не документировать, это лишняя работа
4. Делать скриншоты интерфейса

23) Прочтите текст и выберите один правильный ответ.

Какую роль играют автоматизированные тесты в соблюдении ТЗ?

1. Они постоянно проверяют, что код соответствует требованиям
2. Они замедляют процесс разработки
3. Они нужны только для больших проектов
4. Они заменяют код-ревью

24) Прочтите текст и выберите один правильный ответ.

Почему принцип модульности важен для систем ИИ?

1. Позволяет независимо развивать, тестировать и заменять компоненты (например, модель, обработку данных)
2. Это делает код более запутанным
3. Так исторически сложилось

4. Требуется только для микросервисной архитектуры

25) Прочтите текст и выберите один правильный ответ.

Как выполнить все требования ТЗ в сжатые сроки?

1. Применить итеративную разработку, расставить приоритеты и регулярно отслеживать прогресс
2. Пропустить этап тестирования
3. Увеличить рабочее время команды
4. Взять в команду больше разработчиков

Проверяемая компетенция - ПК 1.3.

Блок заданий закрытого типа

1) Прочтите текст и выберите один правильный ответ.

Почему соблюдение стандартов оформления кода обязательно в командной разработке?

1. Для повышения читаемости, поддерживаемости и единообразия кода
2. Чтобы код работал быстрее
3. Для сокращения количества строк кода
4. Это требование законодательства

2) Прочтите текст и выберите один правильный ответ.

Какая информация должна быть в комментарии к функции?

1. Назначение функции, описание параметров и возвращаемого значения
2. Имя разработчика и дата создания
3. Список использованных переменных
4. Случайные мысли разработчика

3) Прочтите текст и выберите один правильный ответ.

Какое правило именования переменных является хорошей практикой?

1. Использовать короткие абстрактные имена (a, b, x1)
2. Использовать осмысленные имена, отражающие суть данных (user_profile, total_count)
3. Использовать транслит (imyaPolzovatelya)
4. Именовать все в верхнем регистре (USER, DATA)

4) Прочтите текст и выберите один правильный ответ.

Что является хорошей практикой структурирования кода?

1. Логическое разделение на блоки, использование отступов и пробелов
2. Писать весь код в одну строку
3. Не использовать пустые строки
4. Объединять все функции в одну

5) Прочтите текст и выберите один правильный ответ.

Какой стандарт оформления кода является общепринятым для Python?

1. PEP 8
2. ISO 9001
3. ГОСТ 34
4. IEEE 802

6) Прочтите текст и выберите один правильный ответ.

Как правильно оформить длинную строку кода?

1. Оставить ее как есть, пусть будет длинной

2. Разбить на несколько строк с использованием переноса
3. Уменьшить размер шрифта в редакторе
4. Закомментировать ее

7) Прочтите текст и выберите один правильный ответ.

Как документировать сложный алгоритм?

1. Подробно описать логику, шаги и примеры в комментариях
2. Не документировать, чтобы не выдать коммерческую тайну
3. Сделать код еще сложнее, чтобы никто не разобрался
4. Документировать только входные и выходные данные

8) Прочтите текст и выберите один правильный ответ.

Что должно быть задокументировано в заголовке файла с кодом?

1. Назначение файла, автор, дата, возможно, лицензия
2. Полная история изменений файла
3. Список всех функций в файле
4. Системные требования для запуска

9) Прочтите текст и выберите один правильный ответ.

Зачем нужны соглашения об именовании (code style guide)?

Для единства, чтобы код любого разработчика команды выглядел привычно

Чтобы упростить жизнь новичкам

Для уменьшения размера файлов с кодом

Это устаревшая практика

10) Прочтите текст и выберите один правильный ответ.

Какое правило оформления функций является лучшим?

1. Функция должна выполнять одну задачу и иметь понятное имя
2. Функция должна быть максимально длинной и сложной
3. Имя функции может быть любым
4. Не использовать функции, а писать все последовательно

11) Прочтите текст и выберите один правильный ответ.

Как оформить условный оператор для лучшей читаемости?

1. Писать все условия в одну строку
2. Использовать отступы для тела условия и избегать сложных вложений
3. Не использовать else
4. Заменять if на goto

12) Прочтите текст и выберите один правильный ответ.

Как оформить цикл?

1. Использовать понятное имя для итератора, отступы для тела цикла
2. Использовать только while
3. Избегать комментариев внутри цикла
4. Все циклы писать в конце файла

13) Прочтите текст и выберите один правильный ответ.

Почему единство оформления во всем проекте важно?

1. Облегчает чтение, понимание и рефакторинг кода всей командой
2. Это личное дело каждого разработчика
3. Позволяет использовать автоматическое форматирование
4. Требуется только для open-source проектов

14) Прочтите текст и выберите один правильный ответ.

Какие элементы оформления предотвращают синтаксические ошибки?

1. Соблюдение отступов и парности скобок/кавычек
2. Использование темной темы в редакторе
3. Частое сохранение файла
4. Написание кода без пробелов

15) Прочтите текст и выберите один правильный ответ.

Что делать с длинной функцией?

1. Оставить как есть
2. Разделить на несколько более мелких и специализированных функций
3. Закомментировать часть кода
4. Удалить ее

16) Прочтите текст и выберите один правильный ответ.

Как правильно оформить импорт библиотек?

1. Группировать импорты (стандартные, сторонние, собственные), обычно в начале файла
2. Импортировать модули в том месте, где они используются
3. Использовать только `from module import *`
4. Не использовать импорты

17) Прочтите текст и выберите один правильный ответ.

Как оформлять документацию для класса?

1. Использовать docstrings для описания назначения класса, его атрибутов и методов
2. Писать документацию в отдельном Word-файле
3. Не документировать классы
4. Документировать только публичные методы

18) Прочтите текст и выберите один правильный ответ.

Как оформить обработку исключений?

1. Использовать конкретные типы исключений и добавлять информативные сообщения
2. Ловить все исключения одним блоком `except:`
- 3.忽 置略
4. Использовать исключения для управления обычной логикой программы

19) Прочтите текст и выберите один правильный ответ.

Какие правила именования помогают в поддержке кода?

1. Осмысленные имена, консистентный стиль (`snake_case`, `CamelCase`)
2. Использование однобуквенных имен
3. Именование на родном языке
4. Частая смена имен переменных

20) Прочтите текст и выберите один правильный ответ.

Как оформить тестовый код?

1. Размещать в отдельных файлах или директориях, ясно маркированных как тесты
2. Смешивать с продакшен-кодом
3. Удалять после первого успешного прогона
4. Дублировать в каждом модуле

21) Прочтите текст и выберите один правильный ответ.

Почему единый стиль кода важен для проектов ИИ?

1. Облегчает совместную работу над сложными алгоритмами и моделями

2. Модели машинного обучения требуют особого стиля
3. Это требование фреймворков типа TensorFlow
4. Не важен, главное - точность модели

22) Прочтите текст и выберите один правильный ответ.

Что делает код понятным для других?

1. Осмысленные имена, чистая структура, комментарии где необходимо
2. Максимальная краткость
3. Использование продвинутых языковых конструкций
4. Отсутствие комментариев

23) Прочтите текст и выберите один правильный ответ.

Как оформить вложенные условные конструкции?

1. Избегать глубокой вложенности, использовать отступы, возможно, выносить в функции
2. Писать все условия в одну строку
3. Заменять их на match-case где возможно
4. Не использовать больше двух уровней вложенности

24) Прочтите текст и выберите один правильный ответ.

Что сделать с кодом перед сдачей на проверку?

1. Проверить его соответствие стандартам оформления и запустить линтеры
2. Удалить все комментарии
3. Переименовать все переменные в случайные имена
4. Ничего, сдавать как есть

25) Прочтите текст и выберите один правильный ответ.

Как документировать сложный алгоритм машинного обучения?

Описать общую идею, входные/выходные данные, ключевые шаги и параметры

Документировать только формулы

Считать код самодокументируемым

Приложить научную статью

Проверяемая компетенция - ПК 1.4.

Блок заданий закрытого типа

1) Прочтите текст и выберите один правильный ответ.

Какова основная цель системы контроля версий (СКВ) в команде?

1. Совместная работа над кодом с отслеживанием изменений и историей
2. Автоматическое исправление ошибок в коде
3. Компиляция проекта
4. Хранение паролей

2) Прочтите текст и выберите один правильный ответ.

Какой инструмент является СКВ?

1. Docker
2. Git
3. Jenkins
4. PyCharm

3) Прочтите текст и выберите один правильный ответ.

Что такое "ветка" (branch)?

1. Изолированная линия разработки для функциональности или исправления
2. Основная версия проекта
3. История коммитов
4. Файл с настройками

4) Прочтите текст и выберите один правильный ответ.

Почему СКВ необходима в групповой разработке?

1. Для координации работы, разрешения конфликтов и сохранения истории
2. Чтобы автоматически деплоить приложение
3. Для замены системы тестирования
4. Чтобы код работал быстрее

5) Прочтите текст и выберите один правильный ответ.

Что такое "merge" (слияние)?

1. Объединение изменений из одной ветки в другую
2. Создание новой ветки
3. Удаление старой версии кода
4. Отправка кода на сервер

6) Прочтите текст и выберите один правильный ответ.

Какое ключевое преимущество Git для команды?

1. Распределенная архитектура, каждый имеет полную историю
2. Централизованное хранение только на сервере
3. Автоматическое реview кода
4. Интегрированный текстовый редактор

7) Прочтите текст и выберите один правильный ответ.

Что такое "конфликт слияния"?

1. Ошибка компиляции
2. Несовместимые изменения в одном файле, которые Git не может разрешить автоматически
3. Спор между разработчиками
4. Проблема с сетью

8) Прочтите текст и выберите один правильный ответ.

Как минимизировать конфликты при слиянии?

1. Часто делать коммиты и синхронизироваться с удаленной веткой (pull/rebase)
2. Работать только в одной ветке
3. Не делать коммиты
4. Использовать устаревшую версию кода

9) Прочтите текст и выберите один правильный ответ.

Что такое "коммит"?

1. Фиксация набора изменений в локальном репозитории с комментарием
2. Отправка изменений на удаленный сервер
3. Создание ветки
4. Обновление локальной копии из удаленного репозитория

10) Прочтите текст и выберите один правильный ответ.

Почему сообщения к коммитам должны быть осмысленными?

1. Чтобы по истории было понятно, что и зачем было изменено
2. Чтобы увеличить размер репозитория
3. Это требование Git
4. Для автоматического генерации документации

11) Прочтите текст и выберите один правильный ответ.

Что такое "удаленный репозиторий" (remote)?

1. Локальная папка с проектом
2. Общая версия репозитория, размещенная на сервере (например, GitHub)
3. Ветка для экспериментального кода
4. История изменений одного файла

12) Прочтите текст и выберите один правильный ответ.

Какая модель ветвления популярна для командной работы?

1. Git Flow (с ветками main, develop, feature, release, hotfix)
2. Одна ветка на всех
3. Ветка для каждого разработчика
4. Создание ветки для каждого коммита

13) Прочтите текст и выберите один правильный ответ.

Какой процесс объединяет изменения из удаленной ветки в локальную?

1. git push
2. git commit
3. git pull или git fetch + git merge
4. git clone

14) Прочтите текст и выберите один правильный ответ.

Что такое "Pull Request" (PR) или "Merge Request" (MR)?

1. Запрос на удаление ветки
2. Запрос на проверку и слияние кода из одной ветки в другую
3. Запрос на доступ к репозиторию
4. Отчет об ошибках

15) Прочтите текст и выберите один правильный ответ.

Почему рекомендуется делать частые небольшие коммиты?

1. Упрощает отслеживание изменений и отмену конкретных правок
2. Это замедляет работу Git
3. Так требуют менеджеры
4. Чтобы быстрее заполнить историю

16) Прочтите текст и выберите один правильный ответ.

Какой инструмент помогает управлять зависимостями проектов в Git?

1. Git Submodules
2. Dockerfile
3. .gitignore
4. README.md

17) Прочтите текст и выберите один правильный ответ.

Как называется ветка, где ведется основная разработка новых функций?

1. main/master
2. develop
3. hotfix
4. release

18) Прочтите текст и выберите один правильный ответ.

Как организовать контроль доступа к репозиторию?

1. Использовать роли и права доступа на платформе (GitLab/GitHub)
2. Дать всем один пароль
3. Хранить репозиторий локально у тимлида
4. Не ограничивать доступ

19) Прочтите текст и выберите один правильный ответ.

Как защититься от потери кода при использовании Git?

1. Регулярно делать push в удаленный репозиторий
2. Держать все только на локальной машине
3. Периодически удалять историю
4. Архивировать папку проекта

20) Прочтите текст и выберите один правильный ответ.

Чем команда git fetch отличается от git pull?

1. fetch загружает изменения, но не сливает, pull — загружает и сразу сливает
2. fetch отправляет изменения, pull — загружает
3. Это синонимы
4. fetch для создания веток, pull — для их удаления

21) Прочтите текст и выберите один правильный ответ.

Зачем делать git pull перед началом работы?

1. Чтобы получить последние изменения из общей ветки и избежать конфликтов
2. Чтобы проверить, есть ли интернет
3. Это необязательная операция
4. Чтобы очистить историю

22) Прочтите текст и выберите один правильный ответ.

Что делает git rebase в отличие от git merge?

1. Перемещает вашу ветку на верх последнего коммита целевой ветки, "переписывая" историю
2. Создает новый коммит слияния
3. Удаляет ваши коммиты
4. Это то же самое

23) Прочтите текст и выберите один правильный ответ.

Как СКВ помогает в проектах ИИ?

1. Позволяет версионировать не только код, но и данные, модели и эксперименты
2. Ускоряет обучение моделей
3. Автоматически подбирает гиперпараметры
4. Заменяет MLflow

24) Прочтите текст и выберите один правильный ответ.

Как организовать работу над несколькими фичами параллельно?

1. Создать отдельную ветку для каждой функциональности от общей базы
2. Работать всем в main
3. Копировать папку проекта для каждой фичи
4. Использовать разные репозитории

25) Прочтите текст и выберите один правильный ответ.

Что нужно сделать перед слиянием ветки через PR/MR?

1. Провести код-ревью, убедиться, что тесты проходят
2. Слитъ немедленно, не теряя времени
3. Удалить ветку
4. Создать новую ветку

Проверяемая компетенция - ПК 1.1.

Блок заданий открытого типа

1. Прочтите текст. Дайте краткий ответ.

Перечислите основные этапы жизненного цикла разработки программного модуля.

2. Прочтите текст. Дайте краткий ответ.

Какие разделы обычно включает в себя техническое задание (ТЗ) на модуль?

3. Прочтите текст. Дайте краткий ответ.

В чем заключается суть этапа анализа требований?

4. Прочтите текст. Дайте краткий ответ.

Назовите два инструмента для моделирования архитектуры программной системы.

5. Прочтите текст. Дайте краткий ответ.

Как методология Agile влияет на формирование алгоритма разработки?

6. Прочтите текст. Дайте краткий ответ.

Объясните, что такое "спринт" в контексте Scrum.

7. Прочтите текст. Дайте краткий ответ.

Что такое "User Story" и из каких частей она состоит?

8. Прочтите текст. Дайте краткий ответ.

Какие критерии используются для оценки сложности задачи (story points)?

9. Прочтите текст. Дайте краткий ответ.

Как диаграмма Ганта помогает в планировании алгоритма разработки?

10. Прочтите текст. Дайте краткий ответ.

Назовите три основных риска при разработке ПО и способы их mitigation.

11. Прочтите текст. Дайте краткий ответ.

Как требования безопасности интегрируются в процесс разработки?

12. Прочтите текст. Дайте краткий ответ.

Какую роль в алгоритме разработки играет прототипирование?

13. Прочтите текст. Дайте краткий ответ.

Какие виды тестирования должны быть запланированы в алгоритме?

14. Прочтите текст. Дайте краткий ответ.

Зачем нужна документация по проектированию (design doc)?

15. Прочтите текст. Дайте краткий ответ.

Как инструменты CI/CD (непрерывная интеграция/доставка) влияют на процесс разработки?

16. Прочтите текст. Дайте краткий ответ.

Как учитываются ограничения по времени и бюджету при формировании алгоритма?

17. Прочитайте текст. Дайте краткий ответ.

Почему обратная связь от заказчика важна на этапе проектирования?

18. Прочитайте текст. Дайте краткий ответ.

Какие критерии говорят о готовности модуля к передаче в тестирование?

19. Прочитайте текст. Дайте краткий ответ.

Как автоматизировать сбор метрик о процессе разработки?

20. Прочитайте текст. Дайте краткий ответ.

В чем преимущество итеративного подхода для проектов ИИ?

21. Прочитайте текст. Дайте краткий ответ.

Какие особенности данных в ИИ влияют на алгоритм разработки модуля?

22. Прочитайте текст. Дайте краткий ответ.

Как обеспечить масштабируемость архитектуры ИИ-системы на этапе проектирования?

23. Прочитайте текст. Дайте краткий ответ.

Как проходит типичное ревью проектных решений?

24. Прочитайте текст. Дайте краткий ответ.

Какие метрики используют для отслеживания прогресса разработки по спринту?

25. Прочитайте текст. Дайте краткий ответ.

Опишите подготовку к реализации типичного пользовательского сценария.

Проверяемая компетенция - ПК 1.2.

Блок заданий открытого типа

1. Прочитайте текст. Дайте краткий ответ.

Опишите последовательность действий от получения ТЗ до начала кодирования.

2. Прочитайте текст. Дайте краткий ответ.

Как проверить непротиворечивость требований в ТЗ?

3. Прочитайте текст. Дайте краткий ответ.

Какие параметры модуля определяются на этапе проектирования?

4. Прочитайте текст. Дайте краткий ответ.

Как разработчик проверяет, что его код соответствует ТЗ в процессе работы?

5. Прочитайте текст. Дайте краткий ответ.

Назовите три метода проверки соответствия модуля ТЗ.

6. Прочитайте текст. Дайте краткий ответ.

Какую документацию создают в процессе разработки модуля?

7. Прочитайте текст. Дайте краткий ответ.

С какими типичными сложностями сталкивается разработчик при реализации ТЗ?

8. Прочитайте текст. Дайте краткий ответ.

Как организовать код-ревью в команде для проверки соответствия ТЗ?

9. Прочитайте текст. Дайте краткий ответ.

Какие инструменты CI/CD помогают автоматизировать проверку соответствия ТЗ?

10. Прочитайте текст. Дайте краткий ответ.

Как формализовать процесс внесения изменений в ТЗ?

11. Прочитайте текст. Дайте краткий ответ.

По каким критериям тестировщик проверяет соответствие модуля ТЗ?

12. Прочитайте текст. Дайте краткий ответ.

Как декомпозировать крупный раздел ТЗ на задачи для разработчиков?

13. Прочитайте текст. Дайте краткий ответ.

Какие аспекты разработки ИИ-модулей требуют особого внимания к ТЗ (напр., качество данных)?

14. Прочитайте текст. Дайте краткий ответ.

Как в ТЗ отражаются требования к производительности ИИ-модели?

15. Прочитайте текст. Дайте краткий ответ.

Какие тесты наиболее эффективны для проверки алгоритмов машинного обучения?

16. Прочитайте текст. Дайте краткий ответ.

Как взаимодействовать с аналитиком для уточнения неясных требований?

17. Прочитайте текст. Дайте краткий ответ.

Назовите частые ошибки при реализации ТЗ и способы их предотвращения.

18. Прочитайте текст. Дайте краткий ответ.

Какую роль в проверке ТЗ играет создание прототипа или MVP?

19. Прочитайте текст. Дайте краткий ответ.

Какие разделы должны быть в документации к разработанному модулю?

20. Прочитайте текст. Дайте краткий ответ.

Как реализовать в коде требования к безопасности данных в ИИ-системе?

21. Прочитайте текст. Дайте краткий ответ.

Как обеспечить совместимость вашего модуля с другими, используя ТЗ?

22. Прочитайте текст. Дайте краткий ответ.

Опишите процесс управления изменениями в ТЗ по ходу разработки.

23. Прочитайте текст. Дайте краткий ответ.

Как оценить трудозатраты на основе ТЗ?

24. Прочитайте текст. Дайте краткий ответ.

Почему обратная связь от тестировщиков критична для соответствия ТЗ?

25. Прочитайте текст. Дайте краткий ответ.

Назовите три лучшие практики, обеспечивающие высокое качество кода по ТЗ.

Проверяемая компетенция - ПК 1.3.

Блок заданий открытого типа

1. Прочитайте текст. Дайте краткий ответ.

Перечислите ключевые элементы читаемого кода.

2. Прочитайте текст. Дайте краткий ответ.

Почему стандарты кодирования особенно важны в машинном обучении?

3. Прочитайте текст. Дайте краткий ответ.

Какую информацию включает качественный комментарий к функции?

4. Прочитайте текст. Дайте краткий ответ.

Сравните стили именования snake_case и CamelCase. Где они применяются?

5. Прочитайте текст. Дайте краткий ответ.

Как плохое оформление кода влияет на стоимость его поддержки?

6. Прочитайте текст. Дайте краткий ответ.

Назовите два инструмента для автоматической проверки стиля кода на Python.

7. Прочитайте текст. Дайте краткий ответ.

Какова роль документационных строк (docstrings) в коде?

8. Прочитайте текст. Дайте краткий ответ.

Какие особенности оформления кода для нейросетевых моделей вы знаете?

9. Прочитайте текст. Дайте краткий ответ.

Опишите принципы хорошей структуризации кода (разделение на модули, пакеты).

10. Прочитайте текст. Дайте краткий ответ.

Как проверить, что код соответствует стандартам, перед коммитом?

11. Прочитайте текст. Дайте краткий ответ.

Почему единообразные отступы и форматирование важны?

12. Прочитайте текст. Дайте краткий ответ.

Как оформить большой класс с множеством методов для ясности?

13. Прочитайте текст. Дайте краткий ответ.

Какие правила оформления применяют к тестовому коду (например, в pytest)?

14. Прочитайте текст. Дайте краткий ответ.

Как хорошее оформление кода помогает при его отладке?

15. Прочитайте текст. Дайте краткий ответ.

Назовите три типичные ошибки в оформлении кода у начинающих.

16. Прочитайте текст. Дайте краткий ответ.

Как правильно оформить блок обработки исключений в Python?

17. Прочитайте текст. Дайте краткий ответ.

Какие требования предъявляют к оформлению конфигурационных файлов (например, YAML, JSON)?

18. Прочитайте текст. Дайте краткий ответ.

Как добиться единства единого стиля в команде из 5 разработчиков?

19. Прочитайте текст. Дайте краткий ответ.

Почему при интеграции модуля в систему его оформление играет важную роль?

20. Прочитайте текст. Дайте краткий ответ.

Назовите инструмент для автоматического форматирования кода на Python.

21. Прочитайте текст. Дайте краткий ответ.

Как комментарии помогают понять бизнес-логику, а не только код?

22. Прочитайте текст. Дайте краткий ответ.

Какие соглашения по оформлению кода приняты в фреймворке TensorFlow/PyTorch?

23. Прочитайте текст. Дайте краткий ответ.

Что должно входить в документацию по использованию модуля?

24. Прочитайте текст. Дайте краткий ответ.

Почему единый стиль во всем проекте важен и как его поддерживать?

25. Прочитайте текст. Дайте краткий ответ.

Опишите процесс приведения унаследованного (legacy) кода в соответствие со стандартами.

Проверяемая компетенция - ПК 1.4.

Блок заданий открытого типа

1. Прочитайте текст. Дайте краткий ответ.

Почему Git является стандартом де-факто в разработке ИИ?

2. Прочитайте текст. Дайте краткий ответ.

Назовите три преимущества Git для групповой работы.

3. Прочитайте текст. Дайте краткий ответ.

Опишите базовый рабочий цикл Git (modify, commit, push).

4. Прочитайте текст. Дайте краткий ответ.

Какие требования к процессу команды предъявляет при выборе Git?

5. Прочитайте текст. Дайте краткий ответ.

Как структурировать репозиторий для проекта машинного обучения (код, данные, модели)?

6. Прочитайте текст. Дайте краткий ответ.

Опишите стратегию ветвления GitFlow.

7. Прочитайте текст. Дайте краткий ответ.

Как разрешать конфликты слияния в Git?

8. Прочитайте текст. Дайте краткий ответ.

Какие практики коммитов способствуют чистой истории?

9. Прочитайте текст. Дайте краткий ответ.

Как организовать процесс рецензии кода через Pull Request в GitHub/GitLab?

10. Прочитайте текст. Дайте краткий ответ.

Как интегрировать Git с CI/CD пайплайном для автоматического тестирования?

11. Прочитайте текст. Дайте краткий ответ.

Для чего используются теги (tags) в Git?

12. Прочитайте текст. Дайте краткий ответ.

Как настроить права доступа к репозиторию на GitLab?

13. Прочитайте текст. Дайте краткий ответ.

Какие инструменты используют для версионирования больших файлов (данные, модели) в Git?

14. Прочитайте текст. Дайте краткий ответ.

Как работать над двумя разными версиями модели одновременно?

15. Прочитайте текст. Дайте краткий ответ.

Как история коммитов помогает при отладке регрессионной ошибки?

16. Прочитайте текст. Дайте краткий ответ.

Назовите типичные ошибки новичков в Git и как их избежать.

17. Прочитайте текст. Дайте краткий ответ.

Как автоматизировать деплой модели при пуше в определенную ветку?

18. Прочитайте текст. Дайте краткий ответ.

Как Git способствует соблюдению стандартов кодирования?

19. Прочитайте текст. Дайте краткий ответ.

Какие ресурсы использовать для обучения команды работе с Git?

20. Прочитайте текст. Дайте краткий ответ.

Как организовать ветку для эксперимента с новой архитектурой нейросети?

21. Прочитайте текст. Дайте краткий ответ.

В чем разница между централизованной (SVN) и распределенной (Git) СКВ?

22. Прочитайте текст. Дайте краткий ответ.

Как гарантировать консистентность данных и кода модели в разных ветках?

23. Прочитайте текст. Дайте краткий ответ.

Как связать коммиты в Git с задачами в Jira?

24. Прочитайте текст. Дайте краткий ответ.

Как Git помогает управлять версиями релизов (семантическое версионирование)?

25. Прочитайте текст. Дайте краткий ответ.

Опишите шаги по внедрению Git в существующий проект с нуля.