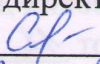


**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

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

УТВЕРЖДАЮ

И.о. зам. директора по учебной работе

 К.А. Семенцына

12 11 2025 г.

КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ

по междисциплинарному курсу

МДК 01.02 РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С

ПОДДЕРЖКОЙ 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 Нехлебаева М.Н. Нехлебаева

КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ
для промежуточной аттестации (дифференцированный зачет)
по МДК.01.02 Разработка мобильных приложений с поддержкой
искусственного интеллекта
для специальности

09.02.13 Интеграция решений с применением технологий искусственного интеллекта

Дифференцированный зачет является промежуточной формой контроля, подводит итог освоения МДК 01.02 Разработка мобильных приложений с поддержкой искусственного интеллекта в 5 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:

- понимание платформ мобильной разработки (Android, iOS) и их особенностей
- владение основами языков программирования Kotlin и Java для мобильной разработки;
- понимание принципов работы фреймворка TensorFlow Lite для мобильных устройств
- применение методов оптимизации моделей искусственного интеллекта для мобильных платформ;
- управление памятью и ресурсами устройства в контексте ИИ-приложений;
- проектирование интуитивного и отзывчивого пользовательского интерфейса (UI)
- использование технологий распознавания и синтеза речи (Text-to-Speech) для мобильных приложений;
- взаимодействие с сенсорами мобильного устройства для сбора данных
- использование систем контроля версий (например, Git) для управления проектом;
- проведение автоматизированного тестирования UI с использованием Espresso и Appium
- организация процесса публикации и сопровождения приложений в магазинах

(Google Play) соблюдение требований к безопасности и конфиденциальности данных в мобильных приложениях

- настройка сред разработки (Android Studio) и эмуляция мобильных устройств;
- создание пользовательских интерфейсов для мобильных приложений интеграция предобученных и кастомных моделей TensorFlow Lite в приложение;
- выполнение предварительной обработки входных данных для работы модели ИИ реализация логики инференса (вывода) модели на мобильном устройстве визуализация и;
- представление результатов работы модели ИИ в интерфейсе;
- обработка ошибок и исключительных ситуаций, связанных с работой ИИ-модулей;
- реализация функций захвата, выбора и обработки изображений для их распознавания;
- разработка и интеграция функционала голосового управления приложением;
- создание диалоговых интерфейсов для взаимодействия с голосовым помощником;
- написание автоматизированных UI-тестов для проверки функциональности ИИ-компонентов;
- проведение альфа- и бета-тестирования приложения перед публичным релизом;
- подготовка приложения к публикации (сборка APK/AAB, настройка метаданных);
- публикация приложения в Google Play Market и управление его жизненным циклом;
- мониторинг работы опубликованного приложения и подготовка обновлений;
- оптимизация производительности приложения, балансируя между скоростью и точностью модели.

Дифференцированный зачет проводится в форме тестирования. Тест содержит 25 из 120 вопросов (15 заданий закрытого типа и 10 вопросов 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 основной целью формализации алгоритма перед его реализацией?

1. Создание графического интерфейса
2. Однозначное определение последовательности действий для решения задачи
3. Написание пользовательской документации
4. Покупка хостинга для приложения

2) Прочитайте текст и выберите один правильный ответ.

Почему важно оценивать вычислительную сложность алгоритма на этапе проектирования?

1. Для предварительной оценки необходимых ресурсов и времени выполнения
2. Для увеличения количества строк кода
3. Для упрощения интерфейса пользователя
4. Для выбора цвета схемы приложения

3) Прочитайте текст и выберите один правильный ответ.

Какой этап следует сразу после определения входных и выходных данных алгоритма?

1. Публикация приложения
2. Описание основных шагов преобразования входов в выходы
3. Написание юнит-тестов
4. Запуск рекламной кампании

4) Прочитайте текст и выберите один правильный ответ.

Что включает в себя создание блок-схемы алгоритма?

1. Реализацию финального кода на Python
2. Визуальное представление логики с использованием стандартных символов
3. Написание текстовых инструкций для пользователя
4. Профилирование производительности

5) Прочитайте текст и выберите один правильный ответ.

Почему при проектировании алгоритма для мобильного приложения важно минимизировать количество операций ввода-вывода?

1. Чтобы снизить энергопотребление и увеличить скорость отклика
2. Чтобы усложнить логику приложения
3. Чтобы использовать больше памяти
4. Чтобы увеличить размер устанавливаемого пакета

6) Прочитайте текст и выберите один правильный ответ.

Какое из следующих действий способствует созданию надежного алгоритма?

1. Учет и обработка потенциальных ошибочных состояний и некорректных данных
2. Предположение, что все входные данные всегда идеальны
3. Игнорирование сообщений системы
4. Использование недокументированных функций ОС

7) Прочитайте текст и выберите один правильный ответ.

Что такое "инвариант" в контексте проектирования алгоритма?

1. Условие, которое остается истинным на протяжении всего выполнения алгоритма
2. Переменная, которая меняется чаще всего
3. Функция с наибольшим количеством параметров
4. Неиспользуемый участок кода

8) Прочитайте текст и выберите один правильный ответ.

Почему важно документировать принятые в алгоритме допущения и ограничения?

1. Для понимания границ применимости алгоритма и облегчения будущих модификаций
2. Для увеличения объема технической документации
3. Чтобы скрыть возможные недостатки
4. Для автоматической генерации кода

9) Прочитайте текст и выберите один правильный ответ.

Какие инструменты помогают формально проверить корректность логики алгоритма?

1. Статический анализ и верификация с помощью формальных методов
2. Только запуск на реальном устройстве
3. Просмотр случайных скриншотов
4. Опрос пользователей о их ожиданиях

10) Прочитайте текст и выберите один правильный ответ.

Что необходимо сделать после формального описания алгоритма перед передачей его на реализацию?

1. Провести ревью логики с коллегами или заинтересованными сторонами
2. Немедленно приступить к кодированию
3. Отправить приложение в магазин приложений
4. Удалить все черновики и схемы

11) Прочитайте текст и выберите один правильный ответ.

Какие критерии важны при оценке оптимальности разработанного алгоритма?

1. Точность решения, использование ресурсов, простота реализации
2. Количество использованных переменных
3. Популярность языка программирования для реализации
4. Субъективная красота блок-схемы

12) Прочитайте текст и выберите один правильный ответ.

Почему при проектировании алгоритмов для интерактивных приложений важно минимизировать время отклика?

1. Чтобы обеспечить плавность работы и избежать "зависаний" интерфейса
2. Чтобы увеличить нагрузку на центральный процессор
3. Чтобы использовать больше оперативной памяти
4. Чтобы усложнить архитектуру приложения

13) Прочитайте текст и выберите один правильный ответ.

Какое действие помогает адаптировать алгоритм под различные разрешения экрана и ориентации устройства?

1. Использование относительных координат и адаптивных шаблонов проектирования
2. Жесткая привязка к абсолютным координатам одного разрешения
3. Игнорирование ориентации портрет/ландшафт
4. Увеличение сложности вычислений без изменения логики вывода

14) Прочитайте текст и выберите один правильный ответ.

Что такое "рекурсивный алгоритм"?

1. Алгоритм, который вызывает сам себя для решения подзадач
2. Алгоритм, который выполняется только один раз
3. Алгоритм, использующий только циклы for
4. Алгоритм без использования функций

15) Прочитайте текст и выберите один правильный ответ.

Почему важно соблюдать принцип "разделяй и властвуй" при проектировании сложных алгоритмов?

1. Для декомпозиции сложной задачи на более простые и управляемые подзадачи
2. Чтобы ускорить процесс разработки без анализа задачи
3. Для увеличения количества взаимозависимостей
4. Чтобы снизить требования к тестированию

16) Прочитайте текст и выберите один правильный ответ.

Какие методы помогают оптимизировать алгоритмы поиска по большим объемам локальных данных на мобильном устройстве?

1. Использование эффективных структур данных (хэш-таблицы, индексы), предварительная сортировка
2. Последовательный перебор всех данных при каждом запросе
3. Игнорирование возможности кэширования результатов
4. Использование только облачного поиска

17) Прочитайте текст и выберите один правильный ответ.

Как обеспечить возможность легкого внесения изменений в алгоритм в будущем согласно ТЗ?

1. Проектировать алгоритм модульно, с минимальной связностью компонентов
2. Создавать монолитные, сильно связанные блоки логики
3. Ограничивать функциональность только самыми базовыми операциями
4. Не документировать принятые решения

18) Прочитайте текст и выберите один правильный ответ.

Что такое "жадный алгоритм" (greedy algorithm)?

1. Алгоритм, принимающий локально оптимальные решения в надежде найти глобальный оптимум
2. Алгоритм, который требует много ресурсов
3. Алгоритм, который всегда работает медленно
4. Алгоритм, который не использует циклы

19) Прочитайте текст и выберите один правильный ответ.

Почему важно учитывать возможность аппаратного ускорения (например, GPU) при проектировании алгоритмов для мобильных устройств?

1. Чтобы значительно повысить производительность вычисление-емких задач (обработка изображений, ИИ)
2. Чтобы использовать только центральный процессор для всех задач
3. Чтобы снизить качество визуализации
4. Чтобы увеличить время разработки

20) Прочитайте текст и выберите один правильный ответ.

Какие действия помогают обеспечить конфиденциальность данных в алгоритме их обработки?

1. Минимизация логгирования чувствительных данных, использование шифрования при хранении/передаче
2. Открытое хранение всех данных в текстовых файлах
3. Передача паролей в незашифрованном виде
4. Использование устаревших протоколов связи

21) Прочитайте текст и выберите один правильный ответ.

Как правильно реализовать алгоритм, требующий принятия решения на основе нескольких условий?

1. Использовать читаемые условные конструкции и таблицы решений для сложных случаев
2. Записывать все условия в одну длинную строку
3. Игнорировать редкие, но возможные комбинации условий
4. Случайным образом выбирать одно из условий

22) Прочитайте текст и выберите один правильный ответ.

Почему важно проводить анализ алгоритма на предмет "состояния гонки" (race condition) в многопоточных приложениях?

1. Чтобы избежать непредсказуемого поведения из-за параллельного доступа к общим ресурсам
2. Чтобы убедиться, что алгоритм работает только в одном потоке
3. Для проверки только пользовательского интерфейса
4. Для автоматического деплоя приложения

23) Прочитайте текст и выберите один правильный ответ.

Какие показатели эффективности следует учитывать при разработке алгоритма для обработки потоковых данных с датчиков?

1. Задержка обработки, пропускная способность, устойчивость к шуму
2. Количество строк кода без учета качества работы
3. Количество функций без тестирования их эффективности
4. Объем используемой памяти без оценки скорости работы

24) Прочитайте текст и выберите один правильный ответ.

Как обеспечить поддержку изменений в бизнес-логике при проектировании алгоритмов?

1. Выносить часто меняющиеся правила в конфигурационные файлы или отдельные модули
2. Жестко встраивать все правила в код алгоритма
3. Обновлять ТЗ только после завершения проекта
4. Не фиксировать изменения — это не важно

25) Прочитайте текст и выберите один правильный ответ.

Почему важно проводить ревью (ревью коллегами) алгоритма перед его реализацией согласно ТЗ?

1. Для выявления логических ошибок, неоптимальных решений и повышения общего качества проекта
2. Чтобы ускорить процесс разработки за счет пропуска этапов проверки
3. Для сокращения времени тестирования после релиза
4. Чтобы сразу перейти к деплою без проверки логики

Проверяемая компетенция - ПК 1.2.

Блок заданий закрытого типа

1) Прочитайте текст и выберите один правильный ответ.

Какой этап является первым при реализации модуля по готовому техническому заданию?

1. Непосредственное написание кода
2. Подготовка среды разработки и настройка проекта
3. Тестирование на реальном устройстве
4. Оптимизация кода

2) Прочитайте текст и выберите один правильный ответ.

Что включает в себя подготовительный этап при реализации модуля?

1. Написание кода и отладка
2. Изучение ТЗ, выбор технологического стека, проектирование архитектуры модуля
3. Внедрение в мобильное приложение
4. Обучение модели ИИ

3) Прочитайте текст и выберите один правильный ответ.

Какой документ служит основным источником требований для реализации модуля?

1. Маркетинговое исследование
2. Техническое задание (ТЗ)
3. Отчет о тестировании конкурентного продукта
4. Руководство по стилю кода

4) Прочитайте текст и выберите один правильный ответ.

Что необходимо учитывать при реализации модуля для работы в фоновом режиме на мобильном устройстве?

1. Ограничения платформы на фоновую активность и энергопотребление
2. Только дизайн интерфейса
3. Только поддержку сетевых протоколов
4. Только интеграцию с облачными сервисами

5) Прочитайте текст и выберите один правильный ответ.

Какой из следующих методов наиболее подходит для реализации отзывчивого интерфейса в модуле?

1. Выполнение длительных операций в основном потоке
2. Использование асинхронного программирования и вынесение тяжелых задач в фоновые потоки
3. Использование только синхронных вызовов
4. Блокировка пользовательского интерфейса на время вычислений

6) Прочитайте текст и выберите один правильный ответ.

Что важно сделать после завершения кодирования модуля?

1. Провести модульное и интеграционное тестирование согласно ТЗ
2. Немедленно публиковать модуль в магазине приложений без проверки
3. Удалить исходный код после компиляции
4. Игнорировать отзывы тестировщиков на этапе разработки

7) Прочитайте текст и выберите один правильный ответ.

Какие действия необходимы для обеспечения соответствия кода модуля техническому заданию?

1. Постоянное сверение реализуемого функционала с пунктами ТЗ и тест-кейсами
2. Игнорирование требований, чтобы ускорить процесс разработки
3. Разработка без учета документации и требований заказчика

4. Внедрение случайных функций без согласования с ТЗ

8) Прочитайте текст и выберите один правильный ответ.

Почему важно использовать систему контроля версий при реализации программного модуля?

1. Для отслеживания изменений, возможности отката и командной работы
2. Для скрытия истории изменений от заказчика
3. Для уменьшения объема кода
4. Для ускорения компиляции

9) Прочитайте текст и выберите один правильный ответ.

Что такое "интерфейс модуля" в контексте разработки?

1. Набор публичных методов, классов и констант, которые предоставляет модуль для взаимодействия
2. Графический интерфейс пользователя
3. Внутренняя реализация функций модуля
4. Документация для конечных пользователей

10) Прочитайте текст и выберите один правильный ответ.

Какие критерии важны при реализации высоконагруженных модулей для мобильных приложений?

1. Производительность, эффективное использование памяти, отзывчивость
2. Только внешний вид интерфейса
3. Только скорость загрузки приложения
4. Только совместимость с устаревшими устройствами

11) Прочитайте текст и выберите один правильный ответ.

Как обеспечить качество реализуемого кода модуля согласно ТЗ?

1. Регулярное проведение код-ревью, написание unit-тестов, статический анализ кода
2. Игнорирование ошибок для быстрого релиза
3. Разработка без документации и тестов
4. Использование только автоматических средств без ручной проверки

12) Прочитайте текст и выберите один правильный ответ.

Что такое "паттерн проектирования" в контексте реализации модулей?

1. Типовое, проверенное решение часто встречающейся проблемы проектирования ПО
2. Случайно выбранный способ написания кода
3. Графический шаблон для интерфейса
4. Ошибка в архитектуре приложения

13) Прочитайте текст и выберите один правильный ответ.

Почему важно следовать принципам SOLID при реализации модуля?

1. Чтобы создать гибкий, понятный и легко поддерживаемый код
2. Чтобы снизить качество продукта ради скорости разработки
3. Чтобы уменьшить объем документации и упростить работу команды
4. Чтобы исключить необходимость тестирования после завершения разработки

14) Прочитайте текст и выберите один правильный ответ.

Какие инструменты помогают при отладке реализуемого программного модуля?

1. Отладчики (debugger), логирование (logging), профилировщики (profiler)
2. Только текстовые редакторы без специальных средств
3. Облачные хранилища без системы контроля изменений

4. Инструменты для маркетинга и продвижения приложения

15) Прочитайте текст и выберите один правильный ответ.

Что необходимо сделать перед сдачей модуля на интеграционное тестирование?

1. Провести внутреннее тестирование модуля и убедиться в выполнении всех пунктов ТЗ
2. Немедленно опубликовать модуль в магазине приложений
3. Удалить все комментарии из кода
4. Отключить все автоматические проверки качества кода

16) Прочитайте текст и выберите один правильный ответ.

Какие особенности следует учитывать при реализации модулей, работающих с геолокацией?

1. Разрешения пользователя, точность и энергоэффективность получения данных
2. Полная независимость от разрешений пользователя
3. Использование только максимальной точности без учета батареи
4. Минимальное использование системных API

17) Прочитайте текст и выберите один правильный ответ.

Как обеспечить повторяемость процесса сборки программного модуля?

1. Использование систем автоматической сборки (CI), скриптов (например, Gradle, Fastlane)
2. Ручная сборка каждым разработчиком на своей машине
3. Постоянное изменение конфигурации сборки
4. Сборка только на одном типе устройств

18) Прочитайте текст и выберите один правильный ответ.

Почему важно использовать соглашения об именовании в коде модуля?

1. Для повышения читаемости, легкости понимания кода и снижения количества ошибок
2. Чтобы усложнить понимание кода другими разработчиками
3. Для уменьшения объема документации проекта
4. Чтобы ускорить публикацию приложения за счет скрытия ошибок

19) Прочитайте текст и выберите один правильный ответ.

Какие действия следует предпринять после завершения кодирования модуля перед его слиянием в основную ветку?

1. Провести финальное тестирование, подготовить документацию и выполнить ревью кода
2. Немедленно сливать код без проверки качества
3. Удалить все комментарии из кода для уменьшения размера файла
4. Отключить все автоматические системы контроля качества

20) Прочитайте текст и выберите один правильный ответ.

Как можно обеспечить безопасное хранение данных внутри модуля?

1. Использование шифрования для конфиденциальных данных, безопасных API для хранения
2. Хранение всех данных открыто в shared preferences
3. Передача всех данных через публичные сети без шифрования
4. Не использовать аутентификацию пользователей

21) Прочитайте текст и выберите один правильный ответ.

Какой подход наиболее эффективен для реализации кеширования данных в модуле?

1. Использование стратегий кеширования (LRU), учет времени жизни данных

2. Сохранение всех данных в кеше навсегда без ограничений
3. Отказ от кеширования для упрощения логики
4. Использование только облачных решений без локального кеша

22) Прочитайте текст и выберите один правильный ответ.

Что является ключевым при создании технической документации на реализованный модуль?

1. Описание API, примеры использования, ограничения и зависимости
2. Только список функций без описания
3. Общие рекомендации по использованию приложения без конкретных примеров
4. Инструкции по маркетингу и продвижению модуля после релиза

23) Прочитайте текст и выберите один правильный ответ.

Почему важно учитывать доступность (accessibility) при реализации интерфейсных модулей?

1. Чтобы обеспечить возможность использования приложения людьми с ограниченными возможностями
2. Чтобы усложнить взаимодействие и снизить использование модуля пользователями
3. Для уменьшения времени разработки за счет игнорирования accessibility
4. Чтобы сосредоточиться только на технических характеристиках без учета восприятия

24) Прочитайте текст и выберите один правильный ответ.

Какие меры необходимо предпринять для обеспечения обратной совместимости реализованного модуля?

1. Тестирование на старых версиях ОС, избегание несовместимых изменений в публичном API
2. Создание модуля только для последней версии ОС
3. Исключительно использование устаревших технологий, чтобы снизить риски
4. Не документировать изменения в API

25) Прочитайте текст и выберите один правильный ответ.

Как правильно организовать процесс работы над модулем в команде согласно ТЗ?

1. Распределить задачи между участниками, использовать code review, проводить регулярные синки
2. Работать индивидуально без координации и документации
3. Делать все самостоятельно без делегирования задач
4. Публиковать промежуточные версии без проверки и согласования с командой

Проверяемая компетенция - ПК 1.3.

Блок заданий закрытого типа

1) Прочитайте текст и выберите один правильный ответ.

Какое из следующих правил наиболее важно для именования функций согласно ТЗ?

1. Использовать аббревиатуры и короткие, непонятные имена
2. Имена должны отражать действие, которое выполняет функция, и следовать стилю проекта
3. Писать функции с названиями из одной буквы для ускорения разработки
4. Использовать одинаковые имена для разных функций

2) Прочитайте текст и выберите один правильный ответ.

Что должно быть обязательно включено в заголовочный комментарий к классу согласно ТЗ?

1. Только название класса
2. Описание назначения класса, его ответственности и пример использования
3. Инструкции по сборке проекта
4. Историю изменений кода

3) Прочитайте текст и выберите один правильный ответ.

Какой стиль оформления отступов рекомендуется использовать для соблюдения требований ТЗ?

1. Смешивать табы и пробелы по своему усмотрению
2. Следовать единому стилю проекта (только пробелы или только табы) и их количеству
3. Не использовать отступы вообще
4. Использовать случайное количество пробелов

4) Прочитайте текст и выберите один правильный ответ.

Почему важно оформлять код в соответствии с принятыми в компании стандартами?

1. Для обеспечения единообразия, что облегчает чтение и поддержку кода всей командой
2. Чтобы усложнить понимание кода сторонними разработчиками
3. Для уменьшения времени написания кода за счет игнорирования стандартов
4. Чтобы скрыть логику работы модуля

5) Прочитайте текст и выберите один правильный ответ.

Что рекомендуется делать с длинными строками кода согласно требованиям оформления?

1. Оставлять их как есть, независимо от длины
2. Разбивать на несколько строк для улучшения читаемости без горизонтальной прокрутки
3. Уменьшать размер шва в редакторе кода
4. Удалять часть кода, чтобы строка стала короче

6) Прочитайте текст и выберите один правильный ответ.

Как следует оформлять структуру данных (например, JSON) в коде в соответствии с ТЗ?

1. Записывать все в одну строку без пробелов
2. Использовать форматирование с отступами и переносами для наглядности
3. Не использовать структуры данных в коде
4. Все структуры данных выносить в отдельные нечитаемые бинарные файлы

7) Прочитайте текст и выберите один правильный ответ.

Какие рекомендации по оформлению импортов/включений заголовочных файлов соответствуют ТЗ?

1. Сортировать их в алфавитном порядке и группировать по типу (системные, сторонние, свои)
2. Импортировать все модули в случайном порядке
3. Не использовать импорты, если модуль используется один раз
4. Дублировать импорты в каждом файле

8) Прочитайте текст и выберите один правильный ответ.

В чем заключается важность использования единого стиля для оформления комментариев?

1. Обеспечить понятность и единообразие документации внутри кода согласно ТЗ
2. Сделать код менее понятным для сторонних разработчиков
3. Ускорить написание кода за счет игнорирования стандартов

4. Сделать код более запутанным для тестировщиков

9) Прочитайте текст и выберите один правильный ответ.

Как правильно оформлять вызов функции с большим количеством параметров?

1. Записывать все параметры в одну строку, даже если она очень длинная
2. Расположить каждый параметр на новой строке с выравниванием для ясности
3. Случайным образом разбивать параметры по строкам
4. Не использовать функции с большим количеством параметров

10) Прочитайте текст и выберите один правильный ответ.

Какие действия необходимо выполнить перед отправкой кода на ревью согласно требованиям оформления?

1. Проверить код с помощью линтера и убедиться в соответствии стилю проекта
2. Игнорировать оформление, главное — функциональность
3. Удалить все комментарии для уменьшения размера файла
4. Объединить все изменения в один большой коммит без проверки

11) Прочитайте текст и выберите один правильный ответ.

Почему важно документировать нетривиальные алгоритмические решения в коде в соответствии с ТЗ?

1. Для облегчения понимания сложной логики другими разработчиками и ускорения онбординга
2. Чтобы скрыть сложность реализации
3. Для уменьшения объема комментариев
4. Документировать не обязательно — главное работает

12) Прочитайте текст и выберите один правильный ответ.

Как следует оформлять перечисления (enum) в соответствии с требованиями ТЗ?

1. Использовать понятные имена для элементов, при необходимости добавлять комментарии
2. Называть элементы случайными буквами и цифрами
3. Не объявлять перечисления отдельно — использовать "магические числа"
4. Названия элементов должны быть короткими и непонятными

13) Прочитайте текст и выберите один правильный ответ.

Что рекомендуется делать при использовании сторонних API в коде?

1. Следовать рекомендациям по их использованию и оформлению согласно документации проекта или ТЗ
2. Использовать API произвольно без учета стандартов оформления
3. Не документировать использование сторонних API
4. Встраивать весь код работы с API прямо в основной класс без отдельной документации

14) Прочитайте текст и выберите один правильный ответ.

Какие правила оформления циклов (for, while) улучшают читаемость?

1. Использовать отступы для тела цикла, избегать сложных вложенных циклов
2. Писать все циклы в одну строку без переносов
3. Не использовать циклы вообще
4. Вложенные циклы делать максимально глубокими для гибкости

15) Прочитайте текст и выберите один правильный ответ.

Как оформить логирование событий в модуле согласно ТЗ?

1. Использовать единый формат логов, включающий уровень важности, метку времени и понятное сообщение
2. Логировать события произвольным образом без формата
3. Логировать только ошибки без контекста
4. Не использовать логирование вообще

16) Прочитайте текст и выберите один правильный ответ.

Почему важно соблюдать требования к максимальной длине файла или функции?

1. Для обеспечения удобства чтения, навигации и понимания кода
2. Чтобы усложнить работу другим разработчикам
3. Для ускорения разработки за счет игнорирования стандартов
4. Чтобы скрыть качество реализации

17) Прочитайте текст и выберите один правильный ответ.

Какие инструменты помогают автоматизировать приведение кода в соответствие со стандартами оформления?

1. Форматтеры кода (например, Prettier, clang-format), линтеры (ESLint, Pylint)
2. Ручная проверка каждым участником команды
3. Использование текстовых редакторов без настроек проверки стиля
4. Игнорирование автоматических проверок

18) Прочитайте текст и выберите один правильный ответ.

Что нужно делать при обнаружении нарушения стиля кода после ревью?

1. Исправить стиль согласно установленным правилам проекта
2. Игнорировать замечания ради скорости разработки
3. Удалить часть кода из-за несоответствия стилю
4. Отклонить ревью и оставить как есть

19) Прочитайте текст и выберите один правильный ответ.

Какие особенности оформления рекомендуется учитывать при работе с конфигурацией сборки?

1. Документировать ключевые параметры и их влияние на сборку согласно ТЗ
2. Оформлять только основной код без документации конфигурации
3. Не комментировать параметры сборки для экономии времени
4. Игнорировать требования оформления при работе с конфигурацией

20) Прочитайте текст и выберите один правильный ответ.

Почему важно соблюдать требования оформления при работе с базами данных в мобильном приложении?

1. Для обеспечения понятности схемы данных, запросов и облегчения поддержки
2. Чтобы усложнить работу тестировщикам
3. Для уменьшения объема документации проекта
4. Оформление не имеет значения при работе с БД

21) Прочитайте текст и выберите один правильный ответ.

Как следует оформлять объявления констант, используемых в нескольких модулях?

1. Выносить их в отдельный файл с понятными именами и комментариями
2. Объявлять их в каждом файле, где они используются
3. Оставлять "магические числа" прямо в коде
4. Дублировать константы с разными значениями

22) Прочитайте текст и выберите один правильный ответ.

Какие правила оформления кода необходимо соблюдать при работе с callback-функциями?

1. Использовать отступы и комментарии для обозначения асинхронного потока, давать callback-ам понятные имена
2. Писать callback-и без комментариев и оформления, чтобы ускорить работу
3. Не использовать callback-и, чтобы упростить оформление кода
4. Вписывать все callback-и в одну строку без переносов

23) Прочитайте текст и выберите один правильный ответ.

Что важно учитывать при оформлении текстовых строк и ресурсов в соответствии с ТЗ?

1. Выносить строки в ресурсные файлы, использовать форматирование для многоязычности
2. Оставлять строки "захардкоженными" в коде без возможности перевода
3. Размещать все строки в одном файле без разделения по категориям
4. Не документировать назначение строковых ресурсов

24) Прочитайте текст и выберите один правильный ответ.

Как правильно оформлять работу с файловой системой согласно требованиям оформления кода?

1. Оборачивать операции ввода-вывода в отдельные функции/классы с комментариями и обработкой ошибок
2. Встраивать операции с файлами прямо в основной код без оформления
3. Не документировать файловые операции для экономии времени
4. Использовать файловые операции без обработки ошибок или логирования

25) Прочитайте текст и выберите один правильный ответ.

Почему важно соблюдать требования оформления при написании конфигурационных файлов для мобильных приложений?

1. Для повышения читаемости, простоты изменения конфигурации и избежания ошибок согласно ТЗ
2. Чтобы усложнить понимание конфигурации другим разработчикам
3. Для уменьшения объема конфигурационных файлов за счет игнорирования стандартов
4. Оформление конфигурационных файлов не имеет значения — главное их содержание

Проверяемая компетенция - ПК 1.4.

Блок заданий закрытого типа

1) Прочитайте текст и выберите один правильный ответ.

Какое из следующих преимуществ обеспечивает использование системы контроля версий в командной разработке?

1. Автоматическое исправление синтаксических ошибок в коде
2. Возможность параллельной работы над разными функциями и отслеживание истории изменений
3. Удаляет необходимость написания тестов
4. Автоматически генерирует код по описанию

2) Прочитайте текст и выберите один правильный ответ.

Какой из вариантов является основным назначением создания feature branch (ветки для функциональности)?

1. Удаление старых версий кода
2. Изоляция разработки новой функции от основной codebase для стабильности main ветки

3. Автоматическая компиляция проекта
4. Хранение бинарных файлов

3) Прочитайте текст и выберите один правильный ответ.

Что такое "fast-forward merge" в системах контроля версий?

1. Процесс объединения веток, при котором история основной ветки просто перемещается вперед к коммитам feature-ветки
2. Процесс удаления ветки после завершения работы над ней
3. Создание новой ветки для экспериментов
4. Процесс автоматической проверки кода на ошибки

4) Прочитайте текст и выберите один правильный ответ.

Почему важно писать осмысленные и атомарные коммиты?

1. Чтобы уменьшить размер репозитория
2. Для простоты понимания истории изменений и отката конкретных изменений при необходимости
3. Чтобы избежать конфликтов при слиянии веток
4. Для автоматической генерации документации

5) Прочитайте текст и выберите один правильный ответ.

Что такое "конфликт слияния" (merge conflict) при использовании системы контроля версий?

1. Ошибка компиляции программы
2. Ситуация, когда изменения в двух ветках затрагивают одни и те же строки кода и система не может автоматически их объединить
3. Устаревшая версия файла, которая должна быть удалена
4. Ошибка при запуске тестов

6) Прочитайте текст и выберите один правильный ответ.

Как лучше всего организовать workflow в системе контроля версий для проекта с непрерывной интеграцией?

1. Все работают напрямую в основной ветке (main)
2. Использовать модель GitFlow или аналогичную, с feature-ветками, которые сливаются в develop, а затем в main через pull request
3. Не использовать ветвление, чтобы избежать конфликтов
4. Работать только локально без синхронизации с центральным репозиторием

7) Прочитайте текст и выберите один правильный ответ.

Что такое "code review" в процессе pull request?

1. Запрос на получение последних обновлений из основной ветки
2. Процесс проверки кода другими членами команды на предмет качества, соответствия стандартам и выявления ошибок перед слиянием
3. Автоматическая проверка кода на наличие ошибок
4. Процесс удаления старых веток после завершения работы

8) Прочитайте текст и выберите один правильный ответ.

Почему важно использовать .gitignore файл?

1. Для автоматической генерации документации проекта
2. Чтобы исключить из репозитория файлы, которые не должны отслеживаться (бинарники, кэш, логи)
3. Для уменьшения размера репозитория
4. Чтобы избежать необходимости писать комментарии к коду

9) Прочитайте текст и выберите один правильный ответ.

Какая команда обычно используется для создания новой ветки в Git?

1. git commit
2. git push
3. git branch <name> или git checkout -b <name>
4. git clone

10) Прочитайте текст и выберите один правильный ответ.

Что означает термин "origin" в системах контроля версий, таких как Git?

1. Имя по умолчанию для удаленного репозитория, с которым осуществляется синхронизация
2. Объединение двух репозиториях
3. Восстановление предыдущей версии файла
4. Удаление ветки проекта

11) Прочитайте текст и выберите один правильный ответ.

Какой инструмент позволяет автоматически собирать и тестировать код при каждом пуше в репозиторий?

1. Системы непрерывной интеграции (CI/CD), например, Jenkins, GitHub Actions
2. Ручное ревью (manual review)
3. Форматтер кода (code formatter)
4. Откат версии (version rollback)

12) Прочитайте текст и выберите один правильный ответ.

Что такое "cherry-pick" в системе контроля версий Git?

1. Перенос одного конкретного коммита из одной ветки в другую
2. Создание новой ветки для разработки новых функций
3. Объединение двух веток без сохранения истории изменений
4. Удаление старых коммитов из истории

13) Прочитайте текст и выберите один правильный ответ.

Почему рекомендуется создавать pull request (PR) даже для небольших изменений?

1. Это обеспечивает видимость изменений, возможность обсуждения и контроля качества перед слиянием
2. Она автоматически удаляет все локальные изменения.
3. Она вызывает конфликты при каждом использовании.
4. Она не совместима со всеми системами контроля версий.

14) Прочитайте текст и выберите один правильный ответ.

Какие преимущества дает использование системы контроля версий при работе с экспериментальными функциями ИИ?

1. Позволяет безопасно экспериментировать в изолированных ветках, не затрагивая стабильную версию, и сравнивать результаты
2. Автоматически обучает модели без участия человека
3. Удаляет необходимость тестирования моделей
4. Обеспечивает только хранение данных без учета кода

15) Прочитайте текст и выберите один правильный ответ.

Что такое "release branch"?

1. Ветка, предназначенная для подготовки новой версии продукта к выпуску, куда вносят только исправления багов

2. Основная рабочая ветка проекта
3. Ветка, содержащая только стабильную версию продукта
4. Ветка, которая автоматически создается при каждом коммите

16) Прочитайте текст и выберите один правильный ответ.

Как правильно обновлять свою feature-ветку, если основная ветка (main) ушла вперед?

1. Сделать merge main в свою ветку или rebase своей ветки на актуальную main
2. Удалить свою ветку и начать заново
3. Прекращать работу над функцией до исчезновения конфликтов
4. Использовать только автоматические инструменты без ручного вмешательства

17) Прочитайте текст и выберите один правильный ответ.

Что такое "Distributed Version Control System" (распределенная система контроля версий)?

1. Система, где каждый разработчик имеет полную копию репозитория с всей историей
2. Система, где все работают только на центральном сервере без локальных копий
3. Система, которая автоматически создает резервные копии данных
4. Система, предназначенная только для хранения документации

18) Прочитайте текст и выберите один правильный ответ.

Какие платформы популярны для хостинга Git-репозитория и совместной работы?

1. GitHub, GitLab, Bitbucket
2. Microsoft Word, Excel
3. Google Docs
4. JIRA без интеграции с системой контроля версий

19) Прочитайте текст и выберите один правильный ответ.

Почему важно делать git pull перед началом работы над новой задачей?

1. Чтобы обеспечить актуальность локальной копии и минимизировать будущие конфликты
2. Чтобы уменьшить размер локального репозитория
3. Чтобы автоматически исправлять ошибки в коде
4. Для ускорения выполнения программы на мобильных устройствах

20) Прочитайте текст и выберите один правильный ответ.

Что означает термин "HEAD" в системе контроля версий Git?

1. Указатель на текущий активный коммит в рабочей директории
2. Обозначение конфликтных файлов
3. Создание новой ветки для экспериментов
4. Удаление старых коммитов из истории проекта

21) Прочитайте текст и выберите один правильный ответ.

Какие действия необходимо выполнить, чтобы предложить исправление в чужой открытый проект на GitHub?

1. Сделать fork репозитория, создать ветку с исправлением, сделать PR из своего форка
2. Удалить все старые ветки проекта
3. Написать письмо автору проекта
4. Закоммитить исправления прямо в чужой репозиторий

22) Прочитайте текст и выберите один правильный ответ.

Какое правило рекомендуется соблюдать при именовании веток в системе контроля версий?

1. Использовать понятные имена, отражающие суть задачи (например, feature/user-auth, bugfix/fix-crash)
 2. Давать веткам случайные имена
 3. Использовать только номера из трекера задач без пояснений
 4. Использовать одну и ту же ветку для всех задач
- 23) Прочитайте текст и выберите один правильный ответ.

В чем заключается преимущество использования системы контроля версий при разработке мобильных приложений?

1. Позволяет легко управлять разными версиями приложения для iOS и Android, отслеживать историю изменений кроссплатформенного кода
2. Автоматически публикует приложение в магазины
3. Удаляет необходимость тестирования на разных устройствах
4. Обеспечивает только хранение исходного кода без учета ресурсов

24) Прочитайте текст и выберите один правильный ответ.

Какая команда обычно используется для получения изменений из удаленного репозитория без автоматического слияния?

1. git push
2. git pull
3. git clone
4. git fetch

25) Прочитайте текст и выберите один правильный ответ.

Почему важно использовать аннотированные теги (annotated tags) для релизов?

1. Они хранят дополнительную информацию (автор, дата, сообщение), что полезно для отметки важных версий
2. Для автоматического исправления ошибок
3. Для скрытия внутренней информации о проекте
4. Для ускорения выполнения программы

Проверяемая компетенция - ПК 1.1.

Блок заданий открытого типа

1. Прочитайте текст. Дайте краткий ответ.

Опишите основные этапы проектирования алгоритма для системы рекомендаций в мобильном приложении.

2. Прочитайте текст. Дайте краткий ответ.

Какие факторы необходимо учитывать при выборе алгоритма машинного обучения для задачи классификации изображений на мобильном устройстве?

3. Прочитайте текст. Дайте краткий ответ.

Объясните, как оценить вычислительную сложность предложенного алгоритма сортировки данных, собираемых с сенсоров телефона.

4. Прочитайте текст. Дайте краткий ответ.

Опишите, как можно оптимизировать алгоритм поиска по геолокации для работы в условиях слабого интернет-соединения.

5. Прочитайте текст. Дайте краткий ответ.

Какие методы можно применить для обеспечения конфиденциальности данных пользователя при проектировании алгоритма аналитики?

6. Прочитайте текст. Дайте краткий ответ.

Как реализовать отказоустойчивый алгоритм обработки платежей в мобильном приложении?

7. Прочитайте текст. Дайте краткий ответ.
Какие техники можно применить для уменьшения времени обучения модели ИИ непосредственно на устройстве?
8. Прочитайте текст. Дайте краткий ответ.
Объясните, как спроектировать алгоритм для плавной анимации, учитывающий частоту кадров устройства.
9. Прочитайте текст. Дайте краткий ответ.
Какие метрики вы бы использовали для оценки эффективности алгоритма сжатия изображений в приложении?
10. Прочитайте текст. Дайте краткий ответ.
Опишите, как спроектировать алгоритм интеллектуального кэширования данных для оффлайн-работы приложения.
11. Прочитайте текст. Дайте краткий ответ.
Какие особенности необходимо учитывать при проектировании алгоритмов для работы с данными камеры в реальном времени?
12. Прочитайте текст. Дайте краткий ответ.
Объясните, как реализовать алгоритм адаптивной подстройки качества потокового видео в зависимости от скорости сети.
13. Прочитайте текст. Дайте краткий ответ.
Какие подходы к обработке ошибок и восстановлению важны при проектировании алгоритмов синхронизации данных?
14. Прочитайте текст. Дайте краткий ответ.
Как реализовать алгоритм предсказания действий пользователя для ускорения работы интерфейса?
15. Прочитайте текст. Дайте краткий ответ.
Опишите, каким образом можно спроектировать алгоритм для обнаружения аномалий в поведении пользователя в целях безопасности.
16. Прочитайте текст. Дайте краткий ответ.
Какие меры безопасности необходимо закладывать в алгоритм аутентификации по биометрии?
17. Прочитайте текст. Дайте краткий ответ.
Объясните, как спроектировать алгоритм автоматического обновления контента в приложении с учетом приоритетов данных.
18. Прочитайте текст. Дайте краткий ответ.
Какие методы уменьшения влияния алгоритма на батарею устройства вы бы предложили?
19. Прочитайте текст. Дайте краткий ответ.
Опишите подходы к тестированию и валидации алгоритма перед его интеграцией в приложение.
20. Прочитайте текст. Дайте краткий ответ.
Как обеспечить возможность кастомизации алгоритма под разных пользователей без потери производительности?
21. Прочитайте текст. Дайте краткий ответ.
Объясните, как спроектировать алгоритм для коллективного использования приложения (multi-user) с конфликтующими операциями.
22. Прочитайте текст. Дайте краткий ответ.
Какие особенности проектирования алгоритмов связаны с использованием AR (дополненной реальности) на мобильных устройствах?
23. Прочитайте текст. Дайте краткий ответ.
Как реализовать алгоритм, способный динамически подстраивать точность вычислений ИИ в зависимости от доступных ресурсов?

24. Прочитайте текст. Дайте краткий ответ.

Опишите подходы к обеспечению согласованности данных (data consistency) в распределенном алгоритме для мобильного приложения.

25. Прочитайте текст. Дайте краткий ответ.

Какие основные вызовы возникают при проектировании алгоритмов для работы с графовыми данными на мобильном устройстве и как их можно преодолеть?

Проверяемая компетенция - ПК 1.2.

Блок заданий открытого типа

1. Прочитайте текст. Дайте краткий ответ.

Как вы подходите к выбору архитектуры (например, MVVM, Clean Architecture) для нового модуля мобильного приложения?

2. Прочитайте текст. Дайте краткий ответ.

Какие критерии вы используете для выбора сторонних библиотек при реализации функционала модуля?

3. Прочитайте текст. Дайте краткий ответ.

Как обеспечить тестируемость кода модуля на этапе его реализации?

4. Прочитайте текст. Дайте краткий ответ.

Какие методы и инструменты вы используете для отладки производительности реализуемого модуля?

5. Прочитайте текст. Дайте краткий ответ.

Как вы организуете код модуля, чтобы обеспечить его легкую замену или обновление в будущем?

6. Прочитайте текст. Дайте краткий ответ.

Какие подходы к обработке состояний (state management) в модуле вы применяете и почему?

7. Прочитайте текст. Дайте краткий ответ.

Как обеспечить корректную работу модуля при повороте экрана или других изменениях конфигурации?

8. Прочитайте текст. Дайте краткий ответ.

Какие критерии вы считаете ключевыми для оценки качества реализованного модуля?

9. Прочитайте текст. Дайте краткий ответ.

Как вы реализуете локальное кэширование данных в модуле для ускорения работы и оффлайн-доступа?

10. Прочитайте текст. Дайте краткий ответ.

Какие особенности учета ограничений памяти необходимо учитывать при реализации модуля с большими объемами данных?

11. Прочитайте текст. Дайте краткий ответ.

Как обеспечить безопасное хранение ключей и паролей внутри модуля?

12. Прочитайте текст. Дайте краткий ответ.

Какие методы тестирования (unit, integration, UI) вы применяете для разных частей модуля и почему?

13. Прочитайте текст. Дайте краткий ответ.

Как организовать взаимодействие вашего модуля с другими модулями приложения с минимальной связностью?

14. Прочитайте текст. Дайте краткий ответ.

Какие подходы к оптимизации размера модуля (уменьшение APK/IPA) вы используете?

15. Прочитайте текст. Дайте краткий ответ.

Как учитывать различные разрешения экрана и плотности пикселей при реализации UI-модуля?

16. Прочитайте текст. Дайте краткий ответ.
Какие особенности реализации модулей, использующих нативные API (Android/iOS), вы выделяете?
17. Прочитайте текст. Дайте краткий ответ.
Как реализовать корректную обработку жизненного цикла (lifecycle) в вашем модуле?
18. Прочитайте текст. Дайте краткий ответ.
Какие меры по обеспечению обратной совместимости модуля с предыдущими версиями ОС вы предусматриваете?
19. Прочитайте текст. Дайте краткий ответ.
Как вы планируете процесс обновления модуля, если изменился его API или структура данных?
20. Прочитайте текст. Дайте краткий ответ.
Какие сложности могут возникнуть при реализации модуля, требующего частого обновления UI из фоновых потоков?
21. Прочитайте текст. Дайте краткий ответ.
Как обеспечить корректную работу модуля в условиях плохой сети или ее отсутствия?
22. Прочитайте текст. Дайте краткий ответ.
Какие критерии приемки модуля вы считаете наиболее важными, кроме соответствия ТЗ?
23. Прочитайте текст. Дайте краткий ответ.
Как организовать документацию по API модуля для других разработчиков?
24. Прочитайте текст. Дайте краткий ответ.
Каким образом можно автоматизировать сборку и тестирование модуля в рамках CI/CD?
25. Прочитайте текст. Дайте краткий ответ.
Какие лучшие практики реализации модулей для кроссплатформенной разработки (KMM, Flutter, React Native) вы можете выделить?

Проверяемая компетенция - ПК 1.3.

Блок заданий открытого типа

1. Прочитайте текст. Дайте краткий ответ.
Какие основные принципы именования переменных и функций вы применяете для обеспечения читаемости кода?
2. Прочитайте текст. Дайте краткий ответ.
Как вы обеспечиваете единообразие в оформлении отступов и переносов строк в команде?
3. Прочитайте текст. Дайте краткий ответ.
Какие стандарты кодирования (Google, Airbnb, внутренние) вы используете и почему?
4. Прочитайте текст. Дайте краткий ответ.
Как оформление кода влияет на скорость разработки и количество ошибок?
5. Прочитайте текст. Дайте краткий ответ.
Какие методы документирования сложной бизнес-логики в коде вы применяете?
6. Прочитайте текст. Дайте краткий ответ.
Как структурировать код модуля, чтобы его было легко navigate и понимать новым разработчикам?
7. Прочитайте текст. Дайте краткий ответ.
Какие подходы к комментированию кода вы используете, чтобы комментарии были полезны, а не избыточны?
8. Прочитайте текст. Дайте краткий ответ.
Как оформлять интерфейсы (API) модуля, чтобы их использование было интуитивно понятным?

9. Прочитайте текст. Дайте краткий ответ.
Каковы особенности оформления кода, связанного с многопоточностью и асинхронными операциями?
10. Прочитайте текст. Дайте краткий ответ.
Как соблюдать единый стиль в смешанных проектах (например, Kotlin/Java в Android)?
11. Прочитайте текст. Дайте краткий ответ.
Какие инструменты автоматической проверки стиля кода вы встраиваете в процесс разработки?
12. Прочитайте текст. Дайте краткий ответ.
Как оформлять обработку ошибок, чтобы это было единообразно и информативно для разработчика?
13. Прочитайте текст. Дайте краткий ответ.
Какие рекомендации по именованию файлов и организации структуры папок в проекте вы можете дать?
14. Прочитайте текст. Дайте краткий ответ.
Как структурировать и оформлять константы и перечисления (enums) в проекте?
15. Прочитайте текст. Дайте краткий ответ.
Какие подходы к рефакторингу помогают улучшить читаемость без изменения функциональности?
16. Прочитайте текст. Дайте краткий ответ.
Как оформлять экспериментальный или устаревший (deprecated) код?
17. Прочитайте текст. Дайте краткий ответ.
Какие особенности оформления длинных выражений или условий для улучшения читаемости?
18. Прочитайте текст. Дайте краткий ответ.
Как обеспечить, чтобы все члены команды соблюдали соглашения по оформлению кода?
19. Прочитайте текст. Дайте краткий ответ.
Какие практики оформления тестов (именование, структура) вы используете?
20. Прочитайте текст. Дайте краткий ответ.
Как оформлять конфигурационные файлы (JSON, YAML, XML) для их лучшей читаемости и поддерживаемости?
21. Прочитайте текст. Дайте краткий ответ.
Какие подходы к оформлению логов и отладочной информации вы считаете эффективными?
22. Прочитайте текст. Дайте краткий ответ.
Как документировать архитектурные решения и выбор технологий непосредственно в коде или рядом с ним?
23. Прочитайте текст. Дайте краткий ответ.
Какие частые ошибки в оформлении кода мешают его пониманию и поддержке?
24. Прочитайте текст. Дайте краткий ответ.
Как оформление кода влияет на эффективность код-ревью?
25. Прочитайте текст. Дайте краткий ответ.
Какие лучшие практики оформления кода вы считаете наиболее критичными для проектов, которые будут поддерживаться годами?

Проверяемая компетенция - ПК 1.4.

Блок заданий открытого типа

1. Прочитайте текст. Дайте краткий ответ.
Какие основные преимущества использования Git по сравнению с централизованными системами контроля версий (например, SVN)?

2. Прочитайте текст. Дайте краткий ответ.
Как выбрать стратегию ветвления (git-flow, trunk-based development) для проекта мобильной разработки?
3. Прочитайте текст. Дайте краткий ответ.
Какие практики коммитинга (частота, размер, сообщения) способствуют чистой истории проекта?
4. Прочитайте текст. Дайте краткий ответ.
Как организовать процесс code review в команде с использованием pull/merge requests?
5. Прочитайте текст. Дайте краткий ответ.
Какие инструменты интеграции с CI/CD вы используете и как они помогают автоматизировать проверки?
6. Прочитайте текст. Дайте краткий ответ.
Как обеспечить безопасность репозитория (управление доступом, сканирование секретов)?
7. Прочитайте текст. Дайте краткий ответ.
Какие подходы к ведению CHANGELOG и семантическому версионированию (SemVer) вы применяете?
8. Прочитайте текст. Дайте краткий ответ.
Как организовать работу над хотфиксами (hotfix) в рамках выбранной модели ветвления?
9. Прочитайте текст. Дайте краткий ответ.
Какие рекомендации по разрешению конфликтов слияния вы можете дать?
10. Прочитайте текст. Дайте краткий ответ.
Как использовать Git для управления задачами и трекинг прогресса (связь коммитов с задачами)?
11. Прочитайте текст. Дайте краткий ответ.
Какие методы защиты основных веток (main, develop) от прямых пушей вы используете?
12. Прочитайте текст. Дайте краткий ответ.
Как организовать хранение и версионирование больших файлов (assets, модели) в Git?
13. Прочитайте текст. Дайте краткий ответ.
Какие особенности использования Git в командах с разным уровнем опыта?
14. Прочитайте текст. Дайте краткий ответ.
Как эффективно использовать теги (tags) для отметки релизов и важных точек в истории?
15. Прочитайте текст. Дайте краткий ответ.
Какие инструменты визуализации истории Git вы используете для анализа проекта?
16. Прочитайте текст. Дайте краткий ответ.
Как настроить pre-commit hooks и другие автоматические проверки для поддержания качества кода?
17. Прочитайте текст. Дайте краткий ответ.
Какие подходы к бэкапу и восстановлению репозитория в случае сбоев?
18. Прочитайте текст. Дайте краткий ответ.
Как организовать работу с несколькими репозиториями (remote) в одном проекте?
19. Прочитайте текст. Дайте краткий ответ.
Какие сложности возникают при слиянии долгоживущих feature-веток и как их минимизировать?
20. Прочитайте текст. Дайте краткий ответ.
Как внедрить и поддерживать соглашения по именованию веток и коммитов в команде?
21. Прочитайте текст. Дайте краткий ответ.
Какие практики использования Git submodules или package management вы считаете эффективными?
22. Прочитайте текст. Дайте краткий ответ.
Как организовать обучение команды эффективной работе с Git?

23. Прочитайте текст. Дайте краткий ответ.

Какие best practices по организации .gitignore файла для мобильных проектов?

24. Прочитайте текст. Дайте краткий ответ.

Как интегрировать систему контроля версий с трекером задач (Jira, YouTrack) для автоматизации?

25. Прочитайте текст. Дайте краткий ответ.

Какие особенности использования Git в распределенных командах разработки мобильных приложений?