

Ассоциация научно-технических организаций "Уральский профессиональный форум"
Автономная некоммерческая профессиональная образовательная организация
"Современный цифровой колледж при Западно-Уральском институте экономики и
права" (АНПОО "СЦК при ЗУИЭП")

РАССМОТРЕНО
на заседании Педагогического совета
протокол № 7 от «18» мая 2026 г.

УТВЕРЖДАЮ:
Директор АНПОО СЦК при ЗУИЭП"

_____ А.С. Волков
«18» мая 2026 г.



**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ
И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
МДК.01.02 Разработка мобильных приложений с поддержкой искусственного
интеллекта**

по специальности

09.02.13 Интеграция решений с применением технологий искусственного интеллекта
квалификация «Специалист по работе с искусственным интеллектом»

форма обучения: очная

форма промежуточной аттестации: экзамен

Пермь 2026

СОДЕРЖАНИЕ

1. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ.....	4
2. КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ.....	7
3. КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	14
4. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ.....	23

1. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ

1.1. Общая характеристика оценочных материалов.

Оценочные материалы предназначен для контроля и оценки результатов освоения междисциплинарного курса МДК.01.02 Разработка мобильных приложений с поддержкой искусственного интеллекта, входящего в ПМ.01 Разработка кода для искусственного интеллекта специальности СПО 09.02.13 Интеграция решений с применением технологий искусственного интеллекта.

Оценочные материалы состоит из оценочных материалов текущего контроля и контрольно-измерительных материалов промежуточной аттестации.

Характеристика текущего контроля

Целью текущего контроля является контроль запланированных по дисциплине знаний и умений. Текущий контроль по дисциплине проводится в следующих формах: тестирование, практическое занятие и других формах.

Текущий контроль проводится в соответствии с рабочей программой по дисциплине и структурных элементов электронного курса в СДО. Периодичность проведения текущего контроля зависит от сроков выполнения заданий обучающимися в пределах установленного учебным планом семестра.

Текущий контроль может проводиться с помощью электронного обучения и дистанционных образовательных технологий. Текущий контроль включает в себя оценку выполнения практической работы и тестирования по каждому разделу тематического плана учебной дисциплины. Порядок проведения текущего контроля определяется оценочными средствами.

Характеристика промежуточной аттестации

Целью контроля промежуточной аттестации является контроль освоения запланированных по дисциплине знаний и умений. Форма промежуточной аттестации: экзамен. Промежуточная аттестация может проводиться с помощью электронного обучения и дистанционных образовательных технологий.

Контрольно-измерительные материалы промежуточной аттестации состоят из:

1) теоретической части, которая выполняется письменно при проведении промежуточной аттестации обучающихся с применением дистанционных образовательных технологий, представляет собой ответ на теоретический вопрос по программе курса;

2) тестовой части, которая выполняется письменно и включает в себя:

- задание типа А (базовый уровень) - 10 тестовых вопросов с 4 вариантами ответов;
- задание типа В (задания повышенной сложности) - 2 задания с 4 вариантами ответов;
- задание типа С (практико-ориентированное комплексное задание) - 1 задание с 4 вариантами ответов.

Первая часть (А) теста включает 10 заданий базового уровня сложности. Данные задания направлены на выявление сформированности полученных в ходе изучения дисциплины базовых знаний, соответствующих заданным стандартом компетенциям. Каждое тестовое задание данной части содержит 4 варианта ответов, один или несколько из которых является правильным.

Вторая часть (В) теста состоит из 2 заданий повышенного уровня сложности. Данные задания направлены на выявление усвоения основных алгоритмов профессиональной деятельности, соответствующих заданным стандартом компетенциям.

Практическая часть (С) состоит из 1 практико-ориентированного задания повышенного уровня сложности (задачи, ситуации).

Для каждого варианта теста подготовлены эталоны ответов для проверки результатов теста.

Данные задания направлены на выявление уровня сформированности опыта практической деятельности, соответствующего заданным стандартам компетенциям, и выполняются с учетом смоделированной ситуации.

1.2. Требования к результатам освоения учебной дисциплины

Код ОК, ПК	Уметь	Знать
ОК.01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности
ОК.02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации, современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности в том числе с использованием цифровых средств
ОК.03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях	содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования; основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты
ОК.04	Эффективно взаимодействовать и работать в коллективе и команде	психологические основы деятельности коллектива,

		психологические особенности личности; основы проектной деятельности
ОК.05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	особенности социального и культурного контекста; правила оформления документов и построения устных сообщений
ОК.06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности; стандарты антикоррупционного поведения и последствия его нарушения
ОК.07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения; принципы бережливого производства; основные направления изменения климатических условий региона
ОК.08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения
ОК.09	Пользоваться профессиональной документацией на государственном и иностранном языках	правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности
ПК 1.1	Анализировать технические задания и выявлять требования к алгоритмам.	Основные методы и подходы к построению алгоритмов (типовые поисковые алгоритмы, жадные

	Применять методы алгоритмизации для решения задач программирования. Разрабатывать оптимальные алгоритмы для решения задач в области ИИ.	алгоритмы, динамическое программирование, рекурсивные подходы). Принципы эффективной обработки данных. Языки программирования, применяемые для разработки алгоритмов.
ПК 1.2	Реализовывать программные модули на основе требований технического задания. Соблюдать при разработке принципы «чистого кода». Использовать стандартные библиотеки и фреймворки для ускорения разработки.	Принципы модульного программирования. Языки программирования для разработки модулей. Стандартные фреймворки и библиотеки для работы с ИИ.
ПК 1.3	Оформлять код в соответствии с принятыми стандартами и требованиями. Документировать разработанный программный код. Соблюдать соглашения о наименованиях переменных, функций и классов (например, PEP8 для Python).	Основные принципы чистого кода (Clean Code). Стандарты и практики документирования программного обеспечения. Инструменты для автоматической проверки качества кода (например, PyLint, ESLint).
ПК 1.4	Работать с системами контроля версий для управления проектами. Организовывать совместную работу над проектом через ветки разработки и слияние изменений. Разрешать конфликты при слиянии кода.	Принципы работы распределенных систем контроля версий. Основные команды и операции в системах контроля версий (например: commit, pull, push, merge). Методы разрешения конфликтов в ходе групповой разработки.
ПК 1.5	Использовать инструменты для отладки программного кода. Идентифицировать и исправлять ошибки в программе. Применять методы логирования для анализа выполнения программ.	Принципы работы отладчиков и логирования. Способы выявления ошибок в программе (отладка по шагам, точки останова). Инструменты для отладки кода (например, PyCharm, Visual Studio Debugger).

2. КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

2.1 Примерная тематика практических заданий

Практическая работа № 1. Создание первого Android-приложения с базовыми интерфейсами

Цель: освоить базовые инструменты разработки Android-приложений в Android Studio, создать простое приложение с пользовательским интерфейсом и подключением к БД.

Инструменты: Android Studio, JDK 8, IntelliJ IDEA или NetBeans (для изучения кода), Microsoft SQL Server Express Edition + SSMS (для моделирования данных).

Задания:

1. Установите **Android Studio** и **JDK 8**. Создайте новый проект с шаблоном *Empty Activity*. Настройте эмулятор Android (AVD) для тестирования.

2. Разработайте интерфейс приложения с использованием XML-разметки.

Включите элементы:

- TextView с приветственным сообщением;
- EditText для ввода имени и фамилии пользователя;
- кнопку Button «Сохранить»;
- TextView для отображения статуса операции (успех/ошибка).

3. В **Microsoft SQL Server Express** с помощью **SSMS** создайте базу данных UserDB и таблицу Users со столбцами:

- id (INT, PRIMARY KEY, IDENTITY);
- name (NVARCHAR(50));
- surname (NVARCHAR(50)).

4. Реализуйте обработчик нажатия кнопки «Сохранить»:

- считывает данные из полей EditText;
- устанавливает соединение с SQL Server через **Microsoft SQL Server Java Connector**;
- выполняет INSERT-запрос для добавления данных в таблицу Users;
- отображает статус операции в TextView.

5. Добавьте валидацию ввода: если поля пустые, выведите сообщение об ошибке через Toast.

6. Протестируйте приложение на эмуляторе и реальном устройстве. Проверьте, что данные корректно добавляются в БД.

7. Составьте отчёт:

- скриншоты интерфейса и таблицы Users в SSMS с добавленными данными;
- фрагмент кода с логикой подключения к БД и INSERT-запросом;
- описание возникших проблем и способов их решения.

Практическая работа № 2. Внедрение TensorFlow Lite модели в Android-приложение

Цель: интегрировать обученную модель машинного обучения (TFLite) в Android-приложение, использовать её для предсказаний и сохранять результаты в БД.

Инструменты: Android Studio, JDK 8, TensorFlow Lite, MySQL Installer for Windows.

Задания:

1. Найдите модель TensorFlow Lite для классификации изображений (например, mobilenet_v1_1.0_224.tflite). Поместите файл в папку assets проекта.

2. Добавьте зависимости TensorFlow Lite в build.gradle модуля приложения:

```
gradle      ⇐ Переносить   ↗ Свернуть   📄 Копировать

implementation 'org.tensorflow:tensorflow-lite:0.0.0-nightly'
```

3. Создайте интерфейс с:
 - ImageView для изображения;
 - кнопкой «Загрузить из галереи»;
 - кнопкой «Распознать»;
 - TextView для результата (класс, уверенность).
4. Реализуйте загрузку изображения из галереи и его отображение в ImageView.
5. Напишите код для:
 - загрузки TFLite-модели через Interpreter;
 - предобработки изображения (изменение размера до 224×224, нормализация);
 - выполнения предсказания;
 - отображения результата в TextView.
6. Установите **MySQL Installer for Windows**, создайте БД RecognitionDB и таблицу Results:
 - id (INT, PK, AUTO_INCREMENT);
 - image_path (VARCHAR(255));
 - predicted_class (VARCHAR(100));
 - confidence (FLOAT);
 - timestamp (DATETIME).
7. Модифицируйте код: после распознавания сохраняйте путь к изображению, предсказанный класс и уверенность в таблицу Results через JDBC-драйвер MySQL.
8. Протестируйте приложение с 5–10 разными изображениями. Проверьте корректность предсказаний и записей в БД.
9. Подготовьте отчёт:
 - скриншоты работы приложения и таблицы Results в MySQL Workbench;
 - код загрузки модели и сохранения результатов в БД;
 - таблица с результатами (изображение, предсказание, уверенность, статус сохранения).

Практическая работа № 3. Разработка мобильного приложения для распознавания изображений

Цель: создать приложение для распознавания объектов в реальном времени с сохранением истории в БД и визуализацией через Visio.

Инструменты: Android Studio, JDK 8, CameraX API, TensorFlow Lite, Microsoft Visio Professional.

Задания:

1. Создайте проект в **Android Studio**. Настройте зависимости для CameraX и TensorFlow Lite.
2. Реализуйте интерфейс:
 - PreviewView для превью камеры;
 - кнопка «Сделать снимок»;
 - область отображения результатов (список объектов с вероятностями);
 - индикатор загрузки.
3. Настройте работу с камерой через CameraX:
 - запросите разрешения;
 - настройте превью;
 - реализуйте захват фото.
4. Интегрируйте TFLite-модель (например, mobilenet). Обработывайте захваченное фото.
5. Оптимизируйте обработку: измените размер изображения до 224×224, нормализуйте пиксели.
6. Создайте БД в **SQL Server Express** (таблица RecognitionHistory):
 - id, image_path, objects_detected (JSON или VARCHAR), timestamp.

7. Сохраняйте в БД: путь к фото и JSON-массив распознанных объектов с вероятностями.

8. В **Microsoft Visio**:

- создайте диаграмму Use Case для приложения (акторы: Пользователь; варианты: «Запустить камеру», «Сделать снимок», «Просмотреть историю»);
- создайте ER-диаграмму для RecognitionHistory.

9. Протестируйте приложение:

- 15–20 фото разных объектов;
- оцените скорость обработки (секунды на кадр);
- проверьте записи в БД.

10. Составьте отчёт:

- видео/скриншоты работы в реальном времени;
- код захвата фото и обработки моделью;
- диаграммы из Visio (Use Case, ER);
- анализ производительности (таблица: фото, объекты, время обработки, статус сохранения).

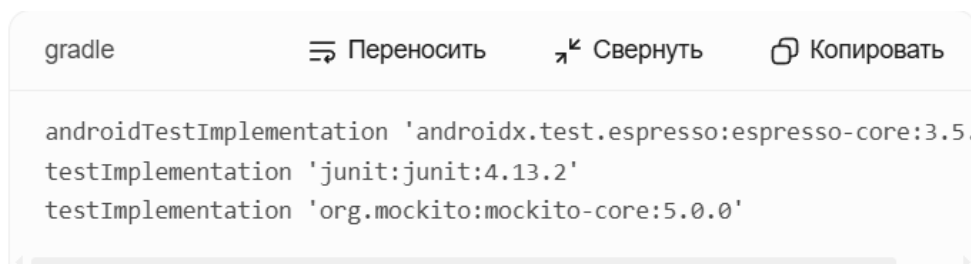
Практическая работа № 4. Автоматизация тестирования мобильного ИИ-приложения

Цель: написать UI- и модульные тесты для ИИ-приложения, проверить взаимодействие с БД, проанализировать покрытие кода.

Инструменты: Android Studio, JDK 8, Espresso, JUnit, Mockito, SQL Server Management Studio.

Задания:

1. В проекте из ПР № 5 настройте зависимости для тестирования:



```
gradle      ⇐ Переносить  ↗ Свернуть  📋 Копировать

androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.
testImplementation 'junit:junit:4.13.2'
testImplementation 'org.mockito:mockito-core:5.0.0'
```

2. Напишите UI-тесты (Espresso):

- тест превью камеры при запуске;
- тест захвата фото (проверка изменения состояния UI);
- тест отображения результатов после обработки.

3. Напишите модульные тесты (JUnit):

- загрузка модели (создание Interpreter без ошибок);
- предобработка изображения (функция изменения размера и нормализации);
- предсказание (мок-данные на входе, проверка формата выхода).

4. Используйте **Mockito** для моков камеры и модели.

5. Запустите тесты, проверьте покрытие кода (Code Coverage в Android Studio).

6. Внесите ошибку в логику обработки (например, неверный размер изображения). Запустите тесты — убедитесь, что хотя бы один падает.

7. В **SSMS** напишите SQL-запрос, проверяющий целостность данных в RecognitionHistory после тестов:

```
sql      ⇐ Переносить      ↗ Свернуть      📋 Копировать

SELECT COUNT(*) AS total_records,
        AVG(confidence) AS avg_confidence
FROM RecognitionHistory
WHERE timestamp > DATEADD(hour, -1, GETDATE());
```

8. Подготовьте отчёт:

- список тестов с описанием сценария;
- скриншот покрытия кода;
- вывод о надёжности набора тестов;
- результаты SQL-запроса (количество записей, средняя уверенность за последний час).

2.2 Задания в тестовой форме

Тестовые задания типа А (Необходимо выбрать один наиболее правильный ответ из нескольких предложенных вариантов)

- Какая платформа использует язык программирования Swift в качестве основного?
A) Android
B) Windows Phone
C) BlackBerry
D) iOS
- Что такое Activity в Android-приложении?
A) Виджет для отображения текста
B) Сервис для фоновых задач
C) База данных приложения
D) Основной экран/окно приложения
- Какой инструмент используется для создания и настройки эмулятора Android-устройства?
A) Git
B) AVD Manager
C) SQL Server
D) Visio
- В какой папке Android-проекта хранятся XML-файлы разметки интерфейса?
A) src/
B) manifests/
C) res/layout/
D) assets/
- Какой язык программирования Google официально рекомендует для разработки под Android?
A) Java
B) C++
C) Python
D) Kotlin
- Что делает метод setContentView() в Android?
A) Сохраняет данные в базу
B) Загружает XML-разметку в Activity
C) Обработывает нажатия кнопок

- D) Запускает новый поток
7. Какой компонент Android используется для отображения списка элементов?
- A) TextView
B) Button
C) RecyclerView
D) EditText
8. Что такое Material Design?
- A) Язык программирования
B) Система контроля версий
C) Тип базы данных
D) Набор рекомендаций Google по дизайну интерфейсов
9. Какой контейнер в Android позволяет гибко позиционировать элементы с помощью ограничений?
- A) LinearLayout
B) RelativeLayout
C) ConstraintLayout
D) FrameLayout
10. Как передать данные между Activity в Android?
- A) Через глобальные переменные
B) С помощью Intent
C) Через XML-файл
D) Через базу данных
11. Что такое TensorFlow Lite?
- A) Полная версия TensorFlow для серверов
B) Язык программирования для ИИ
C) Инструмент для веб-разработки
D) Облегчённая версия TensorFlow для мобильных устройств
12. В каком формате сохраняются модели для использования в TensorFlow Lite?
- A) .py
B) .xml
C) .tflite
D) .apk
13. Какой шаг НЕ входит в процесс внедрения TFLite-модели в Android-приложение?
- A) Размещение модели в папке assets/
B) Добавление зависимости TensorFlow Lite в build.gradle
C) Создание Interpreter для загрузки модели
D) Компиляция модели в .exe-файл
14. Что такое квантование модели в контексте TensorFlow Lite?
- A) Увеличение размера модели
B) Уменьшение точности весов для сокращения размера и ускорения работы
C) Добавление новых слоёв в модель
D) Конвертация модели в другой фреймворк
15. Какой API Android используется для работы с камерой в современных приложениях?
- A) Camera1
B) CameraX
C) OpenGL
D) MediaPlayer
16. Какой инструмент Android Studio помогает анализировать потребление CPU и памяти приложением?
- A) Logcat
B) Layout Inspector
C) Android Profiler

- D) AVD Manager
17. Что такое CI/CD в контексте мобильной разработки?
- A) Язык разметки интерфейса
- B) Тип базы данных
- C) Процесс непрерывной интеграции и доставки (автоматическая сборка и тестирование)**
- D) Метод шифрования данных
18. Какой файл содержит информацию о компонентах приложения, разрешениях и версии?
- A) build.gradle
- B) AndroidManifest.xml**
- C) strings.xml
- D) styles.xml
19. Что такое APK-файл?
- A) Исходный код приложения
- B) Готовый установочный пакет Android-приложения**
- C) Файл базы данных
- D) Конфигурационный файл
20. Какой сервис Google используется для публикации Android-приложений?
- A) Google Drive
- B) Google Docs
- C) Google Play Console**
- D) Google Analytics
21. Какой инструмент используется для контроля версий кода в командной разработке?
- A) Git**
- B) MySQL
- C) Visio
- D) SQL Server
22. Что такое отладка (debugging) приложения?
- A) Публикация приложения в магазине
- B) Оптимизация размера APK
- C) Поиск и исправление ошибок в коде**
- D) Создание дизайна интерфейса
23. Какой метод вызывается при создании Activity в Android?
- A) onPause()
- B) onCreate()**
- C) onDestroy()
- D) onStop()
24. Что такое ViewModel в архитектуре Android-приложений?
- A) Виджет интерфейса
- B) Тип анимации
- C) Класс для хранения данных, переживающих смену конфигурации**
- D) Сервис для работы с сетью
25. Какой формат данных часто используется для передачи информации между мобильным приложением и сервером?
- A) DOCX
- B) PNG
- C) JSON**
- D) MP3
26. Что такое REST API в контексте мобильных приложений?
- A) Тип базы данных
- B) Интерфейс для взаимодействия с сервером по HTTP-запросам**
- C) Язык программирования

- D) Инструмент дизайна
27. Какой метод используется для логирования сообщений в Android?
- A) print()
 - B) Log.d()**
 - C) debug()
 - D) message()
28. Что такое фрагментация Android-устройств?
- A) Ошибка в коде приложения
 - B) Разнообразие версий ОС, размеров экранов и характеристик устройств**
 - C) Процесс удаления приложения
 - D) Тип шифрования данных
29. Какой инструмент помогает создавать диаграммы Use Case и ER-диаграммы?
- A) Android Studio
 - B) Git
 - C) Microsoft Visio**
 - D) MySQL Installer
30. Что такое эмулятор Android?
- A) Реальное мобильное устройство
 - B) Программа для взлома приложений
 - C) Тип процессора
 - D) Программная имитация Android-устройства для тестирования**

3. КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

3.1 Вопросы для подготовки к экзамену

1. Опишите ключевые различия между платформами Android и iOS с точки зрения экосистемы, инструментов разработки и целевой аудитории.
2. Что такое жизненный цикл Android-приложения? Перечислите основные состояния активности (Activity) и кратко поясните каждое.
3. Назовите основные компоненты пользовательского интерфейса в Android и их назначение (например, Activity, Fragment, View).
4. Как происходит обработка событий пользовательского интерфейса в Android-приложении? Приведите пример обработки нажатия кнопки.
5. В чём преимущества и недостатки разработки под Android по сравнению с iOS?
6. Опишите процесс установки и настройки Android Studio для разработки мобильных приложений. Какие компоненты необходимо дополнительно установить?
7. Каковы основные элементы интерфейса Android Studio и их функции (Project, Design, Palette и т.д.)?
8. Объясните структуру типичного Android-проекта (папки app/, src/, res/, manifests/ и их содержимое).
9. Что такое XML-разметка в Android? Приведите пример простого интерфейса с кнопкой и текстовым полем, описанным в XML.
10. Как связать XML-разметку с кодом Kotlin/Java для обработки нажатий кнопок и изменения содержимого виджетов?
11. Сравните языки программирования Kotlin и Java для разработки под Android. Почему Kotlin считается предпочтительным языком?
12. Опишите основные синтаксические конструкции Kotlin (объявление переменных, функции, классы, условные операторы, циклы).
13. Что такое классы и объекты в Kotlin? Приведите пример создания класса и его экземпляра в контексте мобильного приложения.

14. Какие паттерны проектирования наиболее часто применяются в мобильной разработке (например, MVP, MVVM)? Кратко опишите один из них.
15. Как осуществляется отладка и тестирование кода в Android Studio? Опишите инструменты Logcat и Debugger.
16. Перечислите принципы проектирования интуитивно понятных интерфейсов для мобильных устройств. Что такое UX/UI и в чём разница?
17. Опишите основные компоненты UI в Android (кнопки, текстовые поля, списки, контейнеры) и их назначение.
18. Что такое Material Design? Приведите примеры компонентов и рекомендаций Google по дизайну.
19. Как создать адаптивный интерфейс, который корректно отображается на экранах разных размеров и ориентаций? Опишите использование ConstraintLayout и квалификаторов ресурсов.
20. Как организовать навигацию между экранами (Activity/Fragment) в Android-приложении? Кратко опишите использование Intent.
21. Что такое TensorFlow Lite? Каковы его преимущества для использования в мобильных приложениях по сравнению с полной версией TensorFlow?
22. Опишите процесс конвертации обученной модели TensorFlow в формат TensorFlow Lite (.tflite). Какие инструменты для этого используются?
23. Как внедрить предобученную TFLite-модель (например, для классификации изображений) в Android-приложение? Перечислите ключевые шаги.
24. Какие методы оптимизации моделей TFLite существуют для повышения производительности на мобильных устройствах (квантование, прунинг и т. д.)?
25. Как реализовать распознавание изображений в реальном времени с использованием камеры устройства и TFLite-модели? Опишите архитектуру решения.
26. Какие сенсоры мобильного устройства можно использовать для сбора данных для ИИ-приложений? Приведите 2–3 примера их применения.
27. Как минимизировать задержки при обработке данных в реальном времени в мобильном ИИ-приложении (например, при распознавании речи или жестов)?
28. В чём особенности взаимодействия пользователя с ИИ-функционалом в мобильном приложении? Приведите пример сценария использования голосового помощника.
29. Какие подходы и инструменты используются для автоматизированного тестирования мобильных приложений с ИИ-функционалом (Espresso, JUnit, Mockito)? Кратко опишите назначение каждого.
30. Как провести профилирование производительности мобильного ИИ-приложения (потребление CPU, RAM, батареи)? Какие инструменты Android Studio для этого существуют (Android Profiler)?

3.2 Тестовые задания для промежуточной аттестации

Тестовые задания типа А

1. Какой файл в Android-проекте определяет структуру и свойства виджетов интерфейса?
А) build.gradle
В) XML-файл разметки (например, activity_main.xml)
С) AndroidManifest.xml
D) strings.xml
2. Что произойдёт, если в AndroidManifest.xml не указать разрешение на использование интернета, но приложение попытается выполнить сетевой запрос?
А) Приложение автоматически запросит разрешение у пользователя
В) Запрос выполнится, но будет медленнее
С) Произойдёт ошибка (исключение SecurityException)

- D) Система выдаст предупреждение, но запрос выполнится
3. Какой метод вызывается, когда Activity уходит на задний план (пользователь переключается на другое приложение)?
- A) onCreate()
 - B) onResume()
 - C) onPause()**
 - D) onDestroy()
4. Для чего используется RecyclerView вместо ListView?
- A) Для отображения статических списков
 - B) Для более эффективной работы с большими списками за счёт переиспользования элементов**
 - C) Для создания анимаций
 - D) Для работы с базами данных
5. Какой инструмент Android Studio позволяет визуально редактировать интерфейс?
- A) Logcat
 - B) Terminal
 - C) Layout Editor**
 - D) Profiler
6. Что такое Gradle в контексте Android-разработки?
- A) Язык программирования
 - B) Система автоматизации сборки проектов**
 - C) База данных
 - D) Виджет интерфейса
7. Какой модификатор доступа в Kotlin делает свойство видимым только внутри класса?
- A) public
 - B) protected
 - C) private**
 - D) internal
8. Что означает аббревиатура MVP в контексте архитектуры мобильных приложений?
- A) Most Valuable Player
 - B) Model-View-Presenter**
 - C) Mobile View Protocol
 - D) Main View Process
9. Какой компонент Android используется для выполнения фоновых задач без пользовательского интерфейса?
- A) Activity
 - B) Service**
 - C) Fragment
 - D) View
10. Что такое «адаптивный макет» в мобильной разработке?
- A) Приложение, которое меняет цвет в зависимости от темы ОС
 - B) Интерфейс, который корректно отображается на экранах разных размеров и ориентаций**
 - C) Приложение с поддержкой нескольких языков
 - D) Приложение, автоматически обновляющее контент
11. Какой формат файла используется для хранения векторных изображений в Android?
- A) PNG
 - B) JPG
 - C) SVG (в виде Vector Drawable)**
 - D) BMP
12. Что такое «квантование» модели TensorFlow Lite?

- A) Увеличение размера модели для повышения точности
 - B) Уменьшение точности числовых значений (например, с 32-bit до 8-bit) для сокращения размера и ускорения работы**
 - C) Добавление новых слоёв в модель
 - D) Конвертация модели в другой фреймворк
13. Какой класс в TensorFlow Lite используется для загрузки и запуска модели?
- A) ModelLoader
 - B) Interpreter**
 - C) TensorFlow
 - D) Predictor
14. Что необходимо сделать перед использованием камеры в Android-приложении?
- A) Установить специальную библиотеку
 - B) Запросить разрешение у пользователя и проверить его статус**
 - C) Подключить устройство к Wi-Fi
 - D) Обновить ОС
15. Какой метод используется для обработки нажатия кнопки в Android?
- A) onClick()
 - B) Установка OnClickListener и реализация метода onClick()**
 - C) buttonPressed()
 - D) handleClick()
16. Что такое «Intent» в Android?
- A) Механизм для запуска Activity, Services и передачи данных между компонентами**
 - B) Тип базы данных
 - C) Вид анимации
 - D) Инструмент отладки
17. Какой инструмент используется для профилирования производительности Android-приложения (потребление CPU, памяти, батареи)?
- A) Layout Inspector
 - B) AVD Manager
 - C) Android Profiler**
 - D) Terminal
18. Что такое «CI/CD» в разработке мобильных приложений?
- A) Язык разметки интерфейса
 - B) Непрерывная интеграция и доставка (автоматическая сборка, тестирование и развёртывание)**
 - C) Тип базы данных
 - D) Метод шифрования
19. Какой файл содержит строковые ресурсы приложения (тексты кнопок, заголовки и т.д.)?
- A) styles.xml
 - B) strings.xml**
 - C) colors.xml
 - D) dimens.xml
20. Что такое «Fragment» в Android?
- A) Отдельный файл кода
 - B) Модульный компонент пользовательского интерфейса, который может быть частью Activity**
 - C) Тип анимации
 - D) Виджет для отображения текста
21. Какой метод вызывается при уничтожении Activity?
- A) onCreate()
 - B) onStart()

- C) onResume()
D) onDestroy()
22. Для чего используется SharedPreferences в Android?
A) Для хранения небольших объёмов данных в виде пар «ключ-значение»
B) Для работы с камерой
C) Для создания сетевых запросов
D) Для отображения уведомлений
23. Какой инструмент используется для управления версиями кода в командной разработке?
A) MySQL
B) Git
C) Visio
D) Gradle
24. Что такое «эмулятор Android»?
A) Реальное устройство
B) Программная имитация Android-устройства для тестирования приложений
C) Тип процессора
D) Сервис Google Play
25. Какой формат используется для публикации приложений в Google Play?
A) .apk
B) .aab (Android App Bundle)
C) .exe
D) .jar
26. Что такое «Material Design»?
A) Язык программирования
B) Набор рекомендаций Google по дизайну интерфейсов
C) Система контроля версий
D) Тип базы данных
27. Какой метод вызывается, когда Activity становится видимой, но не в фокусе (например, при появлении диалогового окна)?
A) onCreate()
B) onStop()
C) onResume()
D) onPause()
28. Что такое «ViewModel» в архитектуре Android-приложений?
A) Виджет интерфейса
B) Класс для хранения данных, который переживает смену конфигурации (поворот экрана)
C) Сервис для работы с сетью
D) Анимация перехода между экранами
29. Какой инструмент помогает создавать диаграммы Use Case и ER-диаграммы для проектирования приложений?
A) Android Studio
B) Git
C) Microsoft Visio
D) MySQL Installer
30. Какой метод используется для логирования сообщений в Android?
A) print()
B) Log.d()
C) debug()
D) message()

Тестовые задания типа В. (Установите соответствие между элементами)

Задание 1. Соотнесите компоненты Android-приложения с их назначением

Компоненты	Назначение
A. Activity	1. Фоновая задача без интерфейса
B. Service	2. Основной экран/окно приложения
C. Broadcast Receiver	3. Реагирует на системные события (например, SMS, зарядка)
D. Content Provider	4. Обеспечивает доступ к данным для других приложений

Верный ответ: A-2, B-1, C-3, D-4.

Задание 2. Соотнесите файлы проекта Android с их содержимым

Файлы	Содержимое
A. AndroidManifest.xml	1. Строковые ресурсы (тексты кнопок, заголовки)
B. strings.xml	2. Описание компонентов приложения, разрешений, версии
C. build.gradle	3. Настройки сборки проекта, зависимости библиотек
D. activity_main.xml	4. Разметка интерфейса (кнопки, текстовые поля и т.д.)

Верный ответ: A-2, B-1, C-3, D-4.

Задание 3. Соотнесите методы жизненного цикла Activity с их описанием

Методы жизненного цикла	Описание
A. onCreate()	1. Вызывается, когда Activity уходит на задний план
B. onStart()	2. Создаёт интерфейс и инициализирует компоненты
C. onPause()	3. Вызывается перед тем, как Activity становится видимой
D. onDestroy()	4. Вызывается при уничтожении Activity

Верный ответ: A-2, B-3, C-1, D-4.

Задание 4. Соотнесите контейнеры макета в Android с их особенностями

Контейнеры макета	Особенности
A. LinearLayout	1. Располагает элементы в виде сетки
B. RelativeLayout	2. Располагает элементы последовательно (вертикально/горизонтально)
C. ConstraintLayout	3. Позиционирует элементы относительно друг друга или родительского контейнера
D. GridView	4. Гибкий контейнер с ограничениями для адаптивного интерфейса

Верный ответ: A-2, B-3, C-4, D-1.

Задание 5. Соотнесите инструменты разработки с их функциями

Инструменты разработки	Функции
A. Android Studio	1. Система контроля версий для командной разработки
B. Git	2. Интегрированная среда разработки (IDE) для Android

C. Gradle	3. Система автоматизации сборки проектов
D. Logcat	4. Инструмент для просмотра логов приложения

Верный ответ: A-2, B-1, C-3, D-4.

Задание 6. Соотнесите понятия TensorFlow Lite с их описанием

Понятия	Описание
A. .tflite	1. Класс для загрузки и запуска модели в Android
B. Interpreter	2. Формат файла оптимизированной модели для мобильных устройств
C. Квантование	3. Уменьшение точности весов модели для ускорения работы и сокращения размера
D. Предобученная модель	4. Модель, обученная на большом наборе данных (например, MobileNet)

Верный ответ: A-2, B-1, C-3, D-4.

Задание 7. Соотнесите API камеры Android с их характеристиками

API камеры Android	Характеристика
A. Camera1 API	1. Современный API с упрощённым интерфейсом, часть Jetpack
B. Camera2 API	2. Устаревший API с ограниченной функциональностью
C. CameraX	3. Низкоуровневый API с расширенными возможностями настройки

Верный ответ: A-2, B-3, C-1.

Задание 8. Соотнесите типы тестирования с их описанием

Типы тестирования	Описание
A. Unit-тесты	1. Тестирование взаимодействия пользователя с интерфейсом
B. UI-тесты (Espresso)	2. Проверка отдельных модулей кода (функций, классов)
C. Интеграционные тесты	3. Проверка взаимодействия между компонентами приложения
D. Профилирование	4. Анализ потребления ресурсов (CPU, RAM, батарея)

Верный ответ: A-2, B-1, C-3, D-4.

Задание 9. Соотнесите форматы файлов с их назначением в мобильной разработке

Формат файла	Назначение
A. .apk	1. Файл оптимизированной модели ИИ для мобильных устройств
B. .aab	2. Установочный пакет Android-приложения
C. .tflite	3. Формат публикации в Google Play с оптимизацией по размеру
D. .json	4. Формат данных для обмена между приложением и сервером

Верный ответ: A-2, B-3, C-1, D-4.

Задание 10. Соотнесите архитектурные паттерны с их описанием

Архитектурные паттерны	Описание
A. MVC	1. Разделяет логику на Model, View, Presenter (Presenter управляет обновлением View)
B. MVP	2. Классическая архитектура: Model, View, Controller (Controller управляет View)
C. MVVM	3. Разделяет логику на Model, View, ViewModel (ViewModel предоставляет данные для View)
D. Clean Architecture	4. Многослойная архитектура с чётким разделением ответственности

Верный ответ: A-2, B-1, C-3, D-4.

Тестовые задания типа С. (Выбор правильного варианта ответа из предложенного списка)

Задание 1. Создание простого интерфейса для приложения «Калькулятор»

Условие: вы разрабатываете простое приложение-калькулятор с кнопками 0–9, «+», «-», «=»,

а также полем отображения результата. Какой подход лучше всего использовать для разметки интерфейса?

- A) Создать все кнопки и поле ввода программно в классе Activity, без использования XML.
- B) Использовать только LinearLayout с вертикальной ориентацией для всех элементов.
- C) Применить TableLayout для кнопок цифр и операций, а для поля отображения — отдельный TextView над ним.
- D) Использовать ConstraintLayout, разместив TextView сверху, а кнопки сгруппировать в GridLayout или через ограничения внутри ConstraintLayout.**

Задание 2. Внедрение TFLite-модели для распознавания цифр

Условие: вам нужно добавить в приложение функцию распознавания нарисованных цифр (0–9) с помощью TensorFlow Lite. Модель уже обучена и конвертирована в формат .tflite.

Что необходимо сделать в первую очередь для её интеграции в Android-приложение?

- A) Переобучить модель под Android с помощью Android Studio.
- B) Поместить файл модели (model.tflite) в папку assets проекта и добавить зависимость TensorFlow Lite в build.gradle.**
- C) Преобразовать файл .tflite в .jar и добавить его как библиотеку.
- D) Загрузить модель напрямую из интернета при запуске приложения без сохранения.

Задание 3. Обработка нажатия кнопки «=» в калькуляторе

Условие: в приложении-калькуляторе пользователь нажимает кнопку «=».

Приложение должно вычислить результат выражения и отобразить его в поле вывода. Какой фрагмент кода корректно реализует эту логику в Kotlin?

- A)

```
kotlin      ⇨ Переносить  ↗ Свернуть  📋 Копировать

buttonEquals.setOnClickListener {
    val result = "Вычисление..."
    textViewResult.text = result
}
```

В)

```
kotlin      ⇨ Переносить  ↗ Свернуть  📋 Копировать

buttonEquals.setOnClickListener {
    try {
        val expression = textViewInput.text.toString()
        val result = evaluateExpression(expression) // функция вычисляет
        textViewResult.text = result.toString()
    } catch (e: Exception) {
        textViewResult.text = "Ошибка"
    }
}
```

С)

```
kotlin      ⇨ Переносить  ↗ Свернуть  📋 Копировать

buttonEquals.setOnClickListener {
    textViewResult.text = textViewInput.text
}
```

Д)

```
kotlin      ⇨ Переносить  ↗ Свернуть  📋 Копировать

buttonEquals.setOnClickListener {
    startActivity(Intent(this, ResultActivity::class.java))
}
```

Задание 4. Тестирование функции распознавания цифр

Условие: вы внедрили TFLite-модель для распознавания нарисованных цифр. Теперь нужно проверить её работу. Какой способ тестирования наиболее корректен для оценки точности модели на мобильном устройстве?

А) Протестировать модель только на одном изображении цифры «5» и сделать вывод о её работе.

В) Запустить приложение, нарисовать цифру от руки, но не проверять результат — полагаться на корректность предобученной модели.

С) Использовать встроенный инструмент Android Profiler для измерения потребления памяти моделью и считать это достаточным тестированием.

Д) Подготовить набор из 20–30 изображений цифр (разных стилей написания),

запустить распознавание в приложении, сравнить предсказанные классы с истинными значениями и посчитать точность (accuracy).

4. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ

Код ПК, ОК	Критерии оценки результата (показатели освоения компетенций)	Формы контроля и методы оценки
ОК.01	распознает задачу и/или проблему в профессиональном и/или социальном контексте; анализирует задачу и/или проблему; определяет этапы решения задачи; выявляет и эффективно находит информацию, необходимую для решения задачи и/или проблемы; составляет план действия; определяет необходимые ресурсы; оценивает результат и последствия своих действий (самостоятельно или с помощью наставника)	Контрольные работы, зачеты, квалификационные испытания, защита курсовых и дипломных проектов (работ), учебная и производственная практики, экзамены. Интерпретация результатов выполнения практических и лабораторных заданий, оценка решения ситуационных задач, оценка тестового контроля, результатов наблюдений за деятельностью обучающегося в процессе учебной и производственной практики.
ОК.02	определяет задачи для поиска информации; определяет необходимые источники информации; планирует процесс поиска; структурирует полученную информацию; выделяет наиболее значимое в перечне информации; оценивает практическую значимость результатов поиска; оформляет результаты поиска	
ОК.03	определяет актуальность нормативно-правовой документации в профессиональной деятельности; применяет современную научную профессиональную терминологию; определяет и выстраивает траектории профессионального развития и самообразования	
ОК.04	организовывает работу коллектива и команды; взаимодействует с коллегами, руководством, клиентами в ходе профессиональной деятельности	
ОК.05	излагает свои мысли и оформляет документы по профессиональной тематике на государственном языке, проявляет толерантность в рабочем коллективе	
ОК.06	описывает значимость своей специальности	
ОК.07	соблюдает нормы экологической безопасности; определяет направления ресурсосбережения в рамках профессиональной деятельности по специальности	
ОК.08	чередует смену деятельности; выполняет комплекс лечебной гимнастики с учетом профессиональной деятельности	

ОК.09	понимает общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимает тексты на базовые профессиональные темы; участвует в диалогах на знакомые общие и профессиональные темы; пишет простые связные сообщения на знакомые или интересующие профессиональные темы
ПК 1.1	проектирует концептуальные, логические и физические модели базы данных; нормализует и оптимизирует структуру; документирует схему, включая ER-диаграммы, таблицы, права доступа и роли; определяет требования к БД и обеспечивает их реализацию в соответствии с предметной областью и принципами безопасности хранения данных
ПК 1.2	реализует программный модуль в полном соответствии с требованиями технического задания. Соблюдает принципы модульного программирования: обеспечивает чёткое разделение ответственности между компонентами, добивается низкой связности и высокой сплочённости модулей. Использует стандартные библиотеки и фреймворки для ИИ (например, scikit-learn, TensorFlow, PyTorch) там, где это целесообразно. Исключает дублирование кода как внутри модуля, так и между модулями. Обеспечивает корректную обработку исключительных ситуаций и граничных случаев. Оптимизирует реализацию по времени выполнения и использованию памяти. Проверяет работоспособность модуля в составе системы и обеспечивает его корректное взаимодействие с другими компонентами
ПК 1.3	соблюдает стандарты оформления кода (PEP 8 для Python, Google Java Style Guide и т.п.): правильно расставляет отступы, следит за длиной строк и использует корректные соглашения об именовании. Поддерживает единообразие стиля кода во всём проекте — форматирование и соглашения об именах переменных, функций и классов не меняются от файла к файлу. Составляет полную и понятную документацию: добавляет docstrings для функций и классов, создаёт README-файл с описанием проекта и инструкциями по запуску. Пишет самодocumentирующийся код с осмысленными именами идентификаторов, чтобы минимизировать потребность в комментариях. Пропускает код через инструменты статического анализа (PyLint, ESLint, Checkstyle) и исправляет все критические замечания. Оставляет комментарии только в тех местах, где логика неочевидна, — при этом комментарии объясняют «почему», а не «что» делается.

ПК 1.4	правильно использует ветки разработки (feature branches), основную ветку (main/master) и ветки релизов. Составляет осмысленные и информативные сообщения к коммитам (commit messages), описывая суть изменений и следуя шаблону (например, Conventional Commits). Поддерживает чистоту истории коммитов: избегает «фиктивных» коммитов и объединяет мелкие изменения в логические блоки. Грамотно сливает ветки (merge/rebase), не теряя функциональности. Успешно разрешает конфликты при слиянии кода: анализирует различия, выбирает корректную версию и тестирует систему после разрешения конфликта. Регулярно выполняет операции pull/push для синхронизации локальной и удалённой копии репозитория. Настраивает права доступа и использует pull requests/merge requests для ревью кода при организации совместной работы.
ПК 1.5	использует отладчик (PyCharm Debugger, Visual Studio Debugger и т.п.): устанавливает точки останова (breakpoints), выполняет код пошагово (step over/into) и просматривает значения переменных. Применяет методы логирования: настраивает уровни логирования (DEBUG, INFO, WARN, ERROR) и выводит контекст (имя файла, функция, номер строки). Быстро и точно идентифицирует ошибки, локализуя проблему по логам и поведению программы. Создаёт юнит-тесты для ключевых функций и интеграционные тесты для модулей, обеспечивая полное покрытие тестами. Исправляет ошибки без внесения новых багов: проводит регрессионное тестирование и проверяет граничные условия. Анализирует стек вызовов (stack trace) для сложных ошибок (исключения, утечки памяти). Документирует найденные ошибки и способы их устранения, например в issue tracker

Критерии оценки устного ответа обучающихся в 5-балльной системе. При оценке ответа обучающегося учитывается:

- 1) полнота и правильность ответа;
- 2) степень осознанности, понимания изученного;
- 3) языковое оформление ответа. Отметка «5»: ответ исчерпывающий, точный, полный и правильный на основании изученного материала; материал изложен в определенной логической последовательности, литературным языком; ответ самостоятельный.

Отметка «4»: ответ полный, обнаруживающий хорошее знание и понимание изученного материала; материал изложен в определенной логической последовательности, последовательно и грамотно, возможны отдельные затруднения в формулировке выводов.

Отметка «3»: ответ полный, но при этом допущена существенная ошибка, или неполный, несвязный ответ, изложенный нелогично, ставится за ответ, в котором в основном правильно, но схематично или с отклонениями от последовательности

изложения раскрыт материал.

Отметка «2»: при ответе обнаружено непонимание обучающимся основного содержания учебного материала, неумение его анализировать допущены существенные ошибки, которые обучающийся не смог исправить при наводящих вопросах преподавателя, отсутствует логика в изложении материала, нет необходимых обобщений и самостоятельной оценки фактов; недостаточно сформированы навыки устной речи.

Общая классификация ошибок. При оценке знаний, умений, навыков учитываются все ошибки (грубые и негрубые), а также недочёты в работе.

Грубыми считаются ошибки: - незнание определения основных понятий, законов, общепринятых символов обозначений величин; - неумение выделить в ответе главное; обобщить результаты изучения; - неумение применить знания для решения задач, объяснения явления; - неумение подготовить установку или лабораторное оборудование, провести опыт, наблюдение, использовать полученные данные для выводов; - неумение пользоваться первоисточниками, учебником, справочником; - нарушение техники безопасности, небрежное отношение к оборудованию, приборам, материалам.

Негрубыми считаются ошибки: - неточность формулировок, определений, понятий, законов, вызванная неполнотой охвата основных признаков определяемого понятия или заменой 1-3 из этих признаков второстепенными; - ошибки при снятии показаний с измерительных приборов, не связанные с определением цены деления шкалы; - ошибки, вызванные несоблюдением условий проведения опыта, наблюдения, условий работы прибора, оборудования; - нерациональный метод решения задачи, выполнения части практической работы, недостаточно продуманный план устного ответа (нарушение логики изложения, подмена отдельных основных вопросов второстепенными); - нерациональные методы работы со справочной литературой; - неумение решать задачи, выполнять задания в общем виде.

Недочётами являются: - нерациональные приёмы выполнения опытов, наблюдений, практических заданий; - арифметические ошибки в вычислениях; - орфографические и пунктуационные ошибки.

3.2. Критерии оценивания выполнения практического задания обучающихся в 5-балльной системе:

– Отметка «5»: работа выполнена полностью и правильно; сделаны правильные выводы.

– Отметка «4»: работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя.

– Отметка «3»: работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки.

– Отметка «2»: допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя.

3.3. Критерии оценивания результатов тестирования

<i>Оценка в баллах</i>	<i>Степень выполнения задания</i>
Неудовлетворительно	Выполнено не менее 40 % предложенных заданий
Удовлетворительно	Выполнено не менее 41-70 % предложенных заданий
Хорошо	Выполнено не менее 71-95% предложенных заданий
Отлично	Выполнено не менее 96-100% предложенных заданий

3.4. Критерии оценки написания сообщений, докладов:

- оценка «отлично» выставляется обучающемуся, если выполнены все требования к написанию сообщения (доклада): обозначена проблема и обоснована её актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы;

- оценка «хорошо» выставляется обучающемуся, если основные требования к сообщению, докладу и его защите выполнены, но при этом допущены недочёты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объём доклада (сообщения); имеются упущения в оформлении; на дополнительные вопросы даны неполные ответы.

- оценка «удовлетворительно» выставляется обучающемуся, если имеются существенные отступления от требований к написанию сообщения (доклада). В частности, тема освещена лишь частично; допущены фактические ошибки в содержании сообщения (доклада) или при ответе на дополнительные вопросы; во время защиты отсутствует вывод;

- оценка «неудовлетворительно» выставляется обучающемуся, если тема сообщения (доклада) не раскрыта, обнаруживается существенное непонимание проблемы.

3.5 Критерии оценивания презентаций:

– Оценка «отлично»: Содержание: Работа полностью завершена, обучающийся демонстрирует глубокое понимание описываемых процессов, даны интересные дискуссионные материалы, грамотно используется лексика, предлагается собственная интерпретация или развитие темы. Дизайн логичен. Все параметры шрифта хорошо подобраны. Текст хорошо читается. Графика подобрана грамотно, соответствует содержанию. Нет орфографических и синтаксических ошибок.

– Оценка «хорошо»: Полностью сделаны наиболее важные компоненты работы, обучающийся демонстрирует понимание основных моментов, хотя некоторые детали не уточняются. Некоторые материалы носят дискуссионный характер. Научная лексика используется, но иногда не корректно. Обучающийся в большинстве случаев предлагает собственную интерпретацию или развитие темы. Дизайн презентации выдержан и соответствует содержанию. Параметры шрифта подобраны. Графика соответствует содержанию. Минимальное количество ошибок.

– Оценка «удовлетворительно»: В содержании не выделены все важные компоненты. Обучающийся демонстрирует неполное понимание темы. Дискуссионные материалы есть в наличии, но не способствуют раскрытию проблемы. Научная терминология используется не всегда корректно. Дизайн не соответствует полному раскрытию содержания. Параметры шрифта недостаточно хорошо подобраны и могут мешать восприятию. Графика не в полной мере соответствует содержанию. Имеются орфографические и пунктуационные ошибки, мешающие восприятию.

– Оценка «неудовлетворительно»: Работа выполнена фрагментарно и с посторонней помощью, обучающийся демонстрирует минимальное понимание темы. Минимум дискуссионных материалов и научных терминов. Интерпретация ограничена или беспочвенна. Дизайн неясен. Элементы дизайна мешают содержанию. Текст трудночитаемый. Графика не соответствует содержанию. Много орфографических и пунктуационных ошибок, делающих материал трудночитаемым.