

Документ подписан простой электронной подписью
 Информация о владельце:
 ФИО: Стрельникова Светлана Евгеньевна
 Должность: ректор
 Дата подписания: 28.09.2026 17:22:21
 Уникальный программный ключ:
 ed7e7ef871ef28d870011250a92f17921078cc18

**Автономная некоммерческая образовательная организация высшего образования
 «Контур Университет»**



УТВЕРЖДАЮ

Ректор

С.Е. Стрельникова

15.09.2026 г.

Разработка мобильных приложений рабочая программа дисциплины (модуля)

Учебный план 09.03.03_ИИС_2026.plx
 09.03.03 Прикладная информатика

Квалификация **бакалавр**

Форма обучения **очная**

Общая трудоемкость **1 ЗЕТ**

Часов по учебному плану 36

Виды контроля в семестрах:

в том числе:

зачеты 5

аудиторные занятия 16

самостоятельная работа 18

часов на контроль 2

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	5 (3.1)		Итого	
Неделя	17 3/6			
Вид занятий	уп	рп	уп	рп
Практические	16	16	16	16
Итого ауд.	16	16	16	16
Контактная работа	16	16	16	16
Сам. работа	18	18	18	18
Часы на контроль	2	2	2	2
Итого	36	36	36	36

Рабочая программа дисциплины

Разработка мобильных приложений

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки
09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана:

09.03.03 Прикладная информатика

Одобрена Ученым советом вуза

Протокол от 15.09.2026 № 05.

Одобрена учебно-методической комиссией

Протокол от 10.09.2026 г. № 01

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
1.1	Цель освоения дисциплины:
1.2	Формирование у обучающихся системы знаний о технологиях, платформах и инструментах разработки мобильных приложений, а также развитие практических навыков проектирования пользовательских интерфейсов, программирования, отладки и тестирования мобильных приложений для операционных систем Android и iOS .
1.3	Задачи изучения дисциплины:
1.4	Ознакомить с архитектурой и жизненным циклом мобильных приложений, основными платформами (Android, iOS) и инструментами разработки (Android Studio, Xcode) ;
1.5	Сформировать знания о языках программирования для разработки мобильных приложений (Java, Kotlin, Swift) и принципах построения пользовательских интерфейсов ;
1.6	Развить навыки проектирования адаптивных интерфейсов для мобильных устройств с учётом принципов UI/UX дизайна ;
1.7	Изучить методы работы с базами данных, сетевыми протоколами и встроенными устройствами (камера, GPS, датчики) ;
1.8	Сформировать умения тестировать, отлаживать и оптимизировать мобильные приложения, обеспечивая их безопасность и производительность.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	ФТД
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Программирование
2.1.2	Информационные технологии и сервисы
2.1.3	Проектирование информационных систем
2.2	Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Web-технологии

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ИНДИКАТОРЫ ИХ ДОСТИЖЕНИЯ
ПК-3: Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации
ПК-3.3: Разрабатывает рабочие прототипы и программные модули среднего уровня сложности с использованием современных языков программирования и принципов качественного кода
Знать: современные языки программирования, используемые для разработки мобильных приложений (Java, Kotlin, Swift); принципы построения качественного кода (чистота кода, паттерны проектирования, рефакторинг); структуру и компоненты мобильных приложений (Activity, Fragment, Service, Intent) Уметь: разрабатывать программные модули для мобильных платформ в соответствии с техническим заданием; применять принципы объектно-ориентированного программирования при создании мобильных приложений; осуществлять рефакторинг и оптимизацию программного кода Владеть: навыками разработки рабочих прототипов мобильных приложений; методами отладки и тестирования программных модулей; инструментальными средствами разработки мобильных приложений (Android Studio, Xcode)

ПК-6: Способен самостоятельно и в составе коллектива выполнять полный цикл постановки и решения отдельных прикладных информационных задач в рамках научно-исследовательской, проектной и учебно-профессиональной деятельности для поиска, выработки и применения новых решений в области ИКТ
ПК-6.5: Разрабатывает план верификации предлагаемого решения на основе критериев успеха и метрик оценки, проводит апробацию (прототипирование, пилотное внедрение, А/В-тестирование), интерпретирует полученные результаты и оценивает эффективность решения по заданным метрикам
Знать: методы верификации и валидации программного обеспечения; критерии оценки качества мобильных приложений (производительность, удобство использования, безопасность); подходы к проведению А/В-тестирования и сбору пользовательских метрик Уметь: разрабатывать план тестирования мобильного приложения; проводить прототипирование и пилотное внедрение мобильных решений; интерпретировать результаты тестирования и оценивать эффективность разработанного решения по заданным метрикам Владеть:

навыками Unit-тестирования и UI-тестирования мобильных приложений;
методами анализа пользовательского опыта (UX) и сбора обратной связи;
инструментами профилирования и оптимизации производительности мобильных приложений

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Примечание
	Раздел 1.			
1.1	Введение в разработку мобильных приложений: платформы, архитектура, жизненный цикл /Пр/	5	1	
1.2	Введение в разработку мобильных приложений: платформы, архитектура, жизненный цикл /Ср/	5	1	
1.3	Основы программирования для мобильных платформ (Java / Kotlin / Swift) /Пр/	5	1	
1.4	Основы программирования для мобильных платформ (Java / Kotlin / Swift) /Ср/	5	1	
1.5	Среда разработки Android Studio и структура Android- проекта /Пр/	5	1	
1.6	Среда разработки Android Studio и структура Android- проекта /Ср/	5	2	
1.7	Компоненты Android-приложения. Жизненный цикл Activity и Fragment /Пр/	5	1	
1.8	Компоненты Android-приложения. Жизненный цикл Activity и Fragment /Ср/	5	2	
1.9	Создание пользовательского интерфейса в Android: XML-разметка, адаптивный дизайн /Пр/	5	2	
1.10	Создание пользовательского интерфейса в Android: XML-разметка, адаптивный дизайн /Ср/	5	2	
1.11	Меню, диалоговые окна, стили и темы в мобильных приложениях /Пр/	5	2	
1.12	Меню, диалоговые окна, стили и темы в мобильных приложениях /Ср/	5	2	
1.13	Работа с базами данных в мобильных приложениях (SQLite, Room) /Пр/	5	2	
1.14	Работа с базами данных в мобильных приложениях (SQLite, Room) /Ср/	5	2	
1.15	Работа с сетью: HTTP-запросы, REST API, библиотека Retrofit /Пр/	5	2	
1.16	Работа с сетью: HTTP-запросы, REST API, библиотека Retrofit /Ср/	5	2	
1.17	Работа с мультимедиа и встроенными устройствами (камера, GPS, датчики) /Пр/	5	2	
1.18	Работа с мультимедиа и встроенными устройствами (камера, GPS, датчики) /Ср/	5	2	
1.19	Тестирование, отладка и оптимизация мобильных приложений. Обеспечение безопасности /Пр/	5	2	
1.20	Тестирование, отладка и оптимизация мобильных приложений. Обеспечение безопасности /Ср/	5	2	
1.21	/Зачёт/	5	2	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации

Темы для обсуждения на практических занятиях:

Задание № 1 «Анализ мобильных платформ и выбор технологии под проект» (к теме 1.1)

Тема: Введение в разработку мобильных приложений: платформы, архитектура, жизненный цикл.

Цель: сформировать навыки сравнительного анализа мобильных платформ, выбора технологии разработки в зависимости от требований проекта и понимания архитектурных особенностей мобильных приложений.

Условие: Вы — технический консультант ИТ-компании, к которой обратился заказчик с запросом на разработку мобильного приложения. Заказчик — сеть фитнес-клубов «FitLife» с 50 клубами по всей России. Требуется разработать

мобильное приложение для клиентов, которое должно включать: расписание тренировок, запись на занятия, личный кабинет с историей посещений, систему лояльности, интеграцию с фитнес-браслетами, чат с тренерами. Бюджет проекта ограничен, сроки сжатые. Заказчик также планирует выход на рынки стран СНГ в ближайшие 2 года, но не исключает и европейское направление. В компании-заказчике нет опыта разработки мобильных приложений, но есть сильный веб-отдел на React.

Задания для студентов:

1. Проведите сравнительный анализ нативной разработки (Android + iOS), кроссплатформенной разработки (Flutter, React Native) и веб-приложений (PWA) по критериям: стоимость разработки, скорость, производительность, доступ к встроенным устройствам, опыт команды, поддержка и обновления.
2. На основе анализа выберите оптимальный стек технологий для данного проекта. Аргументируйте свой выбор с учётом требований заказчика (бюджет, сроки, масштабирование на СНГ и Европу, интеграция с фитнес-браслетами).
3. Опишите архитектуру выбранного мобильного приложения: какие компоненты (Activity/Fragment, Service, BroadcastReceiver, ContentProvider) будут использоваться, как организован жизненный цикл приложения, как будет реализована работа в фоновом режиме (например, синхронизация данных с сервером).
4. Составьте дорожную карту (план-график) разработки приложения с указанием ключевых этапов, сроков и ответственных ролей.

Форма отчётности: Аналитическая записка в виде структурированного документа с выводами по каждому пункту и обоснованием выбранного решения.

Задание № 2 «Решение алгоритмических задач на Kotlin/Swift» (к теме 1.2)

Тема: Основы программирования для мобильных платформ (Java / Kotlin / Swift).

Цель: сформировать навыки программирования на языке Kotlin (или Swift) для решения прикладных задач; отработать применение объектно-ориентированного подхода.

Условие: Компания разрабатывает мобильное приложение для интернет-магазина. Ваша задача как младшего разработчика — реализовать базовую логику работы с корзиной товаров и обработки заказов. Ожидается, что код будет соответствовать принципам чистого кода и будет готов к интеграции в проект (т.е. написан с учётом возможных изменений).

Задания для студентов:

1. На языке программирования Kotlin (или Swift — по выбору студента, но предпочтение Kotlin для Android-направления):
 - о Спроектировать классы «Товар» (название, цена, артикул, наличие на складе) и «Корзина» (список товаров, общая стоимость, количество позиций).
 - о Реализовать методы для добавления товара в корзину с проверкой наличия на складе, удаления товара из корзины, расчёта итоговой стоимости с учётом скидки (например, 10% при сумме более 5000 рублей) и очистки корзины.
 - о Реализовать функцию «Оформить заказ», которая проверяет наличие всех товаров на складе, формирует объект «Заказ» (номер заказа, список товаров, итоговая сумма, дата, статус «Новый») и уменьшает остатки на складе.
2. Спроектировать иерархию классов для разных типов товаров (например, «Электроника», «Одежда», «Продукты») с переопределением метода расчёта скидки (разные категории могут иметь разные правила скидок).
3. Написать Unit-тесты (JUnit) для методов класса «Корзина» (добавление, удаление, расчёт итоговой суммы с учётом скидки).
4. Оформить код в соответствии с принципами Clean Code (осмысленные имена переменных, методы не длиннее 15 строк, отсутствие дублирования).

Форма отчётности: Исходный код проекта в виде архива (или ссылка на репозиторий), содержащий все классы и Unit-тесты, а также краткое описание (README) с инструкцией по запуску тестов.

Задание № 3 «Создание первого проекта в Android Studio» (к теме 1.3)

Тема: Среда разработки Android Studio и структура Android-проекта.

Цель: сформировать навыки работы в среде разработки Android Studio, понимания структуры проекта, умений настраивать эмуляторы и использовать основные инструменты отладки.

Условие: Вам поручено создать проект-заготовку для будущего мобильного приложения. Проект должен быть правильно структурирован, содержать базовые конфигурации и быть готовым к подключению системы контроля версий Git.

Задания для студентов:

1. Установить и настроить Android Studio (последняя стабильная версия). Создать новый проект с типом «Empty Activity» (пустая Activity), выбрав язык Kotlin и минимальную версию Android API 24 (Android 7.0).
2. Изучить структуру созданного проекта:
 - о Описать назначение основных папок и файлов: app/src/main/java, app/src/main/res, AndroidManifest.xml, build.gradle (Module: app), gradle.properties.
 - о Объяснить, что такое файл манифеста, и какие минимальные настройки в нём должны присутствовать.
3. Настроить эмулятор Android для тестирования: создать виртуальное устройство (Pixel 6 или аналогичное) с версией Android 12 или 13. Запустить проект на эмуляторе и убедиться, что приложение успешно компилируется и отображает «Hello World».
4. Установить на реальное физическое устройство (если есть) режим разработчика и отладку по USB; подключить устройство и запустить приложение на нём.
5. Создать локальный Git-репозиторий для проекта, добавить файл .gitignore с настройками для Android-проектов (стандартный шаблон из GitHub), выполнить первый коммит.

Форма отчётности: Скриншот экрана с запущенным приложением (эмулятор или реальное устройство), ссылка на Git-репозиторий, а также краткое письменное описание структуры проекта в формате «Шпаргалка для начинающего Android-разработчика».

Задание № 4 «Разработка приложения со сменой экранов и передачей данных» (к теме 1.4)

Тема: Компоненты Android-приложения. Жизненный цикл Activity и Fragment.

Цель: сформировать навыки создания многоэкранных приложений, управления жизненным циклом Activity и Fragment, передачи данных между компонентами.

Условие: В рамках проекта по разработке мобильного приложения «Книга рецептов» вам необходимо реализовать экран списка рецептов и экран детальной информации о рецепте. Приложение должно корректно обрабатывать поворот экрана и временную потерю данных (например, при сворачивании приложения). Дизайн не имеет значения — важна правильная архитектура и работа компонентов.

Задания для студентов:

1. Создать два экрана (Activity или Fragment — на выбор):

о Экран 1: список рецептов. Каждый элемент списка содержит название рецепта, краткое описание и иконку категории (можно использовать иконки из стандартной библиотеки). Список должен быть статичным (минимум 5 рецептов, захардкоженных в коде).

о Экран 2: детальная информация о рецепте (название, полное описание, ингредиенты, время приготовления, категория).

2. Реализовать переход с первого экрана на второй при клике на элемент списка с передачей данных о рецепте (использовать Intent с extra или Bundle для Fragment).

3. Добавить обработку жизненного цикла Activity/Fragment:

о Переопределить основные методы жизненного цикла (onCreate, onStart, onResume, onPause, onStop, onDestroy) и добавить в них вывод в Logcat сообщений (например, Log.d("Lifecycle", "onCreate вызван")).

о Сымитировать поворот экрана (или сворачивание/разворачивание приложения) и проследить в логах, какие методы жизненного цикла вызываются.

4. Реализовать сохранение состояния приложения при повороте экрана:

о Если на экране деталей был выбран рецепт, при повороте экрана должен открываться тот же рецепт, а не стартовый экран.

о Использовать onSaveInstanceState и onRestoreInstanceState для сохранения состояния.

5. Добавить кнопку «Назад» на экране деталей для возврата к списку рецептов.

6. (Дополнительно) Заменить стандартный ActionBar на Toolbar и добавить навигационный апплет (кнопку Back) на экране деталей.

Форма отчётности: Исходный код проекта (ссылка на репозиторий), содержащий все классы и ресурсы, скриншоты работы приложения, а также краткий отчёт с анализом вызовов жизненного цикла (таблица с методами и действиями пользователя, которые их вызывают).

Задание № 5 «Создание адаптивного интерфейса для разных экранов» (к теме 1.5)

Тема: Создание пользовательского интерфейса в Android: XML-разметка, адаптивный дизайн.

Цель: сформировать навыки верстки пользовательского интерфейса с использованием XML-разметки, разработки адаптивных интерфейсов для различных размеров экранов и ориентаций.

Условие: Заказчик предоставил дизайн-макет профиля пользователя для мобильного приложения. Макет содержит: аватар (круглое фото), имя и фамилию пользователя, статус (онлайн/офлайн), статистику (количество подписчиков, публикаций, лайков), кнопку «Редактировать профиль» и сетку из 6 последних фотографий пользователя (2 ряда по 3 фото). Дизайн-макет дан только для портретной ориентации на устройстве с диагональю 6 дюймов. Ваша задача — сверстать экран, но при этом обеспечить его корректное отображение на разных устройствах.

Задания для студентов:

1. Создать экран профиля пользователя в Android Studio с использованием XML-разметки. Использовать ConstraintLayout в качестве корневого элемента.

2. Реализовать адаптивный дизайн для следующих случаев:

о Разные размеры экранов: стандартные смартфоны (5-6 дюймов), планшеты (7-10 дюймов) — использовать ресурсные квалификаторы (например, layout-sw600dp для планшетов).

о Разные ориентации (портретная и ландшафтная): предусмотреть отдельные макеты для ландшафтной ориентации (папка layout-land), где расположение элементов изменяется (например, статистика может быть расположена горизонтально, а не вертикально).

о Разные разрешения экрана (плотности пикселей): предусмотреть ресурсы для разных плотностей (mdpi, hdpi, xhdpi, xxhdpi), хотя бы для аватара (иконки разных размеров).

3. Добавить поддержку разных языков (локализация): предусмотреть строковые ресурсы в папке values-ru для русского языка и values-en для английского.

4. Настроить обработку тёмной и светлой темы (использовать values-night для тёмной темы, изменить цвета, стиль текста и фон).

5. Вставить в разметку изображение аватара (можно использовать векторную иконку из стандартной библиотеки) и добавить вместо реальных фотографий placeholder (заглушку) с использованием библиотеки Glide или Picasso.

6. (Дополнительно) Добавить анимацию при нажатии на кнопку «Редактировать профиль» (например, ScaleAnimation или переход на другой экран с анимацией).

Форма отчётности: Исходный код проекта с ресурсными файлами, скриншоты экрана в разных ориентациях и на разных эмуляторах (планшет, смартфон), а также краткое описание использованных ресурсных квалификаторов.

Задание № 6 «Разработка меню и диалогов для приложения» (к теме 1.6)

Тема: Меню, диалоговые окна, стили и темы в мобильных приложениях.

Цель: сформировать навыки реализации различных типов меню, диалоговых окон, настройки стилей и тем для мобильных приложений.

Условие: В приложении «Список задач» (To-Do лист) необходимо реализовать следующие элементы интерфейса: выпадающее меню для добавления новой задачи, контекстное меню для управления существующими задачами, диалоговое окно для ввода данных, а также стилизацию приложения с использованием кастомных тем.

Задания для студентов:

1. На основе ранее созданного или нового проекта (список задач) реализовать следующие типы меню:
 - о Опционное меню (Options Menu): добавить в ActionBar (или Toolbar) кнопку с тремя точками, которая открывает меню с пунктами: «Добавить задачу», «Очистить выполненные», «Настройки», «Выход».
 - о Контекстное меню (Context Menu): при долгом нажатии на задачу открывать меню с пунктами: «Редактировать», «Удалить», «Отметить выполненной».
 - о Всплывающее меню (Popup Menu): при нажатии на иконку шестерёнки (или три точки) рядом с задачей открывать PopupMenu с аналогичными пунктами.
 2. Реализовать диалоговые окна (Dialog) для следующих действий:
 - о AlertDialog для подтверждения удаления задачи («Вы уверены, что хотите удалить задачу?» с кнопками «Да» и «Нет»).
 - о Диалог добавления/редактирования задачи: использовать диалог с полем ввода (EditText) для ввода названия задачи и кнопками «Сохранить» и «Отмена».
 - о DatePickerDialog для выбора даты выполнения задачи.
 - о (Дополнительно) TimePickerDialog для выбора времени выполнения задачи.
 3. Настроить кастомные стили и темы:
 - о Создать кастомную тему в файле styles.xml, изменить основной цвет экрана (colorPrimary, colorPrimaryDark, colorAccent) в соответствии с брендбуком (например, синяя цветовая схема для приложения).
 - о Применить тему ко всему приложению в файле манифеста.
 - о Создать дополнительные стили для отдельных элементов (например, стиль для кнопок, стиль для заголовков).
 4. (Дополнительно) Реализовать BottomNavigationView с переключением между фрагментами (например, «Задачи», «Календарь», «Настройки»).
- Форма отчётности: Исходный код проекта, скриншоты всех типов меню и диалоговых окон, демонстрирующих их работу, а также файл стилей с комментариями.

Задание № 7 «Разработка приложения с базой данных SQLite/Room» (к теме 1.7)

Тема: Работа с базами данных в мобильных приложениях (SQLite, Room).

Цель: сформировать навыки проектирования и работы с локальной базой данных SQLite с использованием библиотеки Room, выполнения операций CRUD, работы с миграциями и тестирования.

Условие: Компания разрабатывает приложение для ведения личного бюджета «Мои финансы». Требуется реализовать локальную базу данных для хранения информации о доходах и расходах с возможностью добавления, редактирования, удаления и отображения записей. База данных должна содержать минимум две связанные таблицы: категории расходов/доходов и транзакции.

Задания для студентов:

1. Спроектировать структуру базы данных для приложения:
 - о Таблица «Категории»: id (первичный ключ), название категории, тип (доход/расход), иконка (строка с названием ресурса).
 - о Таблица «Транзакции»: id (первичный ключ), сумма, дата, описание, id категории (внешний ключ), тип (доход/расход).
 - о (Дополнительно) Таблица «Счета»: id, название, начальный баланс.
2. Реализовать модель данных с использованием библиотеки Room:
 - о Создать Entity-классы для таблиц (с аннотациями @Entity, @PrimaryKey, @ColumnInfo, @ForeignKey, @Relation).
 - о Создать DAO-интерфейсы (Data Access Object) с методами CRUD (insert, delete, update, query) с использованием аннотаций @Insert, @Delete, @Update, @Query.
 - о Создать класс Базы данных (наследуемый от RoomDatabase) с указанием всех Entity и версии базы данных.
3. Реализовать пользовательский интерфейс для работы с базой данных:
 - о Список всех транзакций с возможностью фильтрации по категории.
 - о Форма добавления новой транзакции (сумма, категория из выпадающего списка, дата, описание).
 - о Форма управления категориями (добавление и удаление категорий).
4. Отработать миграцию базы данных:
 - о Создать вторую версию базы данных с добавлением нового поля (например, «Способ оплаты» для транзакции) и написать миграцию, которая добавляет столбец с значением по умолчанию «Наличные».
5. Написать Unit-тесты для проверки работы DAO (например, тестирование вставки и чтения данных). Использовать библиотеку androidx.room:room-testing.

Форма отчётности: Исходный код проекта с реализованной моделью базы данных, скриншоты работы приложения (добавление транзакции, отображение списка, фильтрация), файл с миграцией, а также отчёт о выполнении Unit-тестов.

Задание № 8 «Интеграция приложения с REST API» (к теме 1.8)

Тема: Работа с сетью: HTTP-запросы, REST API, библиотека Retrofit.

Цель: сформировать навыки работы с сетевыми запросами, парсинга JSON-ответов, асинхронной обработки данных и отображения полученной информации в пользовательском интерфейсе.

Условие: Необходимо разработать приложение «Прогноз погоды», которое получает данные о погоде с открытого API (например, OpenWeatherMap или WeatherAPI). Приложение должно отображать текущую погоду по городу, прогноз на 5 дней и позволять пользователю искать погоду по разным городам. Приложение должно корректно обрабатывать ошибки

сети и отображать загрузку данных.

Задания для студентов:

1. Выбрать открытый API для получения данных о погоде (например, OpenWeatherMap). Зарегистрироваться и получить API-ключ.
 2. Реализовать сетевой слой с использованием библиотеки Retrofit:
 - о Создать интерфейс API с методами для получения текущей погоды и прогноза на несколько дней (использовать аннотации @GET, @Query).
 - о Создать модель данных (Data classes) для парсинга JSON-ответа (использовать аннотации @SerializedName для маппинга полей).
 - о Настроить Retrofit-билдер с указанием базового URL и конвертера Gson.
 3. Реализовать экран для отображения погоды:
 - о Поле ввода для названия города.
 - о Кнопка «Получить погоду».
 - о Отображение данных: название города, температура, описание погоды, иконка погоды, влажность, скорость ветра, давление.
 - о Отображение списка прогноза на 5 дней (минимум: день недели, температура, иконка).
 4. Реализовать обработку сетевых ошибок:
 - о Отображать сообщение об ошибке (например, «Город не найден», «Нет подключения к интернету») с использованием Toast или Snackbar.
 - о Показывать индикатор загрузки (ProgressBar) во время выполнения запроса.
 5. Реализовать асинхронную загрузку данных с использованием Coroutines или RxJava (на выбор). Использовать Dispatchers.IO для сетевого вызова и Dispatchers.Main для обновления UI.
 6. (Дополнительно) Добавить кэширование полученных данных (сохранение последнего запроса в SharedPreferences или кэш в памяти) и отображать последние данные при запуске приложения в случае отсутствия сети.
- Форма отчётности: Исходный код проекта с реализованным сетевым слоем, скриншоты работающего приложения с отображением погоды по разным городам, а также демонстрация обработки ошибок (неверный город, отсутствие интернета). В отчёте обязательно указать, какой API использован и какие методы применены.

Задание № 9 «Мультимедиа и работа с датчиками» (к теме 1.9)

Тема: Работа с мультимедиа и встроенными устройствами (камера, GPS, датчики).

Цель: сформировать навыки работы с камерой устройства, получения координат GPS, использования встроенных датчиков и воспроизведения аудио/видео.

Условие: Вам необходимо разработать многофункциональное приложение «Путешественник», которое объединяет в себе: возможность сделать фото и сохранить его с геопозицией, отображение текущих координат на карте, воспроизведение аудиогuida (аудиофайла) о достопримечательности, а также использование акселерометра для демонстрации наклона устройства (уровень).

Задания для студентов (разбить на модули):

Модуль 1. Камера и геолокация:

1. Реализовать запрос разрешений на использование камеры, GPS и внешнего хранилища (если требуется) в рантайме (Runtime Permissions).
2. Реализовать функцию съёмки фото с использованием встроенной камеры (Intent MediaStore.ACTION_IMAGE_CAPTURE). Сохранить полученное изображение в галерею или во внутреннее хранилище.
3. Реализовать получение текущих координат (широта, долгота) с использованием LocationManager или FusedLocationProviderClient. Периодически обновлять координаты при изменении местоположения.
4. Связать фото с GPS-координатами: добавить метаданные (Exif) к снимку с координатами места съёмки.
5. Отобразить миниатюру последнего сделанного снимка и его координаты на экране.

Модуль 2. Карта и геопозиция:

6. Интегрировать Google Maps (или OpenStreetMap) в приложение: отобразить текущее местоположение пользователя на карте с маркером.
7. Добавить возможность ставить маркер в любой точке карты и отображать координаты этой точки.

Модуль 3. Воспроизведение аудио (аудиогид):

8. Реализовать воспроизведение аудиофайла (например, MP3) из ресурсов приложения или из внешнего хранилища с использованием MediaPlayer.
9. Добавить управление плеером: Play / Pause / Stop, отображение длительности и текущего времени воспроизведения.

Модуль 4. Датчики (акселерометр):

10. Реализовать отображение текущего наклона устройства (углы по оси X и Y) в реальном времени с использованием датчика акселерометра (Sensor.TYPE_ACCELEROMETER). Можно отобразить в виде уровня (пузырька) или числовых значений.

Форма отчётности: Исходный код проекта, демонстрирующий работу всех модулей; скриншоты/видеозапись работы приложения (сделанное фото, отображение на карте, воспроизведение аудио, работа датчика).

Задание № 10 «Тестирование, отладка и оптимизация приложения» (к теме 1.10)

Тема: Тестирование, отладка и оптимизация мобильных приложений. Обеспечение безопасности.

Цель: сформировать навыки написания Unit-тестов и UI-тестов, использования инструментов профилирования и отладки, оптимизации производительности и обеспечения безопасности данных.

Условие: Вам предоставлен готовый проект приложения «Список задач» (или любой другой на выбор). Проект работает, но содержит ряд проблем: потенциальные утечки памяти, неоптимальный код в адаптере списка, отсутствие тестов,

незащищённое хранение данных пользователя, отсутствие обработки крашей. Ваша задача — выполнить рефакторинг, покрыть код тестами и оптимизировать приложение.

Задания для студентов:

Часть 1. Unit-тестирование (модульные тесты):

1. Написать Unit-тесты для ViewModel или презентера приложения (если используется MVVM/MVP): проверить логику добавления, удаления, обновления задач.
2. Использовать фреймворки JUnit и Mockito для моков зависимостей (репозиторий, база данных).
3. Проверить граничные случаи: добавление пустой задачи, удаление несуществующей задачи, обновление задачи с некорректными данными.

Часть 2. UI-тестирование (интерфейсные тесты):

4. Написать Espresso-тесты для проверки пользовательского интерфейса:

- o Проверка отображения списка задач после загрузки.
- o Проверка создания новой задачи через диалоговое окно.
- o Проверка отметки задачи как выполненной.
- o Проверка удаления задачи через контекстное меню.

5. Настроить запуск UI-тестов на эмуляторе.

Часть 3. Отладка и профилирование:

6. Использовать Android Profiler для анализа:

- o CPU: проверить загрузку процессора, найти потенциальные "узкие места" (например, тяжелые операции на главном потоке).
- o Memory (Память): проверить наличие утечек памяти (Memory Leaks) при пересоздании Activity или фрагментов. Использовать инструмент LeakCanary для обнаружения утечек.
- o Network (Сеть): если приложение использует сеть, проанализировать сетевые запросы (задержки, размер ответов).
- o Energy (Энергопотребление): оценить влияние приложения на батарею устройства.

7. Использовать Logcat для отладки: добавить логи с различными уровнями (Debug, Error, Info) для отслеживания ключевых событий в приложении.

8. Найти и исправить не менее 2-х проблем производительности (например, неправильное использование адаптера, тяжёлые вычисления в OnDraw, отсутствие переиспользования View в RecyclerView).

Часть 4. Обеспечение безопасности:

9. Проанализировать, какие данные хранятся в приложении. Обеспечить безопасное хранение:

- o Заменить SharedPreferences для хранения токенов/паролей на EncryptedSharedPreferences (Android 6.0+) или использовать Android Keystore System.

- o Если приложение работает с сетью, убедиться, что используется HTTPS (SSL/TLS) и настроить сертификаты.

- o Обеспечить защиту от обратной разработки (ProGuard/R8 обфускация кода в релизной сборке).

10. Настроить обработку ошибок (Crash Reporting): интегрировать Firebase Crashlytics для сбора отчётов о падениях.

Форма отчётности: Исходный код проекта с добавленными тестами, оптимизированным кодом и улучшенной безопасностью; отчёт о проведённой оптимизации (включая скриншоты Profiler до и после исправлений, описание найденных проблем и способов их решения); отчёт о покрытии кода тестами.

Итоговое задание (для получения зачета)

«Разработка мобильного приложения от идеи до публикации» (тема выбирается студентом самостоятельно)

Задание: Разработать полноценное мобильное приложение (Android или iOS — на выбор) по одной из предложенных тем или по согласованной с преподавателем. Приложение должно содержать:

- Пользовательский интерфейс с несколькими экранами (список, детали, форма добавления).
- Работу с локальной базой данных (SQLite/Room).
- Работу с сетью (REST API).
- Использование мультимедиа или датчиков.
- Unit-тесты и UI-тесты (минимум по 3 теста каждого типа).
- Оптимизированный код с отсутствием утечек памяти.
- Релизную сборку с обфускацией кода и настроенным Crash Reporting.

Форма отчётности: Готовый к публикации проект, презентация приложения (5–7 минут), отчётная документация (техническое задание, описание архитектуры, результаты тестирования, руководство пользователя).

Примерные вопросы для зачета:

1. Классификация мобильных приложений (нативные, веб-приложения, кроссплатформенные). Преимущества и недостатки каждого типа.
2. Архитектура мобильных приложений. Основные компоненты Android-приложения (Activity, Service, BroadcastReceiver, ContentProvider).
3. Жизненный цикл Activity в Android. Назовите и охарактеризуйте основные методы (onCreate, onStart, onResume, onPause, onStop, onDestroy).
4. Жизненный цикл Fragment. В чем отличие от жизненного цикла Activity?
5. Основы языка программирования Kotlin: переменные, типы данных, управляющие конструкции (условные операторы, циклы).
6. Объектно-ориентированное программирование в Kotlin: классы, наследование, полиморфизм, интерфейсы.
7. Основы языка программирования Swift: переменные, константы, коллекции, функции.
8. Среда разработки Android Studio: структура проекта, назначение файла манифеста, Gradle-сборка.
9. Создание пользовательского интерфейса в Android: XML-разметка, основные Layout-элементы (LinearLayout,

ConstraintLayout, RelativeLayout).

10. Адаптивный дизайн для различных размеров экранов и ориентаций. Ресурсные квалификаторы.
11. Меню в Android-приложениях: виды меню (опционное, контекстное, всплывающее).
12. Диалоговые окна в Android: AlertDialog, DatePickerDialog, TimePickerDialog.
13. Стили и темы в Android-приложениях. Способы кастомизации внешнего вида.
14. Фрагменты (Fragments): назначение, создание, добавление в Activity, взаимодействие между фрагментами.
15. Intent в Android: явные и неявные Intent. Передача данных между Activity.
16. База данных SQLite в мобильных приложениях. Основные операции (CRUD).
17. Библиотека Room. Преимущества перед SQLite, основные аннотации и компоненты.
18. SharedPreferences: назначение, способы хранения данных, примеры использования.
19. Работа с файловой системой в Android: внутреннее и внешнее хранилище.
20. Многопоточность в мобильных приложениях. AsyncTask, Handler, Coroutines.
21. Работа с сетью в Android. HTTP-запросы, библиотека Retrofit.
22. REST API: понятие, основные методы (GET, POST, PUT, DELETE). Примеры интеграции в мобильное приложение.
23. Работа с камерой в Android: получение изображения, сохранение в галерею.
24. Работа с GPS и геолокацией: получение координат, разрешения, службы местоположения.
25. Работа с датчиками устройства (акселерометр, гироскоп, датчик освещенности).
26. Воспроизведение аудио и видео в мобильных приложениях. MediaPlayer, ExoPlayer.
27. Пуш-уведомления в мобильных приложениях: принцип работы, Firebase Cloud Messaging.
28. Unit-тестирование мобильных приложений: JUnit, Mockito.
29. UI-тестирование мобильных приложений: Espresso. Принципы написания тестов.
30. Отладка и профилирование в Android Studio. Инструменты Logcat, Profiler, Debugger.
31. Оптимизация производительности мобильных приложений: управление памятью, энергопотребление, работа с большими данными.
32. Обеспечение безопасности данных в мобильных приложениях: шифрование, безопасное хранение токенов, защита от обратной разработки.
33. Сбор пользовательской аналитики в мобильных приложениях. Firebase Analytics, Crashlytics.
34. Кроссплатформенная разработка мобильных приложений: Flutter, React Native. Сравнение с нативной разработкой.
35. Процесс публикации мобильного приложения в Google Play и App Store. Требования к публикации.

5.2. Темы письменных работ

не предусмотрены

5.3. Фонд оценочных средств

Тестовые задания:

ВОПРОСЫ НА СООТВЕТСТВИЕ (№ 1–5)

Инструкция: к каждому элементу из левого столбца подберите ОДИН соответствующий элемент из правого столбца. Ответ запишите в виде сочетания цифр и букв (например: 1-А, 2-Б, 3-В, 4-Г). Каждая буква используется только один раз.

ПК-3.3. Вопрос № 1. Соответствие между компонентом Android-приложения и его назначением

Установите соответствие между компонентом и его функцией:

Компонент Назначение

1. Activity А. Компонент для выполнения длительных операций в фоновом режиме без пользовательского интерфейса
2. Service Б. Компонент для управления данными приложения и обеспечения доступа к ним других приложений
3. BroadcastReceiver В. Экран приложения с пользовательским интерфейсом, с которым взаимодействует пользователь
4. ContentProvider Г. Компонент для реагирования на системные события и широковебательные сообщения

Ответ: 1 __, 2 __, 3 __, 4 __

ПК-3.3. Вопрос № 2. Соответствие между методом жизненного цикла Activity и его описанием

Установите соответствие между методом и моментом его вызова:

Метод Activity Когда вызывается

1. onCreate() А. При остановке Activity (переход в фон)
2. onStart() Б. При уничтожении Activity
3. onResume() В. При создании Activity (вызывается один раз)
4. onPause() Г. При возобновлении работы Activity после паузы
5. onDestroy() Д. Когда Activity становится видимым для пользователя

Ответ: 1 __, 2 __, 3 __, 4 __, 5 __

ПК-3.3. Вопрос № 3. Соответствие между типом меню и его описанием

Установите соответствие между типом меню и его характеристикой:

Тип меню Описание

1. Опционное меню (Options Menu) А. Появляется при длительном нажатии на элемент и привязано к конкретному View
2. Контекстное меню (Context Menu) Б. Отображается в ActionBar/Toolbar при нажатии на иконку с тремя точками
3. Всплывающее меню (Popup Menu) В. Меню, привязанное к конкретному View и отображающееся в виде всплывающего списка
4. BottomNavigationView Г. Панель навигации в нижней части экрана с несколькими пунктами

Ответ: 1 __, 2 __, 3 __, 4 __

ПК-3.3. Вопрос № 4. Соответствие между типом Layout и его характеристикой

Установите соответствие между типом разметки и её описанием:

Тип разметки Описание

1. LinearLayout А. Размещает дочерние элементы в виде сетки с указанием количества столбцов
2. RelativeLayout Б. Размещает дочерние элементы в виде иерархической структуры с вкладками
3. ConstraintLayout В. Размещает дочерние элементы последовательно (вертикально или горизонтально)
4. GridLayout Г. Размещает дочерние элементы относительно друг друга или родительского контейнера
5. FrameLayout Д. Гибкая разметка с плоской иерархией, использующая ограничения (constraints)

Ответ: 1 __, 2 __, 3 __, 4 __, 5 __

ПК-6.5. Вопрос № 5. Соответствие между типом тестирования и его описанием

Установите соответствие между типом тестирования и его характеристикой:

Тип тестирования Описание

1. Unit-тестирование А. Тестирование пользовательского интерфейса и взаимодействия с ним
2. UI-тестирование Б. Тестирование отдельных модулей (классов, методов) в изоляции от других компонентов
3. Интеграционное тестирование В. Тестирование приложения в реальных условиях с участием пользователей
4. Нагрузочное тестирование Г. Проверка взаимодействия между различными компонентами приложения
5. Приёмочное тестирование Д. Проверка производительности и стабильности приложения при высоких нагрузках

Ответ: 1 __, 2 __, 3 __, 4 __, 5 __

ВОПРОСЫ С ОДНИМ ВЕРНЫМ ОТВЕТОМ (№ 6–15)

Инструкция: прочитайте текст, выберите правильный вариант ответа и запишите только букву выбранного варианта ответа (А, Б, В или Г).

ПК-3.3. Вопрос № 6. Какой метод жизненного цикла Activity вызывается при первом создании Activity?

- А) onStart()
- Б) onCreate()
- В) onResume()
- Г) onRestoreInstanceState()

Ответ:

ПК-3.3. Вопрос № 7. Какой компонент Android используется для выполнения фоновых задач без пользовательского интерфейса?

- А) Activity
- Б) Fragment
- В) Service
- Г) BroadcastReceiver

Ответ:

ПК-3.3. Вопрос № 8. Какой язык программирования является официальным для разработки под Android?

- А) Swift
- Б) Java
- В) C#
- Г) JavaScript

Ответ:

ПК-3.3. Вопрос № 9. Какой тег используется в XML-разметке для создания кнопки в Android?

- А) TextView
- Б) EditText
- В) Button
- Г) ImageView

Ответ:

ПК-3.3. Вопрос № 10. Какой компонент используется для отображения списка с возможностью прокрутки и переиспользованием элементов?

- А) ListView
- Б) RecyclerView
- В) GridView
- Г) ScrollView

Ответ:

ПК-3.3. Вопрос № 11. Какую библиотеку рекомендуется использовать для работы с базой данных SQLite в современном Android-разработке?

- А) SQLiteOpenHelper
- Б) Room
- В) Firebase Realtime Database
- Г) SharedPreferences

Ответ:

ПК-3.3. Вопрос № 12. Какой метод используется для отправки запроса к REST API в библиотеке Retrofit?

- А) call.enqueue()
- Б) execute()
- В) send()
- Г) request()

Ответ:

ПК-3.3. Вопрос № 13. Какой инструмент используется для профилирования производительности Android-приложений?

- А) Logcat
- Б) Android Profiler
- В) ADB
- Г) Gradle

Ответ:

ПК-6.5. Вопрос № 14. Что такое А/В-тестирование в контексте разработки мобильных приложений?

- А) Тестирование на двух устройствах разных производителей
- Б) Сравнение двух версий приложения (или его элементов) на разных группах пользователей для оценки эффективности
- В) Тестирование приложения в двух странах для проверки локализации
- Г) Тестирование двух разных языков программирования

Ответ:

ПК-6.5. Вопрос № 15. Какой инструмент используется для сбора отчётов о падениях (крашах) приложения?

- А) GitHub
- Б) Android Studio
- В) Firebase Crashlytics
- Г) JUnit

Ответ:

ВОПРОСЫ С НЕСКОЛЬКИМИ ВЕРНЫМИ ОТВЕТАМИ (№ 16–20)

Инструкция: прочитайте текст, выберите не менее двух правильных вариантов ответов и запишите только буквы выбранных вариантов ответов (А, Б, В или Г).

ПК-3.3. Вопрос № 16. Какие компоненты являются основными компонентами Android-приложения?

- А) Activity
- Б) Service
- В) BroadcastReceiver
- Г) ContentProvider

Ответ:

ПК-3.3. Вопрос № 17. Какие языки программирования официально поддерживаются для разработки мобильных приложений под Android?

- А) Java
- Б) Kotlin
- В) Swift
- Г) C++

Ответ:

ПК-3.3. Вопрос № 18. Какие типы меню доступны в Android-приложениях?

- А) Опционное меню (Options Menu)
- Б) Контекстное меню (Context Menu)
- В) Всплывающее меню (Popup Menu)
- Г) Выпадающее меню (Dropdown Menu)

Ответ:

ПК-3.3. Вопрос № 19. Какие устройства могут быть использованы для тестирования мобильного приложения?

- А) Эмулятор Android
- Б) Реальное физическое устройство
- В) Виртуальное устройство на основе QEMU
- Г) Калькулятор

Ответ:

ПК-6.5. Вопрос № 20. Какие метрики используются для оценки качества мобильных приложений?

- А) Производительность (скорость работы, время отклика)
- Б) Удобство использования (User Experience)
- В) Количество строк кода
- Г) Безопасность данных

Ответ:

ВОПРОСЫ С ОДНИМ АРГУМЕНТИРОВАННЫМ ОТВЕТОМ (№ 21–25)

Инструкция: Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.

ПК-3.3. Вопрос № 21. Какой из способов хранения данных в Android наиболее безопасен для хранения паролей и токенов аутентификации?

- А) SharedPreferences
- Б) EncryptedSharedPreferences
- В) SQLite база данных
- Г) Внешнее хранилище (файловая система)

Ответ:

ПК-6.5. Вопрос № 22. Для каких целей используется библиотека Espresso в Android-разработке?

- А) Для написания Unit-тестов
- Б) Для написания UI-тестов (тестирование пользовательского интерфейса)
- В) Для профилирования производительности
- Г) Для интеграции с внешними API

Ответ:

ПК-3.3. Вопрос № 23. Какой компонент Android лучше всего подходит для создания многоэкранного интерфейса с переключением между экранами?

- А) Activity
- Б) Fragment
- В) Service
- Г) BroadcastReceiver

Ответ:

ПК-3.3. Вопрос № 24. Какая библиотека является стандартом де-факто для выполнения HTTP-запросов в современных Android-приложениях?

- А) AsyncTask
- Б) Retrofit
- В) Volley
- Г) HttpURLConnection

Ответ:

ПК-6.5. Вопрос № 25. Какой метод оптимизации производительности является наиболее эффективным для длинных списков в RecyclerView?

- А) Использование сложных Layout-элементов для каждого элемента
- Б) Переиспользование ViewHolder и оптимизация onBindViewHolder()
- В) Загрузка всех данных сразу в память
- Г) Использование LinearLayout вместо RecyclerView

Ответ:

ВОПРОСЫ НА УСТАНОВЛЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ (№ 26–30)

Инструкция: Прочитайте текст, установите правильную последовательность действий и запишите аргументы, обосновывающие выбранный порядок.

ПК-3.3. Вопрос № 26. Установите правильную последовательность методов жизненного цикла Activity при обычном запуске приложения.

- А) onResume()
- Б) onCreate()
- В) onStart()

Ответ:

ПК-3.3. Вопрос № 27. Установите правильную последовательность методов жизненного цикла Activity при повороте экрана.

- А) onDestroy()
- Б) onStop()
- В) onSaveInstanceState()
- Г) onCreate()
- Д) onPause()
- Е) onRestoreInstanceState()

Ответ:

ПК-3.3. Вопрос № 28. Установите правильную последовательность этапов создания Android-проекта.

- А) Выбор типа Activity (Empty Activity, Basic Activity)
- Б) Установка и настройка Android Studio

В) Настройка эмулятора и запуск приложения
 Г) Настройка структуры проекта и добавление зависимостей в Gradle
 Д) Создание нового проекта с указанием имени, языка и минимальной версии SDK
 Ответ:

ПК-6.5. Вопрос № 29. Установите правильную последовательность процесса тестирования мобильного приложения.

- А) Написание Unit-тестов для проверки бизнес-логики
- Б) Проведение интеграционного тестирования (проверка взаимодействия компонентов)
- В) Проведение UI-тестирования (проверка пользовательского интерфейса)
- Г) Проведение нагрузочного тестирования и оптимизация производительности
- Д) Приёмочное тестирование (UAT — User Acceptance Testing) с участием заказчика

Ответ:

ПК-6.5. Вопрос № 30. Установите правильную последовательность действий при публикации приложения в Google Play.

- А) Создание релизной сборки приложения (Release Build) с подписью
- Б) Проведение финального тестирования и исправление критических ошибок
- В) Заполнение карточки приложения в Google Play Console (описание, скриншоты, категория)
- Г) Создание аккаунта разработчика в Google Play Console
- Д) Загрузка APK/AAB-файла и отправка на проверку (Review)
- Е) Публикация приложения после успешной проверки

Ответ:

5.4. Перечень видов оценочных средств и критерии оценивания

Контроль результатов освоения дисциплины является формой управления качеством образования.

Контроль результатов освоения дисциплины осуществляется в виде текущего контроля и промежуточной аттестации (зачета).

Система текущего контроля включает:

1. контроль работы обучающихся на лекционных и практических занятиях;
2. контроль участия в совместной работе группы;
3. контроль выполнения индивидуальных заданий;
4. контроль выполнения обучающимися заданий для самостоятельной работы.

Промежуточная аттестация осуществляется в виде зачета, который проводится в устной либо письменной форме.

Задачи промежуточной аттестации:

1. определить общий уровень освоения дисциплины в целом;
2. определить уровень основных знаний по каждой теме курса;
3. объективизировать результат контроля, минимизировать возможную субъективность преподавателя.

Критерии и рубрики оценивания Элемент контроля/Критерий/Индикаторы/Уровни (0–5)/Комментарий к уровню «5»

Практическое задание № 1 «Сравнительный анализ мобильных платформ и выбор технологии» (к теме 1.1)

Элемент контроля: Сравнительный анализ платформ, обоснование выбора стека, архитектурное решение, дорожная карта / ПК-3.3, ПК-6.5 / 0 / 3 / 5 / Таблица сравнения заполнена по всем критериям; выбор технологии аргументирован с учетом бюджета, сроков, масштабирования и интеграций; архитектура содержит все необходимые компоненты с указанием жизненного цикла; дорожная карта детализирована с реалистичными сроками; выводы обоснованы.

Практическое задание № 2 «Решение алгоритмических задач на Kotlin/Swift» (к теме 1.2)

Элемент контроля: Качество программного кода, применение ООП, Unit-тесты, соответствие Clean Code / ПК-3.3 / 0 / 3 / 5 / Классы спроектированы с применением ООП; методы реализованы без ошибок с учетом всех проверок; иерархия классов построена логично; Unit-тесты покрывают ключевые методы и граничные случаи; код соответствует Clean Code (осмысленные имена, методы короткие, нет дублирования).

Практическое задание № 3 «Создание первого проекта в Android Studio» (к теме 1.3)

Элемент контроля: Работа в Android Studio, понимание структуры проекта, настройка эмулятора, Git / ПК-3.3 / 0 / 3 / 5 / Android Studio установлена и настроена; проект создан с указанными параметрами; описано назначение всех ключевых папок и файлов; эмулятор настроен и приложение запущено; Git-репозиторий создан с корректным .gitignore; представлены скриншоты и шпаргалка по структуре проекта.

Практическое задание № 4 «Разработка приложения со сменой экранов и передачей данных» (к теме 1.4)

Элемент контроля: Создание многоэкранного приложения, управление жизненным циклом, передача данных, сохранение состояния / ПК-3.3 / 0 / 3 / 5 / Реализованы два экрана с передачей данных через Intent/Bundle; жизненный цикл отслежен через Logcat; состояние сохраняется при повороте экрана через onSaveInstanceState; навигация между экранами работает корректно; код структурирован и соответствует MVVM/MVP (при использовании).

Практическое задание № 5 «Создание адаптивного интерфейса для разных экранов» (к теме 1.5)

Элемент контроля: Верстка XML-макета, адаптивный дизайн, локализация, темная тема / ПК-3.3, ПК-6.5 / 0 / 3 / 5 / Макет сверстан с использованием ConstraintLayout; реализована адаптация под разные размеры экранов (layout-sw600dp) и ориентации (layout-land); учтены разные плотности пикселей (ресурсы mipmap); добавлена локализация (ru/en); реализована темная тема (values-night); скриншоты демонстрируют корректное отображение на разных устройствах.

Практическое задание № 6 «Разработка меню и диалогов для приложения» (к теме 1.6)

Элемент контроля: Реализация меню (опционное, контекстное, всплывающее), диалоговых окон, кастомных стилей и тем / ПК-3.3 / 0 / 3 / 5 / Реализованы все три типа меню с корректной обработкой событий; диалоговые окна (AlertDialog, диалог с вводом, DatePickerDialog) реализованы и функциональны; создана кастомная тема с измененными цветами; стили применены ко всему приложению; скриншоты демонстрируют работу всех элементов.

Практическое задание № 7 «Разработка приложения с базой данных SQLite/Room» (к теме 1.7)

Элемент контроля: Проектирование БД, реализация Room (Entity, DAO, Database), CRUD-операции, миграция, Unit-тесты / ПК-3.3 / 0 / 3 / 5 / Структура БД спроектирована корректно (2+ таблицы с внешними ключами); Entity, DAO и Database реализованы с использованием Room; CRUD-операции работают; миграция БД реализована для добавления нового поля; интерфейс позволяет добавлять, редактировать и удалять записи; написаны Unit-тесты для DAO.

Практическое задание № 8 «Интеграция приложения с REST API» (к теме 1.8)

Элемент контроля: Работа с сетью (Retrofit), парсинг JSON, асинхронность, обработка ошибок, кэширование / ПК-3.3 / 0 / 3 / 5 / API-интерфейс реализован с аннотациями Retrofit; модели данных корректно мапятся на JSON; асинхронная загрузка выполнена с использованием Coroutines; реализована обработка ошибок (отсутствие сети, неверный город); индикатор загрузки отображается; реализовано кэширование последнего запроса; скриншоты демонстрируют работу приложения.

Практическое задание № 9 «Мультимедиа и работа с датчиками» (к теме 1.9)

Элемент контроля: Работа с камерой, GPS, медиаплеером, датчиками (акселерометр) / ПК-3.3 / 0 / 3 / 5 / Реализованы запросы разрешений в рантайме; камера работает (фото сохраняется с геотегами); GPS определяет координаты и отображает на карте; аудиоплеер воспроизводит файл с управлением (Play/Pause/Stop); акселерометр отображает наклон устройства в реальном времени; все модули работают без ошибок.

Практическое задание № 10 «Тестирование, отладка и оптимизация приложения» (к теме 1.10)

Элемент контроля: Unit-тесты (JUnit, Mockito), UI-тесты (Espresso), профилирование, оптимизация, безопасность / ПК-3.3, ПК-6.5 / 0 / 3 / 5 / Написаны Unit-тесты для ViewModel/презентера с использованием Mockito; UI-тесты (Espresso) покрывают ключевые сценарии; проведен анализ профилировщика (CPU, Memory, Network), найдены и исправлены утечки памяти; реализовано безопасное хранение (EncryptedSharedPreferences); настроен Firebase Crashlytics; код обфусцирован (ProGuard/R8); представлен отчет с описанием оптимизации.

Тест

Корректность ответов на вопросы: $\geq 90\%$ правильных ответов — 5; /75–89% — 4; /60–74% — 3; /< 60% — 2.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**6.1. Рекомендуемая литература**

- Соколова, В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений : учебник для вузов / В. В. Соколова. — Москва : Издательство Юрайт, 2025. — 160 с. — (Высшее образование). — ISBN 978-5-534-16302-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561336>.
- Соколова, В. В. Разработка мобильных приложений : учебник для среднего профессионального образования / В. В. Соколова. — Москва : Издательство Юрайт, 2025. — 160 с. — (Профессиональное образование). — ISBN 978-5-534-16868-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566082>.
- Маркин, А. В. Программирование на SQL : учебник для среднего профессионального образования / А. В. Маркин. — Москва : Издательство Юрайт, 2025. — 435 с. — (Профессиональное образование). — ISBN 978-5-534-11093-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566220>.
- Тузовский, А. Ф. Проектирование и разработка web-приложений : учебник для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с. — (Высшее образование). — ISBN 978-5-534-16300-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561176>.
- Полужктова, Н. Р. Разработка веб-приложений : учебник для вузов / Н. Р. Полужктова. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 204 с. — (Высшее образование). — ISBN 978-5-534-18645-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588458>.

6.2. Электронные учебные издания и электронные образовательные ресурсы

Э1	Сетевые ресурсы Государственного автономного учреждения культуры Свердловской области «Свердловская областная универсальная научная библиотека им. В.Г. Белинского». Режим доступа: https://book.urait.ru/el_library/resursy_dostup
Э2	Образовательная платформа Юрайт. Режим доступа: https://urait.ru/
Э3	Научная библиотека Cyberleninka. Режим доступа: https://cyberleninka.ru/

6.3.1 Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства

6.3.1.1	Операционная система Astra Linux
6.3.1.2	Офисная система LibreOffice
6.3.1.3	Среда разработки с открытой лицензией (например, Android Studio — на базе IntelliJ IDEA Community Edition, Apache 2.0) для нативных приложений.
6.3.1.4	Кроссплатформенные фреймворки с открытым кодом (например, Flutter (BSD))

6.3.1.5	Контур.Класс
6.3.1.6	Контур.Толк
6.3.1.7	1С:Университет ПРОФ
6.3.1.8	Среда электронного обучения 3LK (русский Moodle)

7. МТО (оборудование и технические средства обучения)	
7.1	Учебные аудитории для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенные оборудованием и техническими средствами обучения:
7.2	Адрес (местоположение) учебных аудиторий Перечень оборудования и технических средств обучения
7.3	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 3 площадью 66,2 м² в помещении 501 Стол – 13 шт.
7.4	Стул – 25 шт.
7.5	ПК (с возможностью подключения к сети «Интернет») – 1 шт.
7.6	Телевизор (75”) – 1 шт.
7.7	Беспроводная система презентаций – 1 шт.
7.8	Спикерфон – 1 шт.
7.9	ПК (ноутбук) – 12 шт.
7.10	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 19 площадью 31,7 м² в помещении 501 Стол – 7 шт.
7.11	Стул – 13 шт.
7.12	ПК (с возможностью подключения к сети «Интернет») – 1 шт.
7.13	Магнитно-маркерная доска – 1 шт.
7.14	Телевизор (75”) – 1 шт.
7.15	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 21 площадью 35,6 м² в помещении 501 Стол – 7 шт.
7.16	Стул – 13 шт.
7.17	ПК (с возможностью подключения к сети «Интернет») – 1 шт.
7.18	Магнитно-маркерная доска – 1 шт.
7.19	Телевизор (75”) – 1 шт.
7.20	
7.21	Помещение для самостоятельной работы обучающихся, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета:
7.22	Адрес (местоположение) помещения для самостоятельной работы обучающихся Перечень компьютерной техники и оборудования
7.23	620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 5 площадью 44,3 м² в помещении 501 ПК (ноутбук) – 9 шт.
7.24	Стол – 11 шт.
7.25	Стул – 11 шт.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)
Для успешного освоения дисциплины рекомендуется ознакомиться с рабочей программой, целями и задачами дисциплины, её связями с другими дисциплинами образовательной программы. Также рекомендуется изучить основную и дополнительную литературу, перечень заданий. Перед лекцией просматривать конспект предыдущей лекции для более глубокого восприятия материала. При подготовке к практическим занятиям изучить основную литературу, ознакомиться с дополнительной и методической литературой, учесть рекомендации преподавателя. Важно также опираться на конспекты лекций. В ходе занятий вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных понятий, явлений и процессов, научные выводы и практические рекомендации по их применению. При пропуске занятия самостоятельно изучить материал и предоставить преподавателю отчёт по пропущенной теме в часы индивидуальных консультаций. При подготовке к текущей аттестации тщательно изучать материал по блокам тем, акцентировать внимание на определениях, терминах, содержании понятий. Для этого рекомендуется использовать основную литературу, а также конспекты лекций, составленные в ходе изучения всей дисциплины. При подготовке к промежуточной аттестации в дополнение к изучению конспектов лекций, словарей, учебных пособий, целесообразно пользоваться учебной литературой, рекомендованной в программе, и электронными ресурсами. При подготовке к зачёту или экзамену организовать учёбу так, чтобы все виды работ и заданий, предусмотренные рабочей программой, были выполнены в срок. Основное в подготовке — это повторение всего материала учебной дисциплины.