

**КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)
АВТОНОМНОЙ НЕКОММЕРЧЕСКОЙ ОБРАЗОВАТЕЛЬНОЙ
ОРГАНИЗАЦИИ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»**

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Специальность: 10.05.01 Компьютерная безопасность

Специализация: «Разработка защищенного программного обеспечения»

Формы обучения: очная

Объем дисциплины:

в зачетных единицах: 3 з.е.

в академических часах: 108 ак.ч.

Форма промежуточной аттестации: зачет

Оценочные материалы по дисциплине Разработка мобильных приложений

обсуждены и рекомендованы к утверждению решением кафедры гуманитарных дисциплин и иностранных языков от 21 марта 2025 г., протокол № 7

одобрены Научно-методическим советом Казанского кооперативного института (филиала) от «2» апреля 2025 г., протокол № 3.

СОДЕРЖАНИЕ

1. Паспорт оценочных материалов по дисциплине
2. Оценочные материалы по дисциплине:
 - 2.1. Оценочные материалы для проведения текущего контроля.
 - 2.2. Оценочные материалы для проведения промежуточной аттестации.

Обновление оценочных материалов дисциплины

Целью оценочных материалов по дисциплине (модулю) (далее –ОМ) является комплексная оценка результатов обучения по дисциплине Разработка мобильных приложений и установление уровня сформированности компетенций обучающегося.

ОМ предназначены для проведения текущего контроля успеваемости и промежуточной аттестации.

ОМ включают оценочные средства для проведения текущего контроля успеваемости и промежуточной аттестации.

Оценочные материалы разработаны в соответствии с рабочей программой дисциплины Разработка мобильных приложений.

1. Паспорт оценочных материалов по дисциплине «Разработка мобильных приложений»

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения	Контролируемые разделы, темы дисциплины	Наименование оценочного средства ¹	
				Текущий контроль	Промежуточная аттестация
1	2	3	4	5	6
ПК-3.2 Способен участвовать в проектировании программных и аппаратных средств защиты информации	ПК-3.2.1 Знает подходы к проектированию программных и аппаратных средств защиты информации	Знать: основы алгоритмизации и программирования Принципы построения и виды архитектуры компьютерного программного обеспечения при разработке мобильных приложений в среде Android Studio / XCode Стандарты оформления кода для используемых языков программирования на языке Java / Kotlin / Swift Уметь: применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов при разработке мобильных приложений в среде Android Studio / XCode Владеть: проектирование программных интерфейсов при разработке мобильных приложений в среде Android Studio / XCode Проектирование структур данных в среде Android Studio / XCode Разработка технической документации на компьютерное программное	Тема 1. Архитектура Android и процесс компиляции мобильного приложения. Модель MVC. Тема 2. Жизненный цикл Activity. Взаимодействие Activity. Понятие Intent Тема 3. Создание пользовательских интерфейсов с использованием макетов и виджетов Тема 4. Базовые элементы навигации. Меню. Диалоговые окна Тема 5. Списковые представления. Элементы навигации Navigation Drawer,	Реферат (Тема 1-8) Тест (тема 1-8)	Тест (П 1-10 тестового задания)

		обеспечение с использованием существующих стандартов при разработке мобильных приложений	TabHost и TabWidget, ViewPager Тема 6. UI-фрагменты и FragmentManager Тема 7. Room. LiveData. Архитектура приложения. Тема 8. Работа с базой данных SQLite		
	ПК-3.2.2 Умеет разрабатывать предложения по совершенствованию существующих методов и средств, применяемых для контроля и защиты информации и повышения эффективности этой защиты	Знать: основы алгоритмизации и программирования Стандарты оформления кода для используемых языков программирования Java / Kotlin / Swift Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения при разработке мобильных приложений в среде Android Studio / XCode Уметь: применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов при разработке мобильных приложений в среде Android Studio / XCode Владеть: проектирование программных интерфейсов при разработке мобильных приложений Проектирование структур данных при	Тема 1. Архитектура Android и процесс компиляции мобильного приложения. Модель MVC. Тема 2. Жизненный цикл Activity. Взаимодействие Activity. Понятие Intent Тема 3. Создание пользовательских интерфейсов с использованием макетов и виджетов Тема 4. Базовые элементы навигации. Меню. Диалоговые окна Тема 5. Списковые представления.	Реферат (Тема 1-8) Тест (Тема 1-8) Практическое задание (Тема 1-8)	Тест (П 11-20 тестового задания)

		разработке мобильных приложений в среде Android Studio / XCode	Элементы навигации Navigation Drawer, TabHost и TabWidget, ViewPager Тема 6. UI-фрагменты и FragmentManager Тема 7. Room. LiveData. Архитектура приложения. Тема 8. Работа с базой данных SQLite		
	ПК-3.2.3 Владеет методами и средствами разработки архитектуры и интерфейсов систем защиты информации, процедуры восстановления работоспособности и средств и систем защиты после сбоев	Знать: стандарты оформления кода для используемых языков программирования Java / Kotlin / Swift Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения при разработке мобильных приложений в среде Android Studio / XCode Уметь: применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов при разработке мобильных приложений в среде Android Studio / XCode Владеть: проектирование программных интерфейсов при разработке мобильных приложений в среде	Тема 1. Архитектура Android и процесс компиляции мобильного приложения. Модель MVC. Тема 2. Жизненный цикл Activity. Взаимодействие Activity. Понятие Intent Тема 3. Создание пользовательских интерфейсов с использованием макетов и виджетов Тема 4. Базовые элементы навигации. Меню. Диалоговые окна	Реферат (Тема 1-8) Тест (Тема 1-8) Практическое задание (Тема 1-8)	Тест (П 21-30 тестового задания)

		Android Studio / XCode Разработка, изменение архитектуры компьютерного программного обеспечения и ее согласование с системным аналитиком и архитектором программного обеспечения при разработке мобильных приложений в среде Android Studio / XCode	Тема 5. Списковые представления. Элементы навигации Navigation Drawer, TabHost и TabWidget, ViewPager Тема 6. UI-фрагменты и FragmentManager Тема 7. Room. LiveData. Архитектура приложения. Тема 8. Работа с базой данных SQLite	
--	--	--	---	--

2. Оценочные материалы по дисциплине «Разработка мобильных приложений»

2.1 Оценочные материалы для проведения текущего контроля успеваемости

Тематика рефератов

ТЕМА 1. АРХИТЕКТУРА ANDROID И ПРОЦЕСС КОМПИЛЯЦИИ

1. Архитектура платформы Android. Уровень ядра.
2. Архитектура платформы Android. Уровень библиотек.
3. Архитектура платформы Android. Уровень каркаса приложений.
4. Архитектура платформы Android. Уровень приложений.
5. Архитектура платформы Android. Dalvik Virtual Machine.
6. Среда разработки для Android. Eclipse IDE. Плагин ADT.
7. Android Virtual Device (AVD): создание, настройка и использование.
8. Android SDK. Версии SDK и Android API Level.
9. Структура проекта Android-приложения в Eclipse.
10. Файл R.java и каталоги ресурсов в Android-проекте.
11. Процесс компиляции и сборки Android-приложения (от кода до APK).
12. Модель MVC и ее аналоги (MVP, MVVM) в контексте Android.

ТЕМА 2. ЖИЗНЕННЫЙ ЦИКЛ ACTIVITY. INTENT

1. Объекты Activity и их жизненный цикл.
2. Состояния Activity и управление ими.
3. Взаимодействие между Activity.
4. Использование объектов Intent для запуска компонентов.
5. Неявные Intent и Intent-фильтры.
6. Передача данных между Activity с помощью Intent.
7. Процессы и потоки в Android при работе с Activity.
8. Задачи (Tasks) и back stack в Android.
9. Способы запуска Activity: стандартный и с ожиданием результата (startActivityForResult).
10. Восстановление состояния Activity (onSaveInstanceState, onRestoreInstanceState).
11. Сравнение Intent в Android и механизмов навигации в других мобильных ОС.
12. Обработка конфигурационных изменений (например, поворот экрана).

ТЕМА 3. ПОЛЬЗОВАТЕЛЬСКИЕ ИНТЕРФЕЙСЫ: МАКЕТЫ И ВИДЖЕТЫ

1. Графический интерфейс пользователя в Android-приложениях.
2. Создание UI с помощью XML-разметки.
3. Типы компоновок: FrameLayout.
4. Типы компоновок: LinearLayout.
5. Типы компоновок: TableLayout.
6. Типы компоновок: RelativeLayout.
7. Базовые элементы управления: TextView и его разновидности.
8. Базовые элементы управления: EditText и типы ввода текста.
9. Параметры отображения и настройки программной клавиатуры.
10. Базовые элементы управления: ImageView и работа с изображениями.
11. Создание пользовательского интерфейса программно (в коде Java/Kotlin).
12. Оптимизация производительности UI: ViewStub, Merge, Include.
13. Поддержка разных размеров экранов и плотности пикселей.

ТЕМА 4. НАВИГАЦИЯ: МЕНЮ И ДИАЛОГОВЫЕ ОКНА

1. Диалоговые окна в Android: обзор и использование.
2. Создание и настройка AlertDialog.
3. Использование ProgressDialog для отображения загрузки.
4. Работа с DatePickerDialog и TimePickerDialog.
5. Создание пользовательских (кастомных) диалоговых окон.
6. Создание и управление контекстным меню (Context Menu).
7. Создание и управление всплывающим меню (Popup Menu).
8. Опции меню (Options Menu) и панель приложения (App Bar).
9. Навигация с помощью BottomNavigationView.
10. Паттерны навигации в мобильных приложениях.
11. Использование виджета BottomAppBar.
12. Создание и использование всплывающих окон (PopupWindow).

ТЕМА 5. СПИСКОВЫЕ ПРЕДСТАВЛЕНИЯ И НАВИГАЦИЯ

1. Адаптеры данных (Adapter): понятие и использование.
2. Отображение данных в компоненте ListView.
3. Отображение данных в компоненте GridView.
4. Использование autoCompleteTextView и MultiAutoCompleteTextView.
5. Элемент навигации Navigation Drawer: создание и реализация.
6. Использование TabHost и TabWidget для организации вкладок.
7. Современный подход к вкладкам: ViewPager и TabLayout.
8. Использование ViewPager2 для организации прокручиваемых экранов.
9. Реализация бесконечной прокрутки (пагинации) в списках.
10. Оптимизация производительности списков (ViewHolder Pattern).
11. Кастомные адаптеры для сложных списков (BaseAdapter).
12. Разделители и элементы оформления в списковых представлениях.

ТЕМА 6. UI-ФРАГМЕНТЫ И FRAGMENTMANAGER

1. Понятие Фрагмента (Fragment) и его жизненный цикл.
2. Динамическое управление фрагментами с помощью FragmentManager.
3. Создание и использование статических фрагментов в XML-разметке.
4. Взаимодействие между Фрагментом и Activity.
5. Взаимодействие между двумя Фрагментами.
6. Back stack для Фрагментов и управление навигацией.
7. Создание адаптивных интерфейсов с помощью Фрагментов.
8. Диалоговые фрагменты (DialogFragment) как лучшая практика.
9. Вложенные фрагменты (Nested Fragments).
10. Использование ViewModel совместно с Фрагментами.
11. Архитектурные подходы к построению UI на фрагментах: Single Activity.
12. Анимация переходов между фрагментами.

ТЕМА 7. ROOM, LIVEDATA, АРХИТЕКТУРА ПРИЛОЖЕНИЯ

1. Библиотека Room: обзор, компоненты (Entity, DAO, Database).
2. Внедрение зависимостей в Android (Dagger/Hilt) для работы с Room.
3. Использование LiveData для реактивного программирования.
4. Наблюдаемые данные: сравнение LiveData, Flow и StateFlow.
5. Архитектурные компоненты Android (Android Architecture Components).
6. Паттерн MVVM в связке с LiveData и Room.
7. Репозиторий (Repository) как единый источник правды.
8. Работа с Room в фоновых потоках (с использованием Coroutines/RxJava).
9. Миграция баз данных в Room.

10. Создание сложных запросов с связями в Room.
11. Тестирование компонентов архитектуры (Room, ViewModel).
12. Рекомендации по архитектуре приложения от Google (Guide to App Architecture).

ТЕМА 8. РАБОТА С БАЗОЙ ДАННЫХ SQLITE

1. Файловая система Android. Работа с внутренним и внешним хранилищем.
2. Чтение и запись файлов в Android-приложениях.
3. База данных SQLite в Android: обзор и базовые операции.
4. Создание и управление базой данных с помощью SQLiteOpenHelper.
5. Выполнение CRUD-операций в SQLite (Create, Read, Update, Delete).
6. Использование чистого SQL для работы с базой данных.
7. Обертка над SQLite: сравнение Room и других ORM-библиотек.
8. Миграция схемы базы данных в SQLite.
9. Резервное копирование и восстановление базы данных SQLite.
10. Работа с транзакциями в SQLite.
11. Оптимизация производительности запросов к SQLite.
12. Безопасное хранение конфиденциальных данных в БД.

Критерии оценки выполнения рефератов по учебной дисциплине

Оценка «отлично» выставляется студенту, если тема реферата раскрыта в полной мере и все требования по написанию реферата соблюдены.

Оценка «хорошо» выставляется студенту, если недостаточно раскрыта тема реферата, но все требования по написанию реферата соблюдены.

Оценка «удовлетворительно» выставляется студенту, если недостаточно раскрыта тема реферата, не все требования по написанию реферата соблюдены, присутствуют ошибки и неточности в работе.

Оценка «неудовлетворительно» выставляется студенту, если отсутствует реферат.

Рекомендации по написанию реферата:

Реферат — письменная работа, выполняемая обучающимся в течение определенного срока.

Реферат — краткое точное изложение сущности какого-либо вопроса, темы на основе одной или нескольких книг, монографий или других первоисточников. Реферат должен содержать основные фактические сведения и выводы по рассматриваемому вопросу.

Реферат отвечает на вопрос — что содержится в данной публикации (публикациях). Реферат — не механический пересказ работы, а изложение ее сущности.

Кроме реферирования прочитанной литературы, от обучающегося требуется аргументированное изложение собственных мыслей по рассматриваемому вопросу. Тему реферата предлагает преподаватель или сам обучающийся, в последнем случае она должна быть согласована с преподавателем.

В реферате нужны развернутые аргументы, рассуждения, сравнения. Материал подается не столько в развитии, сколько в форме констатации или описания.

Содержание реферируемого произведения излагается объективно от имени автора. Если в первичном документе главная мысль сформулирована недостаточно четко, в реферате она должна быть конкретизирована и выделена.

Структура реферата:

1. Титульный лист.
2. После титульного листа на отдельной странице следует оглавление (план, содержание), в котором указаны названия всех разделов (пунктов плана) реферата и номера страниц, указывающие начало этих разделов в тексте реферата.

3. После оглавления следует введение. Объем введения составляет 1,5-2 страницы.

4. Основная часть реферата может иметь одну или несколько глав, состоящих из 2-3 параграфов (подпунктов, разделов) и предполагает осмысленное и логичное изложение главных положений и идей, содержащихся в изученной литературе. В тексте обязательны ссылки на первоисточники. В том случае если цитируется или используется чья-либо неординарная мысль, идея, вывод, приводится какой-либо цифрой материал, таблицу - обязательно сделайте ссылку на того автора у кого вы взяли данный материал.

5. Заключение содержит главные выводы, и итоги из текста основной части, в нем отмечается, как выполнены задачи и достигнуты ли цели, сформулированные во введении.

6. Приложение может включать графики, таблицы, расчеты.

7. Библиография (список литературы) здесь указывается реально использованная для написания реферата литература. Список составляется согласно правилам библиографического описания.

Этапы работы над рефератом.

Работу над рефератом можно условно подразделить на три этапа:

1. Подготовительный этап, включающий изучение предмета исследования;
2. Изложение результатов изучения в виде связного текста;
3. Устное сообщение по теме реферата.

1. Подготовительный этап работы.

Формулировка темы. Подготовительная работа над рефератом начинается с формулировки темы. Тема в концентрированном виде выражает содержание будущего текста, фиксируя как предмет исследования, так и его ожидаемый результат. Для того чтобы работа над рефератом была успешной, необходимо, чтобы тема заключала в себе проблему, скрытый вопрос (даже если наука уже давно дала ответ на этот вопрос, студент, только знакомящийся с соответствующей областью знаний, будет вынужден искать ответ заново, что даст толчок к развитию проблемного, исследовательского мышления).

Поиск источников. Грамотно сформулированная тема зафиксировала предмет изучения; задача обучающегося — найти информацию, относящуюся к данному предмету и разрешить поставленную проблему. Выполнение этой задачи начинается с поиска источников. На этом этапе необходимо вспомнить, как работать с энциклопедиями и энциклопедическими словарями.

Работа с источниками. Работу с источниками надо начинать с ознакомительного чтения, т. е. просмотреть текст, выделяя его структурные единицы. При ознакомительном чтении закладками отмечаются те страницы, которые требуют более внимательного изучения. В зависимости от результатов ознакомительного чтения выбирается дальнейший способ работы с источником. Если для разрешения поставленной задачи требуется изучение некоторых фрагментов текста, то используется метод выборочного чтения. Если в книге нет подробного оглавления, следует обратить внимание на предметные и именные указатели.

Избранные фрагменты или весь текст (если он целиком имеет отношение к теме) требуют вдумчивого, неторопливого чтения с «мысленной проработкой» материала. Такое чтение предполагает выделение: 1) главного в тексте; 2) основных аргументов; 3) выводов. Особое внимание следует обратить на то, вытекает тезис из аргументов или нет. Необходимо также проанализировать, какие из утверждений автора носят проблематичный, гипотетический характер и уловить скрытые вопросы.

Понятно, что умение таким образом работать с текстом приходит далеко не сразу. Наилучший способ научиться выделять главное в тексте, улавливать проблематичный характер утверждений, давать оценку авторской позиции — это сравнительное чтение, в ходе которого студент знакомится с различными мнениями по одному и тому же вопросу, сравнивает весомость и доказательность аргументов сторон и делает вывод о наибольшей убедительности той или иной позиции.

Создание конспектов для написания реферата.

Подготовительный этап работы завершается созданием конспектов, фиксирующих

основные тезисы и аргументы. Здесь важно вспомнить, что конспекты пишутся на одной стороне листа, с полями и достаточным для исправления и ремарок межстрочным расстоянием (эти правила соблюдаются для удобства редактирования). Если в конспектах приводятся цитаты, то непременно должно быть дано указание на источник (автор, название, выходные данные, № страницы).

По завершении предварительного этапа можно переходить непосредственно к созданию текста реферата.

2. Создание текста.

Общие требования к тексту.

Текст реферата должен подчиняться определенным требованиям: он должен раскрывать тему, обладать связностью и цельностью.

Раскрытие темы предполагает, что в тексте реферата излагается относящийся к теме материал и предлагаются пути решения содержащейся в теме проблемы; связность текста предполагает смысловую соотносительность отдельных компонентов, а цельность - смысловую законченность текста.

С точки зрения связности все тексты делятся на тексты-констатации и тексты-рассуждения. Тексты-констатации содержат результаты ознакомления с предметом и фиксируют устойчивые и несомненные суждения. В текстах-рассуждениях одни мысли извлекаются из других, некоторые ставятся под сомнение, дается им оценка, выдвигаются различные предположения.

План реферата.

Универсальный план реферата – введение, основной текст и заключение.

Требования к введению.

Во введении аргументируется актуальность исследования, - т. е. выявляется практическое и теоретическое значение данного исследования. Далее констатируется, что сделано в данной области предшественниками; перечисляются положения, которые должны быть обоснованы. Введение может также содержать обзор источников или экспериментальных данных, уточнение исходных понятий и терминов, сведения о методах исследования. Во введении обязательно формулируются цель и задачи реферата.

Объем введения - в среднем около 10% от общего объема реферата.

Основная часть реферата.

Основная часть реферата раскрывает содержание темы. Она наиболее значительна по объему, наиболее значима и ответственна. В ней обосновываются основные тезисы реферата, приводятся развернутые аргументы, предполагаются гипотезы, касающиеся существа обсуждаемого вопроса. Важно проследить, чтобы основная часть не имела форму монолога. Аргументируя собственную позицию, можно и должно анализировать, и оценивать позиции различных исследователей, с чем-то соглашаться, чему-то возражать, кого-то опровергать. Текст основной части делится на главы, параграфы, пункты. План основной части может быть составлен с использованием различных методов группировки материала: классификации (эмпирические исследования), типологии (теоретические исследования), периодизации (исторические исследования).

Заключение.

Заключение — последняя часть научного текста. В ней краткой и сжатой форме излагаются полученные результаты, представляющие собой ответ на главный вопрос исследования. Здесь же могут намечаться и дальнейшие перспективы развития темы. Небольшое по объему сообщение также не может обойтись без заключительной части - пусть это будут две-три фразы. Но в них должен подводиться итог проделанной работы.

Список использованной литературы.

Реферат любого уровня сложности обязательно сопровождается списком используемой литературы. Названия книг в списке располагают по алфавиту с указанием выходных данных использованных книг.

Требования, предъявляемые к оформлению реферата

Объемы рефератов – от 10-18 машинописных страниц. Работа выполняется на одной стороне листа стандартного формата. По обеим сторонам листа оставляются поля размером 35 мм. слева и 15 мм. справа, рекомендуется шрифт 12-14, интервал - 1,5. Все листы реферата должны быть пронумерованы. Каждый вопрос в тексте должен иметь заголовок в точном соответствии с наименованием в плане-оглавлении. При написании и оформлении реферата следует избегать типичных ошибок, например, таких:

- поверхностное изложение основных теоретических вопросов выбранной темы, когда автор не понимает, какие проблемы в тексте являются главными, а какие второстепенными,

- в некоторых случаях проблемы, рассматриваемые в разделах, не раскрывают основных аспектов выбранной для реферата темы,

- дословное переписывание книг, статей, заимствования рефератов из интернета и т. д.

При проверке реферата преподавателем оцениваются:

1. Знания и умения на уровне требований стандарта конкретной дисциплины: знание фактического материала, усвоение общих представлений, понятий, идей.

2. Характеристика реализации цели и задач исследования (новизна и актуальность поставленных в реферате проблем, правильность формулирования цели, определения задач исследования, правильность выбора методов решения задач и реализации цели; соответствие выводов решаемым задачам, поставленной цели, убедительность выводов).

3. Степень обоснованности аргументов и обобщений (полнота, глубина, всесторонность раскрытия темы, логичность и последовательность изложения материала, корректность аргументации и системы доказательств, характер и достоверность примеров, иллюстративного материала, широта кругозора автора, наличие знаний интегрированного характера, способность к обобщению).

4. Качество и ценность полученных результатов (степень завершенности реферативного исследования, спорность или однозначность выводов).

5. Использование литературных источников.

6. Культура письменного изложения материала.

7. Культура оформления материалов работы.

Комплект типовых практических заданий

ТЕМА 1. АРХИТЕКТУРА ANDROID И ПРОЦЕСС КОМПИЛЯЦИИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ. МОДЕЛЬ MVC

Задание 1: Исследование структуры Android проекта

- Создайте новый проект в Android Studio с именем "ArchitectureStudy"
- Изучите и зарисуйте схему папок и файлов проекта
- Объясните назначение каждой папки: manifests, java, res, drawable, layout, values
- Найдите файл R.java и объясните принцип его генерации
- Создайте текстовый файл с описанием назначения каждого компонента

Задание 2: Анализ файла сборки build.gradle

- Откройте файл build.gradle (Module: app)
- Изучите и опишите назначение каждой секции: plugins, android, dependencies
- Измените versionCode и versionName
- Добавьте новую зависимость в раздел dependencies
- Соберите проект и убедитесь, что изменения применились

Задание 3: Создание и настройка виртуального устройства

- Зайдите в AVD Manager и создайте 2 разных эмулятора:
 - Pixel 4 с API 30
 - Nexus 5 с API 28

- Настройте разные параметры памяти: 2GB и 4GB
- Запустите приложение на обоих эмуляторах
- Сравните производительность и внешний вид приложения

Задание 4: Исследование процесса компиляции

- Включите подробное логирование в Build Output
- Выполните чистую сборку проекта (Clean Project)
- Выполните пересборку (Rebuild Project)
- Проанализируйте этапы компиляции: компиляция Java → преобразование в dex → сборка ресурсов → создание APK
- ЗадOCUMENTИРУЙТЕ каждый этап

Задание 5: Реализация паттерна MVC в простом приложении

- Создайте приложение "Калькулятор" с архитектурой MVC:
 - Model: класс Calculator с методами сложения, вычитания
 - View: activity_main.xml с кнопками и полями ввода
 - Controller: MainActivity, обрабатывающий нажатия кнопок
- Реализуйте операции: +, -, *, /
- Убедитесь, что бизнес-логика отделена от интерфейса

Задание 6: Анализ APK файла

- Соберите signed APK
- Используйте APK Analyzer для изучения содержимого
- Определите размер различных компонентов приложения
- Найдите скомпилированные ресурсы и код
- Сделайте выводы об оптимизации размера приложения

ТЕМА 2. ЖИЗНЕННЫЙ ЦИКЛ ACTIVITY. ВЗАИМОДЕЙСТВИЕ ACTIVITY. ПОНЯТИЕ INTENT

Задание 1: Создание журнала жизненного цикла Activity

- Создайте приложение "LifecycleLogger"
- Переопределите ВСЕ методы жизненного цикла в MainActivity
- Добавьте запись в LogCat при вызове каждого метода с тегом "LIFECYCLE"
- Протестируйте сценарии: запуск, сворачивание, разворачивание, поворот экрана, завершение
- Создайте таблицу с наблюдаемыми состояниями

Задание 2: Сохранение и восстановление состояния

- Создайте счетчик нажатий кнопки
- Реализуйте сохранение значения счетчика при повороте экрана
- Используйте методы onSaveInstanceState() и onRestoreInstanceState()
- Добавьте TextView для отображения текущего значения
- Убедитесь, что значение сохраняется при любых изменениях конфигурации

Задание 3: Создание многоэкранного приложения

- Создайте 3 Activity: Главная, Настройки, О программе
- Реализуйте навигацию между ними с помощью Intent
- Добавьте кнопку "Назад" на второстепенных экранах
- Настройте анимации переходов между Activity
- Проверьте работу стека навигации

Задание 4: Передача данных между Activity

- Создайте форму регистрации на MainActivity (имя, email, возраст)
- Реализуйте передачу данных на SecondActivity через Intent
- Используйте putExtra() для передачи различных типов данных
- На SecondActivity отобразите полученные данные в TextView
- Добавьте валидацию вводимых данных

Задание 5: Работа с неявными Intent

- Создайте приложение с кнопками для выполнения системных действий:
 - Открытие веб-страницы
 - Отправка email
 - Открытие карт с определенными координатами
 - Совершение телефонного звонка
- Обработайте случаи, когда приложение для действия не найдено

Задание 6: Запуск Activity с возвратом результата

- Создайте приложение "Выбор цвета"
- MainActivity содержит кнопку и View с изменяемым цветом фона
- SecondActivity содержит палитру цветов (кнопки с разными цветами)
- Реализуйте возврат выбранного цвета на MainActivity
- Используйте startActivityForResult() и onActivityResult()

ТЕМА 3. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ С ИСПОЛЬЗОВАНИЕМ МАКЕТОВ И ВИДЖЕТОВ

Задание 1: Создание комплексной формы заказа

- Разработайте макет для формы заказа пиццы
- Используйте LinearLayout с вертикальной ориентацией
- Элементы формы:
 - Выбор размера пиццы (RadioGroup с RadioButton)
 - Выбор топпингов (CheckBox для каждого ингредиента)
 - Поле для особых пожеланий (Multi-line EditText)
 - Кнопка "Заказать"
- Настройте правильные отступы и выравнивание

Задание 2: Адаптивный интерфейс с RelativeLayout

- Создайте профиль пользователя с использованием RelativeLayout
- Элементы:
 - Аватар (ImageView) в центре верхней части
 - Имя пользователя под аватаром
 - Статистика (друзья, фото, записи) внизу экрана
 - Кнопка "Редактировать" в правом верхнем углу
- Убедитесь, что макет корректно отображается на разных размерах экранов

Задание 3: Табличный макет для расписания

- Создайте расписание занятий на неделю с помощью TableLayout
- Столбцы: Дни недели (Пн-Пт)
- Строки: Время пар (8:00-9:30, 10:00-11:30 и т.д.)
- Заполните ячейки названиями предметов
- Добавьте чередование цветов строк для лучшей читаемости

Задание 4: Кастомные стили и темы

- Создайте собственную тему приложения:
 - Определите цветовую палитру в colors.xml
 - Создайте стили для кнопок, текстовых полей, заголовков
 - Реализуйте светлую и темную тему
 - Добавьте переключатель тем в настройки приложения
- Убедитесь, что все элементы используют кастомные стили

Задание 5: Интерактивный опросник

- Создайте приложение для проведения опросов
- Динамическое создание вопросов с разными типами ответов:
 - Один вариант (RadioGroup)
 - Несколько вариантов (CheckBox)
 - Текстовый ответ (EditText)
 - Оценка по шкале (SeekBar)

- Реализуйте навигацию между вопросами
- **Задание 6: Валидация пользовательского ввода**
- Создайте форму регистрации с валидацией:
 - Email (проверка формата)
 - Пароль (минимальная длина, обязательные символы)
 - Телефон (проверка формата)
 - Дата рождения (проверка возраста)
- Добавьте отображение ошибок под полями ввода
- Реализуйте подсветку полей при ошибке

ТЕМА 4. БАЗОВЫЕ ЭЛЕМЕНТЫ НАВИГАЦИИ. МЕНЮ. ДИАЛОГОВЫЕ ОКНА

Задание 1: Создание комплексной системы меню

- Реализуйте все типы меню в одном приложении:
 - Options Menu (в Toolbar)
 - Context Menu (долгое нажатие на элементе списка)
 - Popup Menu (при нажатии на кнопку "Еще")
- Добавьте иконки для пунктов меню
- Реализуйте обработку выбора для каждого пункта

Задание 2: Кастомные диалоговые окна

- Создайте 5 различных типов диалогов:
 - AlertDialog с списком выбора
 - ProgressDialog с индикатором загрузки
 - DatePickerDialog для выбора даты
 - TimePickerDialog для выбора времени
 - Кастомный диалог с собственной разметкой
- Реализуйте передачу данных из диалогов в Activity

Задание 3: Система уведомлений в приложении

- Создайте службу уведомлений:
 - Toast для кратковременных сообщений
 - Snackbar с действиями
 - Status Bar уведомления
 - Диалоги подтверждения важных действий
- Реализуйте различные сценарии использования каждого типа

Задание 4: Bottom Navigation с Fragment

- Создайте приложение с нижней навигацией
- 3 раздела: Главная, Поиск, Профиль
- Каждый раздел - отдельный Fragment
- Реализуйте плавные переходы между разделами
- Сохраняйте состояние каждого Fragment

Задание 5: Navigation Drawer с заголовком

- Создайте боковое выдвижное меню
- Добавьте заголовок с информацией пользователя
- Реализуйте разделы меню с группами и подпунктами
- Добавьте иконки для каждого пункта меню
- Реализуйте подсветку текущего раздела

Задание 6: Комплексная система диалогов для приложения "Заметки"

- Создайте приложение для управления заметками
- Диалоги для:
 - Создания новой заметки
 - Редактирования существующей
 - Удаления с подтверждением

- Выбор категории
 - Настройки сортировки
- Реализуйте передачу данных между диалогами и основным экраном

ТЕМА 5. СПИСКОВЫЕ ПРЕДСТАВЛЕНИЯ. ЭЛЕМЕНТЫ НАВИГАЦИИ NAVIGATION DRAWER, TABHOST И TABWIDGET, VIEWPAGER

Задание 1: Реализация многоуровневого списка

- Создайте ExpandableListView с группами и элементами
- Группы: категории товаров (электроника, одежда, книги)
- Элементы: конкретные товары в каждой категории
- Добавьте иконки для групп и элементов
- Реализуйте обработку кликов на разных уровнях

Задание 2: Кастомный адаптер для списка контактов

- Создайте ListView с кастомным макетом элемента
- Каждый элемент содержит:
 - Аватар (ImageView)
 - Имя контакта (TextView)
 - Статус онлайн/офлайн (индикатор)
 - Кнопка вызова

- Реализуйте ViewHolder pattern для оптимизации

Задание 3: Реализация ViewPager с фрагментами

- Создайте приложение с 3 вкладками через ViewPager
- Каждая вкладка - отдельный Fragment
 - Вкладка 1: Список задач
 - Вкладка 2: Статистика
 - Вкладка 3: Настройки

- Добавьте TabLayout для индикации текущей вкладки
- Реализуйте синхронизацию данных между вкладками

Задание 4: Navigation Drawer с динамическим контентом

- Создайте боковое меню с разными типами элементов:
 - Заголовок с информацией пользователя
 - Основные разделы приложения
 - Настройки
 - Выход

- Реализуйте загрузку данных меню из ресурсов
- Добавьте подсветку активного раздела

Задание 5: Комбинированная навигация (Tabs + Drawer)

- Создайте приложение с одновременным использованием:
 - Bottom Navigation для основных разделов
 - Navigation Drawer для дополнительных функций
 - ViewPager для подразделов
- Реализуйте согласованное поведение всех элементов навигации

Задание 6: Бесконечная прокрутка и поиск в списке

- Создайте список с пагинацией (подгрузка при прокрутке)
- Реализуйте поиск с фильтрацией данных
- Добавьте сортировку по различным параметрам
- Реализуйте "pull to refresh" для обновления данных
- Добавьте анимации загрузки и обновления

ТЕМА 6. UI-ФРАГМЕНТЫ И FRAGMENTMANAGER

Задание 1: Динамическое управление фрагментами

- Создайте Activity с контейнером для фрагментов

- Реализуйте добавление, замену, удаление фрагментов
 - Используйте FragmentTransaction с анимациями
 - Добавьте поддержку back stack для навигации
 - Продемонстрируйте различные способы добавления фрагментов
- Задание 2: Адаптивный интерфейс для планшетов и телефонов**

- Создайте макеты для разных конфигураций:
 - Телефон: один фрагмент на экране
 - Планшет: два фрагмента рядом (master-detail)
- Используйте qualifiers для ресурсов
- Реализуйте различное поведение для разных размеров экранов

Задание 3: Взаимодействие между фрагментами

- Создайте 2 фрагмента: список и детали
- Реализуйте интерфейс для коммуникации через Activity
- При выборе элемента в списке обновляйте детали
- Используйте ViewModel для общих данных
- Реализуйте обработку различных состояний

Задание 4: Fragment с собственной навигацией

- Создайте фрагмент с внутренним ViewPager
- Реализуйте вложенную навигацию
- Управляйте back stack для внутренней навигации
- Синхронизируйте состояние с родительским Activity

Задание 5: Диалоговые фрагменты (DialogFragment)

- Создайте кастомные DialogFragment для:
 - Выбора даты и времени
 - Подтверждения действий
 - Ввода пользовательских данных
 - Отображения прогресса

- Реализуйте передачу данных в обе стороны

Задание 6: Сохранение состояния фрагментов

- Реализуйте сохранение и восстановление состояния:
 - При повороте экрана
 - При переходе между фрагментами
 - При пересоздании Activity
- Используйте различные подходы: Bundle, ViewModel, сохраненные инстансы

ТЕМА 7. ROOM. LIVEDATA. АРХИТЕКТУРА ПРИЛОЖЕНИЯ

Задание 1: Создание полной архитектуры приложения с Room

- Реализуйте приложение "Заметки" с Clean Architecture:
 - Entity: Note (id, title, content, date)
 - DAO: методы для CRUD операций
 - Repository: абстракция над источником данных
 - ViewModel: предоставление данных для UI
 - Activity/Fragment: отображение данных

Задание 2: Работа с отношениями в Room

- Создайте связанные сущности:
 - User (id, name) и Post (id, userId, content)
- Реализуйте one-to-many отношение
- Создайте запросы с JOIN для получения данных
- Реализуйте каскадное удаление

Задание 3: Наблюдаемые запросы с LiveData

- Создайте DAO методы, возвращающие LiveData
- Реализуйте автоматическое обновление UI при изменении данных

- Добавьте преобразование данных с помощью Transformations
- Реализуйте пагинацию с LiveData

Задание 4: Миграция базы данных

- Создайте первоначальную версию БД с 2 таблицами
- Добавьте миграцию для новой версии с изменениями:
 - Добавление новой таблицы
 - Добавление столбца в существующую таблицу
 - Изменение типа данных столбца
- Протестируйте миграцию на существующих данных

Задание 5: Репозиторий с несколькими источниками данных

- Создайте репозиторий, который работает с:
 - Room (локальная БД)
 - Retrofit (удаленный сервер)
 - SharedPreferences (настройки)

- Реализуйте стратегию кэширования
- Обеспечьте согласованность данных

Задание 6: Тестирование архитектурных компонентов

- Напишите unit тесты для:
 - DAO методов
 - Repository
 - ViewModel
- Используйте тестовую базу данных
- Реализуйте тесты для LiveData

ТЕМА 8. РАБОТА С БАЗОЙ ДАННЫХ SQLITE

Задание 1: Создание и управление базой данных

- Реализуйте SQLiteOpenHelper для приложения "Библиотека"
- Таблицы: Books, Authors, Categories
- Создайте отношения между таблицами
- Реализуйте методы для инициализации начальными данными

Задание 2: Комплексные SQL запросы

- Напишите запросы для:
 - Поиска книг по автору с JOIN
 - Статистики по категориям с GROUP BY
 - Поиска с полнотекстовым поиском
 - Сортировки по различным параметрам
- Реализуйте пагинацию на уровне SQL

Задание 3: Транзакции и целостность данных

- Реализуйте операции, требующие транзакций:
 - Добавление книги с автором
 - Обновление нескольких связанных записей
 - Удаление с каскадным обновлением
- Обеспечьте целостность данных при ошибках

Задание 4: Резервное копирование и восстановление

- Реализуйте экспорт базы данных в файл
- Создайте функционал импорта из backup
- Добавьте валидацию данных при восстановлении
- Реализуйте инкрементальное 备份

Задание 5: Оптимизация производительности

- Проанализируйте производительность запросов
- Создайте индексы для часто используемых полей
- Реализуйте кэширование часто запрашиваемых данных

- Оптимизируйте работу с Cursor
- **Задание 6: Миграция с SQLite на Room**
- Возьмите существующий проект с SQLite
- Реализуйте миграцию на Room
- Сохраните существующие данные
- Сравните производительность и удобство использования

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Типовые тесты

ТЕМА 1: АРХИТЕКТУРА ANDROID И ПРОЦЕСС КОМПИЛЯЦИИ

1. **Какой компонент архитектуры Android отвечает за управление памятью и драйверами устройств?**
 - a) Уровень библиотек
 - b) Уровень приложений
 - c) Уровень ядра Linux [✓]
 - d) Уровень каркаса приложений
2. **Что такое Dalvik Virtual Machine?**
 - a) Эмулятор для тестирования приложений
 - b) Виртуальная машина, на которой выполнялись приложения до Android 5.0 [✓]
 - c) Система управления базами данных
 - d) Тип компилятора для сборки APK
3. **Файл R.java...**
 - a) Содержит исходный код активности
 - b) Генерируется автоматически и содержит ID ресурсов [✓]
 - c) Это файл манифеста приложения
 - d) Хранит настройки компилятора
4. **Что такое AVD?**
 - a) Android Virtual Device — эмулятор Android-устройства [✓]
 - b) Android Version Director — менеджер версий SDK
 - c) Automatic View Designer — инструмент для создания UI
 - d) Application Verification Dialog — диалог проверки приложения
5. **Какова роль плагина ADT для Eclipse?**
 - a) Обеспечивает поддержку языка Swift
 - b) Добавляет в Eclipse инструменты для разработки под Android [✓]

- c) Это альтернативное название для Android SDK
- d) Используется для создания иконок приложения
- 6. **Уровень каркаса приложений (Application Framework) предоставляет разработчикам...**
 - a) Доступ к аппаратным датчикам напрямую
 - b) Исходный код ядра Linux
 - c) API-интерфейсы для создания приложений [✓]
 - d) Готовые скомпилированные APK-файлы
- 7. **Процесс компиляции Android-приложения заканчивается созданием файла...**
 - a) .JAVA
 - b) .CLASS
 - c) .DEX [✓]
 - d) .XML
- 8. **Модель MVC в Android условно можно представить как:**
 - a) Activity (View), XML-разметка (Controller), данные (Model)
 - b) Activity (Controller), XML-разметка (View), данные (Model) [✓]
 - c) Activity (Model), XML-разметка (View), данные (Controller)
 - d) Activity (Model), XML-разметка (Controller), данные (View)
- 9. **Android API Level — это...**
 - a) Уровень заряда батареи устройства
 - b) Уникальный идентификатор приложения в магазине
 - c) Целое число, уникально идентифицирующее версию платформы API [✓]
 - d) Количество функций в приложении
- 10. **Какой каталог в структуре проекта Android содержит строковые ресурсы?**
 - a) res/layout/
 - b) res/drawable/
 - c) res/values/ [✓]
 - d) res/raw/

ТЕМА 2: ЖИЗНЕННЫЙ ЦИКЛ ACTIVITY. INTENT

- 1. **Первый метод, вызываемый при запуске Activity?**
 - a) onStart()
 - b) onResume()
 - c) onCreate() [✓]
 - d) onRestart()
- 2. **Метод onPause() вызывается, когда...**
 - a) Activity становится полностью видимым пользователю
 - b) Activity запускается впервые
 - c) Activity теряет фокус, но частично видна (например, из-за диалога) [✓]
 - d) Приложение завершает свою работу
- 3. **Для запуска другой Activity используется...**
 - a) startService()
 - b) startActivity() [✓]
 - c) bindService()
 - d) loadActivity()
- 4. **Что такое Intent?**
 - a) Объект для описания операции, которую нужно выполнить [✓]
 - b) Модель данных для списка
 - c) Тип макета для размещения виджетов
 - d) База данных приложения

5. **Какой метод используется для передачи данных в запускаемую Activity?**
 - a) putParameter()
 - b) addData()
 - c) putExtra() [✓]
 - d) setArgument()
6. **Неявный Intent...**
 - a) Всегда указывает на конкретный класс Activity
 - b) Описывает действие, которое нужно выполнить (например, отправку email) [✓]
 - c) Используется только для запуска служб
 - d) Не требует разрешений в манифесте
7. **Метод onDestroy() гарантированно вызывается...**
 - a) При повороте экрана
 - b) Когда Activity уничтожается системой или пользователем [✓]
 - c) Каждый раз, когда пользователь нажимает кнопку "Назад"
 - d) При переходе на другую Activity в том же приложении
8. **Для получения результата из запущенной Activity используется метод...**
 - a) startActivityForResult() [✓]
 - b) getActivityResult()
 - c) launchActivityWithResult()
 - d) onActivityResult() (это метод обратного вызова, а не запуска)
9. **Intent-фильтры объявляются в...**
 - a) Файле R.java
 - b) Файле build.gradle
 - c) Файле AndroidManifest.xml [✓]
 - d) Исходном коде Activity
10. **Состояние Activity между onStart() и onPause() называется...**
 - a) Stopped
 - b) Created
 - c) Resumed
 - d) Started [✓]

ТЕМА 3: СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

1. **Какой макет используется для расположения элементов друг за другом по вертикали или горизонтали?**
 - a) FrameLayout
 - b) LinearLayout [✓]
 - c) RelativeLayout
 - d) TableLayout
2. **Какой атрибут в EditText определяет тип вводимого содержимого?**
 - a) android:inputType [✓]
 - b) android:textType
 - c) android:inputStyle
 - d) android:contentType
3. **Какой макет позволяет позиционировать элементы относительно друг друга или родителя?**
 - a) LinearLayout
 - b) RelativeLayout [✓]
 - c) FrameLayout
 - d) GridLayout
4. **Элемент TextView предназначен для...**
 - a) Ввода текста пользователем

- b) Отображения неизменяемого текста [✓]
 - c) Отображения изображений
 - d) Создания кнопок
5. **Какой атрибут LinearLayout задает ориентацию элементов?**
- a) android:orientation [✓]
 - b) android:direction
 - c) android:arrangement
 - d) android:layout_gravity
6. **Для чего используется элемент ImageView?**
- a) Для просмотра видео
 - b) Для отображения изображений [✓]
 - c) Для ввода пароля
 - d) Для создания фона активности
7. **Атрибут android:layout_weight используется в...**
- a) RelativeLayout
 - b) FrameLayout
 - c) LinearLayout [✓]
 - d) AbsoluteLayout
8. **Какой метод используется в коде Activity, чтобы найти виджет по ID?**
- a) getView()
 - b) findViewById() [✓]
 - c) getWidget()
 - d) inflateView()
9. **Какой макет по умолчанию помещает все дочерние элементы в левый верхний угол?**
- a) LinearLayout
 - b) RelativeLayout
 - c) FrameLayout [✓]
 - d) TableLayout
10. **Атрибут android:hint в EditText задает...**
- a) Текст ошибки при неверном вводе
 - b) Подсказку, которая исчезает при вводе текста [✓]
 - c) Значение по умолчанию для поля
 - d) Название поля для доступности

ТЕМА 4: БАЗОВЫЕ ЭЛЕМЕНТЫ НАВИГАЦИИ. ДИАЛОГИ

1. **Какой класс используется для создания стандартного диалогового окна с сообщением и кнопками?**
- a) Dialog
 - b) ProgressDialog
 - c) AlertDialog [✓]
 - d) Toast
2. **Для отображения простого всплывающего сообщения используется...**
- a) AlertDialog
 - b) Snackbar
 - c) Toast [✓]
 - d) Notification
3. **Какой метод AlertDialog.Builder используется для задания заголовка диалога?**
- a) .setMessage()
 - b) .setTitle() [✓]

- c) .setHeader()
- d) .setPositiveButton()
- 4. Для создания диалога выбора даты используется...
 - a) TimePickerDialog
 - b) DatePickerDialog [✓]
 - c) CalendarDialog
 - d) ChoiceDialog
- 5. Что такое ProgressDialog?
 - a) Диалог для отображения прогресса выполнения операции [✓]
 - b) Диалог для настройки параметров приложения
 - c) Диалог для выбора файла
 - d) Диалог для ввода текста
- 6. Контекстное меню (Context Menu) вызывается...
 - a) Долгим нажатием на элемент [✓]
 - b) Нажатием на кнопку "Меню" на устройстве
 - c) Свайпом сверху вниз
 - d) Всегда отображается на экране
- 7. Чтобы показать диалог, созданный через DialogFragment, используется метод...
 - a) .show()
 - b) .display()
 - c) .start()
 - d) .show() менеджера фрагментов [✓] (Конкретно: show(FragmentManager, String))
- 8. Кнопка "NEGATIVE" в AlertDialog обычно используется для...
 - a) Подтверждения действия
 - b) Отмены действия [✓]
 - c) Перехода к следующему шагу
 - d) Вызова справки
- 9. Опции меню (Options Menu) traditionally размещаются в...
 - a) В нижней части экрана
 - b) В выдвижной панели (Navigation Drawer)
 - c) В панели действий (Action Bar / App Bar) [✓]
 - d) В контекстном меню
- 10. Для создания кастомного диалога используется...
 - a) Только стандартные билдеры AlertDialog
 - b) Свой макет, который передается в AlertDialog.Builder через .setView() [✓]
 - c) Нельзя создать кастомный диалог
 - d) Только наследование от класса Window

ТЕМА 5: СПИСКОВЫЕ ПРЕДСТАВЛЕНИЯ

- 1. Какой компонент используется для отображения прокручиваемого списка элементов?
 - a) ScrollView
 - b) RecyclerView (лучшая практика) или ListView [✓]
 - c) GridView
 - d) ListActivity
- 2. Что такое Adapter в контексте ListView?
 - a) Это макет для каждого элемента списка
 - b) Это мост между данными и списком, который их отображает [✓]
 - c) Это анимация прокрутки списка
 - d) Это тип данных для хранения списка строк

3. Для отображения данных в виде сетки используется...
 - a) ListView
 - b) RecyclerView с GridLayoutManager или GridView [✓]
 - c) TableLayout
 - d) StackView
4. Что такое ViewHolder Pattern?
 - a) Паттерн для хранения данных списка
 - b) Паттерн для кэширования ссылок на View элементов списка и повышения производительности [✓]
 - c) Паттерн для создания анимаций в списке
 - d) Паттерн для сортировки данных в адаптере
5. Navigation Drawer — это...
 - a) Панель вкладок в верхней части экрана
 - b) Выдвижная панель меню сбоку экрана [✓]
 - c) Нижняя панель навигации
 - d) Диалоговое окно с опциями
6. Какой компонент позволяет организовать интерфейс с помощью свайпа между экранами?
 - a) TabHost
 - b) ViewPager [✓]
 - c) DrawerLayout
 - d) ScrollView
7. Для чего используется AutoCompleteTextView?
 - a) Для ввода многострочного текста
 - b) Для ввода текста с подсказками в виде выпадающего списка [✓]
 - c) Для скрытия введенного пароля
 - d) Для отображения ссылок
8. Современная замена TabHost — это связка...
 - a) ViewPager и TabLayout [✓]
 - b) ListView и Adapter
 - c) DrawerLayout и NavigationView
 - d) LinearLayout и Button
9. Метод getView() в BaseAdapter вызывается для...
 - a) Получения всех данных списка
 - b) Создания или переиспользования View для каждого элемента списка [✓]
 - c) Получения количества элементов в списке
 - d) Обработки нажатия на элемент списка
10. Какой метод RecyclerView.Adapter уведомляет список об изменении данных?
 - a) invalidate()
 - b) refreshData()
 - c) notifyDataSetChanged() [✓]
 - d) updateList()

ТЕМА 6: UI-ФРАГМЕНТЫ И FRAGMENTMANAGER

1. Основное преимущество использования Фрагментов — это...
 - a) Увеличение скорости работы приложения
 - b) Создание модульных и переиспользуемых компонентов UI для разных размеров экранов [✓]
 - c) Автоматическая работа с сетью
 - d) Упрощение работы с базой данных

2. Для добавления Фрагмента в Activity используется...
 - a) FragmentManager и FragmentTransaction [✓]
 - b) FragmentContainer
 - c) LayoutInflater
 - d) ViewGroup
3. Жизненный цикл Фрагмента тесно связан с жизненным циклом...
 - a) Приложения (Application)
 - b) Службы (Service)
 - c) Активности (Activity), в которой он находится [✓]
 - d) Другого Фрагмента
4. Чтобы добавить Фрагмент в back stack, используется метод...
 - a) addToBackStack() [✓]
 - b) pushToBackStack()
 - c) saveToBackStack()
 - d) commitToBackStack()
5. Для связи Фрагмента с его макетом используется метод...
 - a) onCreate()
 - b) onCreateView() [✓]
 - c) onActivityCreated()
 - d) onAttach()
6. Рекомендуемый способ взаимодействия между Фрагментом и его Activity — это...
 - a) Прямой вызов методов Activity
 - b) Использование интерфейсов-обратных вызовов (callbacks) [✓]
 - c) Использование статических переменных
 - d) Через Intent
7. DialogFragment используется для...
 - a) Замены устаревших диалоговых классов [✓]
 - b) Создания новых Activity
 - c) Отображения полноэкранных окон
 - d) Управления другими Фрагментами
8. Статический способ добавления Фрагмента подразумевает...
 - a) Добавление тега <fragment> в XML-разметку активности [✓]
 - b) Использование FragmentTransaction в коде
 - c) Создание Фрагмента только программно
 - d) Использование FragmentManager из сервиса
9. Метод findFragmentById() используется для...
 - a) Поиска Фрагмента по его ID в макете [✓]
 - b) Поиска Фрагмента по тегу
 - c) Получения всех Фрагментов в активности
 - d) Создания нового экземпляра Фрагмента
10. При повороте экрана Фрагмент...
 - a) Уничтожается и создается заново вместе с Activity [✓]
 - b) Остается в памяти без изменений
 - c) Автоматически сохраняет все свои данные без дополнительных усилий
 - d) Игнорирует поворот экрана

ТЕМА 7: ROOM, LIVEDATA, АРХИТЕКТУРА

1. Room — это...
 - a) Библиотека для создания пользовательского интерфейса
 - b) Надстройка над SQLite, предоставляющая абстракцию над базой данных [✓]

- c) Система для управления памятью
- d) Тип макета для комнат в приложении
- 2. **Основные компоненты Room — это...**
 - a) Activity, Fragment, ViewModel
 - b) Entity, DAO, Database [✓]
 - c) Repository, LiveData, Adapter
 - d) Model, View, Controller
- 3. **LiveData — это...**
 - a) Класс для хранения данных в оперативной памяти
 - b) Наблюдаемый класс, учитывающий жизненный цикл компонентов [✓]
 - c) Библиотека для сетевых запросов
 - d) Тип базы данных
- 4. **Преимущество LiveData заключается в...**
 - a) Автоматическом обновлении UI только тогда, когда данные активны и актуальны [✓]
 - b) Увеличении скорости работы приложения в 2 раза
 - c) Возможности работать без базы данных
 - d) Автоматической записи данных в сеть
- 5. **Класс, обозначенный аннотацией @Entity в Room, представляет...**
 - a) Таблицу в базе данных [✓]
 - b) Запрос к базе данных
 - c) Саму базу данных
 - d) Модель представления (View Model)
- 6. **Паттерн MVVM расшифровывается как...**
 - a) Model-View-ViewModel [✓]
 - b) Main-View-Visual-Model
 - c) Multiple-Views-Variable-Models
 - d) Model-View-Variable-Method
- 7. **Роль Repository в рекомендованной архитектуре Android — это...**
 - a) Управление жизненным циклом Activity
 - b) Единый источник данных для всего приложения (абстракция над источниками данных) [✓]
 - c) Отображение данных на экране
 - d) Хранение строковых ресурсов
- 8. **Аннотация @Insert в Room DAO используется для...**
 - a) Выбора данных из таблицы
 - b) Вставки новых данных в таблицу [✓]
 - c) Обновления существующих данных
 - d) Удаления данных из таблицы
- 9. **ViewModel предназначена для...**
 - a) Хранения и управления UI-данными, устойчивыми к изменениям конфигурации (например, повороту экрана) [✓]
 - b) Создания разметки экрана
 - c) Непосредственного взаимодействия с базой данных
 - d) Запуска сетевых запросов
- 10. **Для выполнения операций Room в фоновом потоке рекомендуется использовать...**
 - a) Главный поток (Main Thread)
 - b) Корутины (Coroutines) или RxJava [✓]
 - c) AsyncTask (устарел)
 - d) Поток UI (UI Thread)

ТЕМА 8: РАБОТА С SQLITE

1. **Класс SQLiteOpenHelper используется для...**
 - a) Выполнения простых SQL-запросов
 - b) Создания и управления версиями базы данных [✓]
 - c) Отображения данных из БД в списке
 - d) Шифрования базы данных
2. **Метод onCreate() в SQLiteOpenHelper вызывается, когда...**
 - a) База данных открывается для записи
 - b) База данных не существует и ее нужно создать [✓]
 - c) Версия базы данных увеличилась
 - d) Приложение запускается
3. **CRUD означает...**
 - a) Create, Read, Update, Delete [✓]
 - b) Copy, Remove, Undo, Delete
 - c) Create, Remove, Undo, Deploy
 - d) Calculate, Read, Use, Deploy
4. **Для обновления данных в SQLite используется метод...**
 - a) insert()
 - b) update() [✓]
 - c) execSQL()
 - d) query()
5. **Какой класс представляет результат запроса к базе данных SQLite?**
 - a) SQLiteDatabase
 - b) Cursor [✓]
 - c) ContentValues
 - d) SQLiteQuery
6. **Класс ContentValues используется для...**
 - a) Хранения результатов запроса
 - b) Добавления или обновления строк в таблице (хранит пары ключ-значение) [✓]
 - c) Создания самой базы данных
 - d) Удаления таблиц
7. **Операции с базой данных SQLite не следует выполнять в основном потоке, потому что...**
 - a) Это может заблокировать UI и вызвать "ANR" (Application Not Responding) [✓]
 - b) База данных работает только в фоновом режиме
 - c) Это запрещено Google Play
 - d) Главный поток быстрее для операций с БД
8. **Метод onUpgrade() в SQLiteOpenHelper вызывается, когда...**
 - a) База данных создается впервые
 - b) Нужно сделать резервную копию данных
 - c) Версия базы данных, указанная в коде, больше версии на устройстве [✓]
 - d) Приложение обновляется в магазине
9. **Для выполнения произвольного SQL-запроса (например, DELETE) без возврата результата используется...**
 - a) rawQuery()
 - b) execSQL() [✓]
 - c) query()
 - d) delete()
10. **База данных SQLite, созданная приложением, по умолчанию хранится в...**
 - a) Внешнем хранилище (SD-карта)

- b) Внутреннем хранилище приложения, недоступном другим приложениям [✓]
- c) Облачном хранилище Google
- d) Оперативной памяти (RAM)

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

2.2 Оценочные материалы для проведения промежуточной аттестации

ПК-3.2.1 Знает подходы к проектированию программных и аппаратных средств защиты информации

П1. Какой подход к проектированию защиты информации в мобильном приложении является наиболее фундаментальным?

- a) Шифрование всех передаваемых данных;
- b) Реализация аутентификации по отпечатку пальца;
- в) Принцип минимальных привилегий (запрос только необходимых разрешений); [✓]
- г) Использование HTTPS для сетевых запросов.

Ответ: в) Принцип минимальных привилегий

П2. Что из перечисленного относится к аппаратным средствам защиты информации на мобильном устройстве?

- a) Защищенное хранилище ключей (KeyStore/KeyChain); [✓]
- б) Библиотека SQLCipher для шифрования БД;
- в) Использование алгоритма AES-256;
- г) Хэширование паролей.

Ответ: а) Защищенное хранилище ключей

П3. Какие элементы архитектуры Android OS напрямую способствуют защите информации на уровне системы?

- a) Межпроцессное взаимодействие (IPC);
- б) Механизм разрешений (Permissions) и песочница (Sandbox) для приложений; [✓]
- в) Наличие Activity и Fragment;
- г) Использование XML для разметки UI.

Ответ: б) Механизм разрешений и песочница

П4. При проектировании системы хранения чувствительных данных в мобильном приложении необходимо учитывать:

- a) Удобство интерфейса для быстрого доступа;
- б) Производительность приложения в офлайн-режиме;
- в) Скорость работы алгоритмов шифрования;
- г) Риски утечки данных и способы их минимизации (шифрование, безопасное место хранения). [✓]

Ответ: г) Риски утечки данных и способы их минимизации

П5. Установите соответствие между угрозой безопасности и подходом к ее парированию на этапе проектирования:

Угроза	Подход к защите
1. Перехват сетевого трафика	а) Шифрование данных на устройстве перед отправкой, использование TLS (HTTPS).
2. Неавторизованный доступ к данным на устройстве	б) Использование системного KeyStore для хранения криптографических ключей, шифрование локальной БД.
3. Утечка данных через логи приложения	в) Запрет логирования чувствительной информации, использование ProGuard для обфускации кода.

Правильный ответ:

1 – а, 2 – б, 3 – в

П6. Установите соответствие между компонентом ОС Android и его ролью в защите информации:

Компонент / Механизм	Роль в защите
1. Sandbox (Песочница)	а) Изоляция данных и кода каждого приложения от других приложений и системы.
2. Permission Model (Модель разрешений)	б) Контроль доступа приложений к защищенным API и пользовательским данным.
3. Hardware-backed KeyStore	в) Аппаратное обеспечение криптографических операций и безопасное хранение ключей.

Правильный ответ:

1 – а, 2 – б, 3 – в

П7. Установите правильную последовательность этапов проектирования защиты для нового мобильного приложения:

- а) Классификация данных (определение, какие данные являются конфиденциальными).
- б) Анализ угроз и рисков для идентифицированных данных.
- в) Выбор и проектирование соответствующих механизмов защиты (шифрование, аутентификация).
- г) Определение требований к безопасности (регуляторные, бизнес-требования).
- д) Реализация и тестирование выбранных механизмов.

Ответ: ГАБВД

П8. Установите правильную последовательность при проектировании безопасного сетевого взаимодействия:

- а) Выбор стойкого алгоритма шифрования и корректной его реализации.
- б) Реализация механизма проверки сертификатов сервера (Certificate Pinning).
- в) Анализ данных, передаваемых по сети, на предмет конфиденциальности.
- г) Определение необходимости использования VPN или дополнительных каналов связи.
- д) Настройка использования только защищенных протоколов (TLS 1.2+).

Ответ: ВАДБГ

П9. Как называется принцип проектирования безопасности, при котором у приложения есть доступ только к тем ресурсам и данным, которые необходимы для его функционирования?

Правильный ответ: Принцип минимальных привилегий (Principle of Least Privilege)

П10. Как называется аппаратный модуль в современных мобильных устройствах, предназначенный для безопасного хранения криптографических ключей и выполнения операций с ними?

Правильный ответ: Защищенная оболочка (Secure Enclave) или Trusted Execution Environment (TEE)

ПК-3.2.2 Умеет разрабатывать предложения по совершенствованию существующих методов и средств, применяемых для контроля и защиты информации

и повышения эффективности этой защиты

П11. При анализе существующего мобильного приложения обнаружено, что пароли хранятся в SharedPreferences в виде plain text. Какое основное предложение по совершенствованию защиты вы разработаете?

- а) Реализовать вход через социальные сети для удобства;
- б) Внедрить хэширование паролей с использованием "соли" (salt) и надежного алгоритма (bcrypt, PBKDF2); [✓]
- в) Скрыть ввод пароля звездочками в интерфейсе;
- г) Увеличить минимальную длину пароля.

Ответ: б) Внедрить хэширование паролей

П12. Существующее приложение использует для сетевых запросов HTTP. Какое усовершенствование будет наиболее эффективным для защиты передаваемой информации?

- а) Сжимать данные перед отправкой для уменьшения размера пакетов;
- б) Реализовать собственную систему шифрования поверх HTTP;
- в) Перевести все сетевые запросы на использование HTTPS с надежной версией TLS; [✓]
- г) Добавить проверку целостности данных с помощью MD5-хэша.

Ответ: в) Перевести запросы на HTTPS

П13. Какие предложения по совершенствованию будут эффективны для защиты от анализа и реверс-инжиниринга скомпилированного APK-файла? (Выберите несколько вариантов)

- а) Обфускация кода с помощью ProGuard/R8; [✓]
- б) Использование детектора root-прав на устройстве;
- в) Шифрование строковых констант в коде; [✓]
- г) Реализация проверки целостности приложения (Anti-tampering); [✓]
- д) Увеличение количества комментариев в исходном коде.

Обоснование: Обфускация, шифрование строк и проверка целостности напрямую усложняют анализ APK.

Правильные ответы: А, В, Г

П14. Какие действия по совершенствованию защиты данных в локальной БД SQLite вы предложите? (Выберите несколько вариантов)

- а) Использование библиотеки SQLCipher для полного шифрования базы данных; [✓]
- б) Регулярное создание бэкапов БД в облако;
- в) Хранение ключа шифрования БД в виде константы в коде приложения;
- г) Обеспечение безопасного хранения ключа шифрования БД в Android KeyStore; [✓]
- д) Удаление чувствительных данных из БД после каждого сеанса работы.

Обоснование: Полное шифрование БД и безопасное хранение ключа – основные меры. Хранение ключа в коде – уязвимость.

Правильные ответы: А, Г

П15. Установите соответствие между выявленной уязвимостью и предлагаемым усовершенствованием:

Выявленная уязвимость	Предлагаемое усовершенствование
1. Данные в LogCat	а) Внедрить биометрическую аутентификацию или многофакторную аутентификацию (MFA).
2. Слабый парольный доступ	б) Реализовать очистку Sensitive Data из памяти после использования (затирание массивов).
3. Остатки данных в оперативной памяти	в) Запретить логирование конфиденциальных данных, использовать обфускацию для продакшн-сборок.

Правильный ответ:

1 – в, 2 – а, 3 – б

П16. Установите соответствие между недостатком текущего метода и предлагаемым улучшением для контроля доступа:

Недостаток текущего метода	Предлагаемое улучшение
1. Статический PIN-код	а) Внедрение сессий с JWT-токенами, имеющими ограниченное время жизни.
2. "Вечная" сессия после логина	б) Реализация контекстной аутентификации (учитывающей местоположение, устройство).
3. Отсутствие адаптивности	в) Смена на одноразовые пароли (ОТР) или динамические коды.

Правильный ответ:

1 – в, 2 – а, 3 – б

П17. Установите правильную последовательность действий при разработке предложений по совершенствованию защиты:

- а) Проведение аудита безопасности и анализ текущего состояния защиты.
- б) Ранжирование предложений по критериям эффективности, стоимости и сложности внедрения.
- в) Выявление конкретных уязвимостей и недостатков.
- г) Формирование перечня конкретных предложений по устранению каждого недостатка.
- д) Составление дорожной карты (roadmap) по внедрению улучшений.

Ответ: АБГД

П18. Установите правильную последовательность улучшений системы аутентификации:

- а) Добавление двухфакторной аутентификации (2FA) по SMS или через приложение-аутентификатор.
- б) Замена хранения паролей в открытом виде на хэширование с "солью".
- в) Внедрение биометрической аутентификации (Face ID, Touch ID) как удобной альтернативы.
- г) Анализ инцидентов и отзывов пользователей о сложности входа.
- д) Реализация механизма восстановления доступа без ущерба безопасности.

Ответ: ГБАВД

П19. Как называется метод повышения безопасности аутентификации, при котором для доступа требуется предоставить два или более доказательства (фактора) из разных категорий (знание, владение, биометрия)?

Правильный ответ: Многофакторная аутентификация (Multi-Factor Authentication, MFA)

П20. Как называется процесс преобразования читаемого кода в трудночитаемый с целью затруднения его анализа, который является стандартным методом улучшения защиты Android-приложения?

Правильный ответ: Обфускация кода (Code Obfuscation)

ПК-3.2.3 Владеет методами и средствами разработки архитектуры и интерфейсов систем защиты информации, процедуры восстановления работоспособности средств и систем защиты после сбоев

П21. Какой метод разработки архитектуры защиты предполагает создание отдельных, независимо защищенных слоев?

- а) Defense in Depth (Глубокая эшелонированная защита); [✓]
- б) Security through Obscurity (Безопасность через неясность);
- в) Zero Trust (Концепция нулевого доверия);
- г) Single Point of Protection (Единая точка защиты).

Ответ: а) Defense in Depth

Р22. Какое средство разработки интерфейсов Android является стандартным для безопасного ввода паролей?

- а) EditText с настройкой android:inputType="textPassword"; [✓]

- б) TextView с кастомным шрифтом;
- в) Button с обработчиком нажатия;
- г) WebView с загрузкой страницы ввода.

Ответ: а) EditText с настройкой android:inputType="textPassword"

П23. Какие методы и средства следует использовать для разработки безопасного интерфейса ввода платежных данных? (Выберите несколько вариантов)

- а) Использование EditText с android:inputType="numberPassword" для CVV; [✓]
- б) Полный отказ от маскирования вводимых данных для удобства пользователя;
- в) Интеграция готовых, сертифицированных SDK для работы с платежными данными (например, Google Pay); [✓]
- г) Самостоятельная реализация логики хранения номера карты в SharedPreferences для быстрого доступа;
- д) Валидация вводимых данных на стороне клиента для предотвращения очевидных ошибок. [✓]

Обоснование: Специализированные SDK и правильные элементы ввода повышают безопасность. Хранение данных карты самостоятельно – нарушение стандартов.

Правильные ответы: А, В, Д

П24. Какие процедуры восстановления работоспособности должны быть предусмотрены в архитектуре на случай сбоя криптографической подсистемы? (Выберите несколько вариантов)

- а) Автоматический откат на резервный алгоритм шифрования;
- б) Четкое логирование ошибок для последующего анализа; [✓]
- в) Уведомление пользователя о невозможности выполнить операцию с рекомендациями; [✓]
- г) Полное блокирование приложения до обновления;
- д) Бесшовное восстановление сессии и данных без участия пользователя.

Обоснование: Логирование и информирование пользователя – ключевые процедуры восстановления после сбоев в безопасности.

Правильные ответы: Б, В

П25. Установите соответствие между архитектурным компонентом системы защиты и его назначением:

Компонент архитектуры	Назначение
1. Менеджер аутентификации (Auth Manager)	а) Централизованное управление всеми криптографическими операциями и ключами.
2. Менеджер сессий (Session Manager)	б) Ответственен за проверку учетных данных пользователя и выдачу токенов.
3. Криптографический провайдер (Crypto Provider)	в) Контроль времени жизни сессии, ее обновление и корректное завершение.

Правильный ответ:

1 – б, 2 – в, 3 – а

П26. Установите соответствие между типом сбоя и процедурой восстановления:

Тип сбоя	Процедура восстановления
1. Повреждение зашифрованной локальной базы данных	а) Принудительный логアウト пользователя, очистка данных сессии, запрос повторной аутентификации.
2. Скомпрометирован или просрочен токен доступа (Access Token)	б) Попытка восстановления из резервной копии, уведомление пользователя о возможной потере данных.
3. Сбой при проверке целостности приложения (Tamper Detection)	в) Блокировка функционирования приложения, уведомление сервера о возможной атаке.

Правильный ответ:

1 – б, 2 – а, 3 – в

П27. Установите правильную последовательность этапов разработки архитектуры системы защиты мобильного приложения:

- а) Определение границ доверия (Trust Boundary) и точек ввода данных.
- б) Декомпозиция системы на модули и определение ответственности каждого за безопасность.
- в) Выбор конкретных криптографических алгоритмов, протоколов и инструментов.
- г) Моделирование угроз (Threat Modeling) для выявления рисков.
- д) Проектирование схемы взаимодействия модулей и потоков данных.

Ответ: АГБДВ

П28. Установите правильную последовательность действий при восстановлении после сбоя сетевого шифрования:

- а) Попытка повторного рукопожатия (re-handshake) по TLS.
- б) Логирование ошибки с указанием причины и контекста.
- в) Переключение на запасной endpoint (если предусмотрен) с другой конфигурацией.
- г) Оповещение пользователя о проблемах с соединением и рекомендация проверить сеть.
- д) Прекращение попыток отправки конфиденциальных данных до восстановления безопасного канала.

Ответ: БАВГД

П29. Как называется архитектурная концепция безопасности, которая предполагает, что угроза может исходить из любой точки сети, и никаким компонентам по умолчанию не доверяет?

Правильный ответ: Zero Trust Architecture (ZTA) / Архитектура нулевого доверия

П30. Как называется стандартный компонент Android, предназначенный для безопасного хранения криптографических ключей, который следует использовать в архитектуре приложения для работы с шифрованием?

Правильный ответ: Android KeyStore System

Критерии оценки для проведения зачета по дисциплине (тестирование)

Шкала оценивания результатов промежуточной аттестации и критерии выставления оценок.

Оценка	Уровень сформированности компетенции	Критерии
«Отлично» («зачтено»)	Высокий	Выполнено более 80% заданий предложенного теста
«Хорошо» («зачтено»)	Хороший	Выполнено 60-80% заданий предложенного теста
«Удовлетворительно» («зачтено»)	Достаточный	Выполнено 35-60% заданий предложенного теста
«Неудовлетворительно» («не зачтено»)	Недостаточный	Выполнено менее 35% заданий предложенного теста

Обновление оценочных материалов дисциплины

обсуждено и рекомендовано к утверждению решением кафедры
наименование кафедры _____
от ____ ____ 20__ г., протокол № ____

одобрено Научно-методическим советом _____ от ____ ____ 20__ г.,
протокол № ____