**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным работам**

**по РМП**

Студент ИСПП-21 22.06.25 Казнин В.А.

Преподаватель 22.06.25 Садовский Р.В.

Архангельск 2025

**Лабораторная работа №1**

**Установка инструментария и настройка среды для разработки мобильных приложений**

* Цель работы

Изучить процесс установки среды для разработки мобильных приложений.

* **Ответы на контрольные вопросы**
* Какие среды разработки поддерживают создание мобильных приложений?

Ответ: Android Studio, Xcode, Visual Studio, Flutter, React Native, Unity.

* На каких языках программирования можно создавать мобильные приложения?

Ответ: Java, Kotlin, Swift, Objective-C, Dart, JavaScript, C#, Python.

* Какие языки программирования доступны в Android Studio?

Ответ: Java, Kotlin, C++ (через NDK).

* Какой язык программирования является основным для разработки приложений под ОС Android?

Ответ: Kotlin (официально рекомендован Google).

* Что дополнительно нужно установить для работы Android Studio?

Ответ: Android SDK, Java Development Kit (JDK), эмуляторы AVD.

* Что такое Java Development Kit и что входит в его состав?

Ответ: JDK — это комплект для разработки на Java; включает компилятор javac, JVM, стандартные библиотеки, утилиты и документацию.

* **Вывод**

В ходе проделанной лабораторной работы был изучен процесс установки среды для разработки мобильных приложений.

**Лабораторная работа №2**

**Установка среды разработки мобильных приложений с применением виртуальной машины**

* **Цель работы**

Изучить процесс установки среды для разработки мобильных приложений с применением виртуальной машины.

* **Ответы на контрольные вопросы**
* Что такое «виртуальная машина»?

Ответ: программная имитация физического компьютера, которая позволяет запускать приложения в изолированной среде.

* Какие IDE позволяют создавать приложения для Android?

Ответ: Android Studio, IntelliJ IDEA, Visual Studio (с Xamarin), Eclipse.

* Какие языки программирования доступны в Intellij IDEA?

Ответ: Java, Kotlin, Groovy, Scala, Python, JavaScript и другие через плагины.

* Какие языки программирования доступны в Android Studio?

Ответ: Java, Kotlin, C++.

* На каких языках программирования можно разрабатывать нативные приложения для Android?

Ответ: Java, Kotlin, C++.

* **Вывод**

В ходе проделанной лабораторной работы был изучен процесс установки среды для разработки мобильных приложений с применением виртуальной машины.

**Лабораторная работа №3**

**Разработка линейных алгоритмов**

* **Цель работы**

Изучить процесс разработки линейных алгоритмов в приложениях на языке Kotlin.

Изучить процесс ввода и вывода данных в приложениях на языке Kotlin.

* **Ответы на контрольные вопросы**
* Как объявить переменные на Kotlin?

Ответ: с помощью ключевых слов var (изменяемая) и val (неизменяемая), например: var age = 25.

* Как выполнить ввод данных на Kotlin?

Ответ: через readLine(), обычно с последующим преобразованием типа, например: val input = readLine()?.toInt().

* Как выполнить вывод данных на Kotlin?

Ответ: с помощью println() или print().

* Как преобразовать значение из строкового в числовой?

Ответ: с помощью методов toInt(), toDouble() и т.д., например: "42".toInt().

* Как округлить данные на Kotlin?

Ответ: с использованием функций kotlin.math.round(), ceil() или floor().

* Как сгенерировать случайное число на Kotlin?

Ответ: через kotlin.random.Random, например: Random.nextInt(0, 100).

* В чем отличие между ключевыми словами var и val в Kotlin?

Ответ: var — переменная, значение можно изменить; val — константа, значение задаётся один раз.

* **Вывод**

В ходе проделанной лабораторной работы был изучен процесс разработки линейных алгоритмов в приложениях на языке Kotlin.

В ходе проделанной лабораторной работы был изучен процесс ввода и вывода данных в приложениях на языке Kotlin.

**Лабораторная работа №4**

**Разработка разветвляющихся алгоритмов**

* **Цель работы**

Изучить процесс разработки разветвляющихся алгоритмов в приложениях на Kotlin.

* **Ответы на контрольные вопросы**
* Каков синтаксис условного оператора на Kotlin?

Ответ: используется if и else, например: if (x > 0) println("Положительное") else println("Отрицательное").

* Каков синтаксис оператора множественного выбора на Kotlin?

Ответ: используется when, например: when (x) { 1 -> println("Один") 2 -> println("Два") else -> println("Другое") }.

* Каков синтаксис тернарного оператора на Kotlin?

Ответ: в Kotlin нет тернарного оператора, его заменяет if как выражение, например: val max = if (a > b) a else b.

* **Вывод**

В ходе проделанной лабораторной работы был изучен процесс разработки разветвляющихся алгоритмов в приложениях на Kotlin.

**Лабораторная работа №5**

**Разработка циклических алгоритмов**

* **Цель работы**

Изучить процесс разработки циклических алгоритмов в приложениях на Kotlin.

* **Ответы на контрольные вопросы**
* Каков синтаксис оператора for на Kotlin?

Ответ: for (i in 1..10) { println(i) } — перебор диапазона или коллекции.

* Каков синтаксис оператора while на Kotlin?

Ответ: while (x < 10) { println(x); x++ }.

* Каков синтаксис оператора do-while на Kotlin?

Ответ: do { println(x); x++ } while (x < 10).

* Какие операторы досрочного выхода из цикла применяются в Kotlin?

Ответ: break — завершает цикл; continue — переходит к следующей итерации.

* **Вывод**

В ходе проделанной лабораторной работы изучен процесс разработки циклических алгоритмов в приложениях на Kotlin.

**Лабораторная работа №6**

**Разработка и вызов функций**

* **Цель работы**

Изучить процесс разработки функций в приложениях на Kotlin.

* **Ответы на контрольные вопросы**
* Каков синтаксис создания функций на Kotlin?

Ответ: fun имяФункции(параметры): ТипВозвращаемогоЗначения { тело функции }.

* Как указать для параметров методов значения по умолчанию на Kotlin?

Ответ: через присвоение в объявлении, например: fun greet(name: String = "Гость").

* Как указать для параметров переменное количество аргументов на Kotlin?

Ответ: с помощью vararg, например: fun sum(vararg numbers: Int).

* Что такое функции высокого порядка в Kotlin?

Ответ: функции, которые принимают другие функции в качестве параметров или возвращают их.

* **Вывод**

В ходе проделанной лабораторной работы изучен процесс разработки функций в приложениях на Kotlin.

**Лабораторная работа №7**

**Разработка классов**

* **Цель работы**

Изучить процесс разработки и применения классов на языке Kotlin.

* **Ответы на контрольные вопросы**
* Как объявить класс на Kotlin?

Ответ: с помощью class ИмяКласса { ... }.

* Как объявить свойства в Kotlin?

Ответ: внутри класса через var или val, например: var name: String.

* Каков синтаксис объявления геттера и сеттера свойства в Kotlin?

Ответ: через get() и set(value), например:

var age: Int = 0

get() = field

set(value) { field = value }

* Как объявить конструктор в Kotlin?

Ответ: через primary-конструктор в объявлении класса, например: class Person(val name: String).

* Как создать объект в Kotlin?

Ответ: с помощью вызова конструктора, например: val p = Person("Alex").

* **Вывод**

В ходе проделанной лабораторной работы изучен процесс разработки и применения классов на языке Kotlin.

**Лабораторная работа №8**

**Наследование классов**

* **Цель работы**

Изучить процесс разработки дочерних классов в приложениях на Kotlin.

* **Ответы на контрольные вопросы**
* Что такое «наследование»?

Ответ: механизм, позволяющий одному классу получать свойства и методы другого.

* Сколько родительских классов может быть у класса в Kotlin?

Ответ: один (множественное наследование классов запрещено).

* Сколько интерфейсов может реализовывать класс в Kotlin?

Ответ: сколько угодно.

* Как указать родительский класс на Kotlin?

Ответ: через двоеточие, например: class Child : Parent().

* Как переопределить реализацию метода в дочернем классе на Kotlin?

Ответ: с помощью ключевого слова override.

* Как вызвать реализацию родительского метода в дочернем классе на Kotlin?

Ответ: через super.имяМетода().

* Как указать, что класс может быть родительским?

Ответ: объявить класс как open.

* Как указать, что у класса нельзя создавать дочерние классы?

Ответ: класс по умолчанию final, или явно использовать final.

* **Вывод**

• В ходе проделанной лабораторной работы изучен процесс разработки дочерних классов в приложениях на Kotlin.

**Лабораторная работа №9**

**Обработка коллекций**

* **Цель работы**
* Изучить процесс разработки дочерних классов в приложениях на Kotlin.
* **Ответы на контрольные вопросы**
* Какие стандартные типы данных коллекций имеются в Kotlin?

Ответ: List, Set, Map.

* Как объявить коллекцию в Kotlin?

Ответ: через функции listOf(), mutableListOf(), setOf(), mapOf() и их изменяемые аналоги.

* Как добавить элемент в список в Kotlin?

Ответ: если список изменяемый — через add(), например: mutableList.add(element).

* Как добавить элемент в словарь в Kotlin?

Ответ: в изменяемый MutableMap через put(key, value) или оператор map[key] = value.

* Как перебрать элементы коллекции в Kotlin?

Ответ: с помощью цикла for (item in collection) { ... }.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс разработки дочерних классов в приложениях на Kotlin.

**Лабораторная работа №10**

**Создание эмуляторов и подключение устройств**

* Цель работы
* Изучить процесс создания эмуляторов и подключения устройств для мобильной разработки.
* **Ответы на контрольные вопросы**
* Что такое «корутина»?

Ответ: лёгкий поток выполнения, позволяющий писать асинхронный код последовательно.

* Для чего используется launch?

Ответ: для запуска новой корутины без блокировки текущего потока.

* Какие особенности использования у suspend-функций?

Ответ: они могут приостанавливать выполнение без блокировки и вызываются только из корутин или других suspend-функций.

* Как приостановить выполнение корутины?

Ответ: используя ключевое слово suspend и функции, поддерживающие приостановку.

* Как отменить выполнение корутины?

Ответ: через вызов cancel() у Job, связанной с корутиной.

* Как подключить библиотеку для работы с корутинами?

Ответ: добавить зависимость kotlinx-coroutines-core в build.gradle.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс создания эмуляторов и подключения устройств для мобильной разработки.

**Лабораторная работа №11**

**Создание эмуляторов и подключение устройств**

* **Цель работы**
* Изучить процесс создания эмуляторов и подключения устройств для мобильной разработки.
* **Ответы на контрольные вопросы**
* Что такое AVD?

Ответ: Android Virtual Device — виртуальное устройство для запуска Android-приложений на компьютере.

* Что такое эмулятор?

Ответ: программа, имитирующая работу другого устройства или системы.

* Что определяет профиль устройства?

Ответ: конфигурацию эмулятора: модель, размер экрана, характеристики и версии ОС.

* Как создать новый эмулятор?

Ответ: через Android Studio в AVD Manager, выбирая устройство и образ системы.

* Как указать расширенные настройки эмулятора?

Ответ: в AVD Manager через опцию "Show Advanced Settings".

* Какие эмуляторы могут использоваться вместо эмуляторов Android Studio?

Ответ: Genymotion, BlueStacks, Nox Player.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс создания эмуляторов и подключения устройств для мобильной разработки.

**Лабораторная работа №12**

**Настройка режима терминала**

* **Цель работы**
* Изучить процесс использования терминала при разработке мобильных приложений Android.
* **Ответы на контрольные вопросы**
* Что такое adb?

Ответ: Android Debug Bridge — инструмент для взаимодействия с устройствами Android через командную строку.

* Как узнать путь к папке с adb?

Ответ: в Android Studio путь обычно в sdk/platform-tools.

* Как запустить утилиту adb?

Ответ: через командную строку, выполнив adb из папки с утилитой.

* Как узнать список доступных команд adb?

Ответ: выполнить adb help.

* Как подключиться к shell из adb?

Ответ: команда adb shell.

* Как сохранить скриншот и видеозапись экрана из adb?

Ответ: adb shell screencap /sdcard/screen.png и adb shell screenrecord /sdcard/demo.mp4.

* Как выполнить обмен файлами между ПК и устройством на Android из adb?

Ответ: adb push <путь\_на\_ПК> <путь\_на\_устройстве> и adb pull <путь\_на\_устройстве> <путь\_на\_ПК>.

* **Вывод**
* В ходе проделанной лабораторной работы изучен использования терминала при разработке мобильных приложений Android.

**Лабораторная работа №13**

**Создание нового проекта**

* **Цель работы**
* Изучить процесс создания приложения в Android Studio.
* **Ответы на контрольные вопросы**
* Что такое Jetpack Compose?

Ответ: современный фреймворк для декларативного создания UI в Android.

* Какой тип проекта нужно выбрать, чтобы создать проект с использованием Jetpack Compose?

Ответ: проект с поддержкой Compose (Compose Activity) в Android Studio.

* Что такое Activity?

Ответ: экран или компонент пользовательского интерфейса в Android-приложении.

* Для чего используется MainActivity?

Ответ: основная точка входа приложения, первый экран при запуске.

* Для чего используется функция onCreate?

Ответ: для инициализации Activity при его создании.

* Что делает функция setContent?

Ответ: задаёт содержимое UI через Compose.

* Для чего используется функция @Composable?

Ответ: обозначает функцию, которая описывает UI-компонент в Compose.

* Для чего указывается @Preview?

Ответ: для предварительного просмотра UI-компонента в Android Studio без запуска приложения.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс создания приложения в Android Studio.

**Лабораторная работа №14**

**Изучение и комментирование кода**

* **Цель работы**
* Изучить процесс документирования и комментирования приложения Android.
* **Ответы на контрольные вопросы**
* Как оформляются комментарии в силе KDoc?

Ответ: с помощью трёх слэшей /// или блоком /\*\* ... \*/ перед элементом кода.

* Какая общая форма комментария в стиле KDoc?

Ответ:

/\*\*

\* Описание

\* @param имя параметра описание

\* @return описание возвращаемого значения

\*/

* Как в комментарии KDoc описать свойства класса?

Ответ: с помощью тега @property, например: @property name имя пользователя.

* Как в комментарии KDoc описать параметры функции?

Ответ: через тег @param, например: @param age возраст пользователя.

* Как в комментарии KDoc описать возвращаемое значение функции?

Ответ: через тег @return, например: @return результат вычисления.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс документирования и комментирования приложения Android.

**Лабораторная работа №15**

**Изменение элементов дизайна**

* **Цель работы**
* Изучить процесс изменения элементов дизайна приложения в Android Studio.
* **Ответы на контрольные вопросы**
* Какой синтаксис у применения модификаторов?

Ответ: модификаторы в Compose цепляются через точку, например: Modifier.padding(16.dp).background(Color.Red).

* Как указать фон элемента?

Ответ: с помощью Modifier.background(Color).

* Как указать размер элемента?

Ответ: через Modifier.size(width.dp, height.dp) или Modifier.fillMaxWidth().

* Как указать отступы элемента?

Ответ: через Modifier.padding() с нужными значениями.

* Как указать настройки текста элемента?

Ответ: в Text() через параметры: fontSize, fontWeight, color и др.

* Как добавить и использовать изображение из ресурсов?

Ответ: импортировать через painterResource(id = R.drawable.image) и вставить в Image().

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс изменения элементов дизайна приложения в Android Studio.

**Лабораторная работа №16**

**Использование ресурсов**

* **Цель работы**
* Изучить процесс добавления и применения ресурсов при разработке мобильных приложений Android.
* **Ответы на контрольные вопросы**
* Что хранится в папке res?

Ответ: все ресурсы приложения — изображения, строки, цвета, стили и прочее.

* Что хранится в папке values?

Ответ: файлы с определениями строк, цветов, стилей, размеров и массивов.

* Что хранится в папке drawable?

Ответ: графические ресурсы — картинки и иконки.

* Для чего используется Resource Manager?

Ответ: для удобного просмотра и управления ресурсами в Android Studio.

* Как использовать строковые ресурсы?

Ответ: через getString(R.string.name) или в Compose stringResource(R.string.name).

* Как использовать ресурсы изображения?

Ответ: через painterResource(R.drawable.name) в Compose или setImageResource в XML/Java.

* Как использовать массивы из ресурсов?

Ответ: через resources.getStringArray(R.array.name) или аналогично в коде.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс добавления и применения ресурсов при разработке мобильных приложений Android.

**Лабораторная работа №17**

**Настройка элементов управления для ввода данных**

* **Цель работы**
* Изучить процесс настройки и использования элементов управления для ввода данных приложения Android.
* **Ответы на контрольные вопросы**
* Какие элементы позволяют ввести текстовые значения?

Ответ: TextField и OutlinedTextField в Jetpack Compose.

* Как настроить тип клавиатуры?

Ответ: через параметр keyboardOptions с указанием типа, например KeyboardType.Number.

* Как связать значение из элемента управления с переменной?

Ответ: через состояние (State), например var text by remember { mutableStateOf("") } и передача в value и onValueChange.

* Для чего используется combobox, какие его основные настройки?

Ответ: для выбора одного значения из списка; настраивается список элементов и выбранный элемент.

* Для чего используется radiobutton, какие его основные настройки?

Ответ: для выбора одного варианта из группы; задаётся состояние выбора и обработчик изменений.

* Для чего используется slider, какие его основные настройки?

Ответ: для выбора числового значения в диапазоне; настраиваются value, onValueChange, valueRange.

* Для чего используется switch, какие его основные настройки?

Ответ: переключатель включения/выключения; настраиваются checked и onCheckedChange.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс настройки и использования элементов управления для ввода данных приложения Android.

**Лабораторная работа №18**

**Настройка кнопок**

* **Цель работы**
* Изучить процесс настройки и использования элементов управления кнопок в приложениях Android.
* **Ответы на контрольные вопросы**
* Как объявить button?

Ответ: через Button(onClick = { ... }) { Text("Текст") }.

* Что такое «IconButton»?

Ответ: кнопка с иконкой вместо текста, например: IconButton(onClick = { ... }) { Icon(...) }.

* Что такое «IconToggleButton»?

Ответ: кнопка-переключатель с иконкой, отображающая активное/неактивное состояние.

* Что такое «FloatingActionButton»?

Ответ: плавающая круглая кнопка, обычно для основного действия на экране.

* Что такое «ExtendedFloatingActionButton»?

Ответ: FAB с текстом и иконкой, больше по размеру и описательнее.

* Какие параметры нужно указать при создании объекта Icon?

Ответ: imageVector или painter, contentDescription, опционально — tint и modifier.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс настройки и использования элементов управления кнопок в приложениях Android.

**Лабораторная работа №19**

**Отображение списков и таблиц**

* **Цель работы**
* Изучить процесс настройки и использования элементов управления для отображения списков в приложении Android.
* **Ответы на контрольные вопросы**
* Что такое «LazyColumn»?

Ответ: вертикальный список с ленивой подгрузкой элементов (рендерит только видимые).

* Что такое «LazyRow»?

Ответ: горизонтальный список с ленивой подгрузкой элементов.

* Что такое «LazyVerticalGrid»?

Ответ: ленивый вертикальный грид с множеством колонок для отображения элементов в сетке.

* Что такое «LazyHorizontalGrid»?

Ответ: ленивый горизонтальный грид с множеством строк, используется для прокрутки по горизонтали в сетке.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс настройки и использования элементов управления для отображения списков в приложении Android.

**Лабораторная работа №20**

**Обработка событий: переключение между экранами**

* **Цель работы**
* Изучить процесс настройки и реализации перехода между экранами в приложении Android.
* **Ответы на контрольные вопросы**
* Какую зависимость нужно добавить в Gradle для поддержки навигации?

Ответ: implementation "androidx.navigation:navigation-compose:<версия>".

* Для чего предназначен NavHost?

Ответ: для отображения экранов в зависимости от текущего маршрута навигации.

* Для чего предназначен NavController?

Ответ: для управления навигацией — переходами между экранами и возвратами.

* Что указывается в графе навигации?

Ответ: маршруты (routes), связанные с конкретными экранами (Composable-функциями).

* В каких случаях для организации маршрутов используется sealed class?

Ответ: когда нужно передавать аргументы и описывать маршруты как объекты с логикой.

* В каких случаях для организации маршрутов используется enum?

Ответ: при простом и фиксированном наборе экранов без параметров.

* Как считать параметры, переданные при навигации?

Ответ: через navBackStackEntry.arguments?.getString("ключ") внутри NavHost.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс настройки и реализации перехода между экранами в приложении Android.

**Лабораторная работа №21**

**Обработка событий: подсказки**

* **Цель работы**
* Изучить процесс настройки и использования элементов управления для отображения уведомлений и подсказок.
* **Ответы на контрольные вопросы**
* Какой класс позволяет описать диалоговое окно?

Ответ: AlertDialog.

* Какая общая форма описания диалогового окна?

Ответ: AlertDialog(onDismissRequest = { ... }, title = { Text("Заголовок") }, text = { Text("Сообщение") }, confirmButton = { ... }).

* Какой класс позволяет описать настройки всплывающего сообщения?

Ответ: Snackbar.

* Для чего в диалоговых окнах используется функция onDismissRequest?

Ответ: для обработки закрытия диалога, например, при нажатии вне окна.

* Для чего применяется объект SnackbarHost?

Ответ: для отображения очереди Snackbar-сообщений в приложении.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс настройки и использования элементов управления для отображения уведомлений и подсказок.

**Лабораторная работа №22**

**Обработка событий: индикация**

* **Цель работы**
* Изучить процесс изменения интерфейса для информирования пользователей в приложениях Android.
* **Ответы на контрольные вопросы**
* Какие классы индикаторов прогресса используются в приложениях Android?

Ответ: LinearProgressIndicator, CircularProgressIndicator.

* Как указать шаг в индикаторах прогресса?

Ответ: для кастомного шага используют значение progress, например: progress = 0.5f.

* Как указать задержку в индикаторах прогресса?

Ответ: через корутину и delay() для имитации загрузки или анимации.

* Как настроить цвет в индикаторах прогресса?

Ответ: с помощью параметра color, например: color = Color.Red.

* Для чего используется BadgedBox?

Ответ: для отображения бейджа (значка) поверх иконки, например, счётчика уведомлений.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс изменения интерфейса для информирования пользователей в приложениях Android.

**Лабораторная работа №23**

**Настройка тем**

* **Цель работы**
* Изучить процесс изменения интерфейса для информирования пользователей в приложениях Android.
* **Ответы на контрольные вопросы**
* Что такое MaterialTheme?

Ответ: контейнер в Compose, задающий стили для цветов, типографики и форм элементов интерфейса.

* Как настроить палитру?

Ответ: через параметры colorScheme в MaterialTheme, передавая свою lightColorScheme() или darkColorScheme().

* Как настроить типографику?

Ответ: через параметр typography в MaterialTheme, передавая собственный набор стилей текста.

* Как сменить тему на светлую/темную?

Ответ: использовать условие isSystemInDarkTheme() и передавать нужную палитру в MaterialTheme.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс изменения интерфейса для информирования пользователей в приложениях Android.

**Лабораторная работа №24**

**Настройка анимации**

* **Цель работы**
* Изучить процесс настройки анимации интерфейса приложений Android.
* **Ответы на контрольные вопросы**
* Для чего используется анимация в приложениях?

Ответ: для улучшения пользовательского опыта, акцентов, обратной связи и плавных переходов между состояниями.

* На какие виды делится анимация в Jetpack Compose?

Ответ: одноразовая (Animate\*AsState), переходная (UpdateTransition), бесконечная (InfiniteTransition), а также анимация видимости (AnimatedVisibility).

* Для чего используется анимация Animate\*AsState?

Ответ: для плавного перехода одного значения к другому при изменении состояния.

* Для чего используется анимация UpdateTransition?

Ответ: для анимации нескольких параметров одновременно при переключении между состояниями.

* Для чего используется анимация InfiniteTransition?

Ответ: для создания бесконечных повторяющихся анимаций, например мигания или пульсации.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс настройки анимации интерфейса приложений Android.

**Лабораторная работа №25**

**Работа с файловой системой**

* **Цель работы**
* Изучить процесс работы с файлами в приложениях Android.
* **Ответы на контрольные вопросы**
* Как добавить папку assets в проект?

Ответ: создать вручную папку assets в src/main и поместить туда нужные файлы.

* Как открыть файл из assets?

Ответ: использовать context.assets.open("имя\_файла").

* Как получить путь к папке приложения?

Ответ: через context.filesDir.path или context.getExternalFilesDir(null)?.path.

* Как считать данные из файла?

Ответ: через inputStream.readBytes() или inputStream.bufferedReader().readText().

* Как записать данные в файл?

Ответ: использовать context.openFileOutput("имя", MODE\_PRIVATE).write(...).

* Как вызвать диалог выбора файла?

Ответ: через Intent.ACTION\_OPEN\_DOCUMENT с startActivityForResult() или ActivityResultLauncher.

* Как вызвать диалог сохранения файла?

Ответ: использовать Intent.ACTION\_CREATE\_DOCUMENT и настроить MIME-тип и имя файла.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс работы с файлами в приложениях Android.

**Лабораторная работа №26**

**Передача данных между модулями**

* **Цель работы**
* Изучить процесс создания стандартных модулей в приложении Android.
* **Ответы на контрольные вопросы**
* Для чего используется модульный подход?

Ответ: для разделения проекта на логические части, упрощения поддержки, тестирования и повторного использования кода.

* Что такое data class и какие у него особенности?

Ответ: это класс для хранения данных; автоматически генерирует equals(), hashCode(), toString() и copy().

* Что такое MVVM?

Ответ: архитектурный шаблон (Model-View-ViewModel), разделяющий бизнес-логику, представление и логику отображения для повышения читаемости и тестируемости кода.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс создания стандартных модулей в приложении Android.

**Лабораторная работа №27**

**Передача данных между модулями**

* **Цель работы**
* Изучить процесс передачи данных между модулями в приложении Android.

* **Ответы на контрольные вопросы**
* Как передать параметры между модулями?

Ответ: через интерфейсы, зависимости или ViewModel, либо с использованием навигационных аргументов.

* Как передать значение простого типа с использованием навигации?

Ответ: добавить в маршрут /{param} и передать через navController.navigate("route/value").

* Как передать объект с использованием навигации?

Ответ: сериализовать объект в JSON (например, с помощью Gson) и передать как строку, затем десериализовать на целевом экране.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс передачи данных между модулями в приложении Android.

**Лабораторная работа №28**

**Работа с БД**

* **Цель работы**
* Изучить процесс работы с БД в приложении Android.
* **Ответы на контрольные вопросы**
* Как создать БД SQLite в приложении Android?

Ответ: через SQLiteOpenHelper, реализовав методы onCreate() и onUpgrade().

* Как получить список строк таблицы БД SQLite в приложении Android?

Ответ: используя query() или rawQuery() с последующим обходом Cursor.

* Как добавить новую строку в таблицу БД SQLite в приложении Android?

Ответ: через insert() с передачей ContentValues.

* Как изменить существующую строку в таблице БД SQLite в приложении Android?

Ответ: через update() с условием WHERE.

* Как удалить строку из таблицы БД SQLite в приложении Android?

Ответ: используя delete() с указанием условия удаления.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс работы с БД в приложении Android.

**Лабораторная работа №29**

**Работа с API**

* **Цель работы**
* Изучить процесс работы с API в приложениях Android.

* **Ответы на контрольные вопросы**
* Что такое REST?

Ответ: архитектурный стиль взаимодействия с веб-сервисами через HTTP с использованием стандартных методов.

* Какие методы являются стандартными для REST?

Ответ: GET, POST, PUT, DELETE, PATCH.

* Какой формат может быть у данных, возвращаемых при использовании REST?

Ответ: чаще всего JSON или XML.

* Для чего используется библиотека retrofit2?

Ответ: для удобного создания HTTP-клиентов и работы с REST API в Android.

* Как в манифесте Android разрешить использование интернет?

Ответ: добавить <uses-permission android:name="android.permission.INTERNET" /> в файл AndroidManifest.xml.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс работы с API в приложениях Android.

**Лабораторная работа №30**

**Тестирование и оптимизация мобильного приложения**

* **Цель работы**
* Изучить процесс тестирования и оптимизации приложений Android.

* **Ответы на контрольные вопросы**
* Что такое модульное тестирование?

Ответ: тестирование отдельных компонентов или функций программы отдельно от всей системы.

* В какой папке проекта хранятся модульные тесты?

Ответ: в папке src/test/java.

* Как добавить точку останова в приложение?

Ответ: кликнуть слева от строки кода в IDE (например, в Android Studio).

* Как запустить приложение в режиме отладки?

Ответ: выбрать Debug вместо Run и запустить через IDE с подключённым отладчиком.

* **Вывод**
* В ходе проделанной лабораторной работы изучен процесс тестирования и оптимизации приложений Android.