

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Программирование мобильных приложений

ОТЧЁТ
по лабораторной работе №1
«ЛАБОРАТОРНАЯ РАБОТА 1»

Выполнил:

студент гр. 253501
Щур М. А.

Проверил:

старший преподаватель
кафедры информатики
Владымцев В.Д.

Минск 2025

1 ЦЕЛЬ РАБОТЫ

Цель лабораторной работы заключается в освоении процесса разработки Android-приложений с использованием Kotlin Multiplatform. В рамках проекта требуется разработать приложение «Unit/Currency Converter», способное выполнять конвертацию валют и различных единиц измерения. Приложение должно поддерживать несколько ориентаций экрана и обладать интуитивно понятным пользовательским интерфейсом.

Приложение должно содержать три категории единиц измерения (например, длина, масса, валюта), каждая из которых включает не менее трех единиц. Для ввода данных необходимо создать цифровую клавиатуру в стиле калькулятора, а также предусмотреть два поля для отображения исходных и конвертированных значений. Пользователь не должен иметь возможности редактировать эти поля с помощью стандартной системной клавиатуры.

Рекомендуется разделить интерфейс на два фрагмента: для данных и для клавиатуры. В портретной ориентации фрагменты должны располагаться вертикально, а в ландшафтной – горизонтально. Кроме того, необходимо реализовать премиальную версию приложения, которая будет включать кнопку для обмена исходными и конвертированными значениями (и единицами). Рядом с каждым полем следует добавить кнопки для копирования данных в буфер обмена. Для различных вариантов сборки необходимо использовать source sets, чтобы определить отдельные макеты и файлы кода.

2 ВЫПОЛНЕНИЕ ЗАДАЧИ

Для выполнения лабораторной работы по разработке приложения «Конвертер единиц и валют» необходимо создать функциональность, которая будет поддерживать преобразование различных категорий единиц измерения, таких как расстояние, масса и валюта. Каждая категория должна включать как минимум три единицы измерения. В пользовательском интерфейсе приложения будут предусмотрены два текстовых поля для отображения исходных значений и результатов преобразования. Ввод данных будет осуществляться через специально разработанную цифровую клавиатуру, которая исключает возможность редактирования полей с использованием стандартной системной клавиатуры.

Приложение будет состоять из двух фрагментов: один фрагмент будет отвечать за ввод и настройку данных, а другой за их отображение. В зависимости от ориентации экрана фрагменты будут располагаться вертикально в портретном режиме и горизонтально в ландшафтном. Для реализации этой функциональности будут использоваться отдельные *XML*-макеты для различных конфигураций экрана.

Для удобства пользователей будет реализована кнопка, которая позволяет переключать местами исходные и конвертированные значения, а также их единицы измерения. Вокруг каждого текстового поля будет расположена кнопка для копирования текущего значения в буфер обмена. В процессе разработки будет создана премиальная версия приложения с использованием *build flavor* и *source sets* для поддержки различных файлов для разных сборок.

Разработка будет осуществляться с применением современных компонентов *Android*, таких как *Activity* и *Fragment*, что позволит эффективно организовать код и интерфейс приложения. Особое внимание будет уделено сохранению состояния приложения при изменении ориентации экрана, чтобы данные оставались корректно отображенными. Также в проекте будут применяться архитектурные подходы для разделения интерфейса, бизнес-логики и данных, что обеспечит устойчивость приложения к изменениям конфигурации устройства.

3 ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

3.1 *Introduction to Activities*

Activity — это ключевой компонент *Android*-приложений, представляющий собой один экран с пользовательским интерфейсом. Это точка взаимодействия пользователя с приложением, и каждое приложение должно содержать хотя бы одну *Activity*, обычно называемую *MainActivity*, которая запускается при старте приложения.

3.2 *Activity Lifecycle*

Жизненный цикл *Activity* описывает, как приложение реагирует на изменения, такие как изменение ориентации экрана или переход в фоновый режим. Основные стадии жизненного цикла:

- 1 *onCreate*: создание *Activity*.
- 2 *onStart*: *Activity* становится видимой.
- 3 *onResume*: *Activity* взаимодействует с пользователем.
- 4 *onPause*: временная потеря фокуса.
- 5 *onStop*: *Activity* становится невидимой.
- 6 *onDestroy*: *Activity* завершается и удаляется из памяти.

3.3 *Fragments*

Fragment представляет собой модульный компонент интерфейса, который работает внутри *Activity*. Один *Activity* может содержать несколько фрагментов, что позволяет гибко управлять интерфейсом, особенно в разных ориентациях экрана (портретной и ландшафтной).

3.4 *Build Variants*

Варианты сборки (Build Variants) — это механизм в *Android*, который позволяет создавать несколько версий приложения (например, бесплатную и премиальную) из одного проекта. Используя *product flavors* и *build types*, можно задавать уникальные ресурсы и макеты для каждой версии, что снижает дублирование кода и облегчает разработку.

3.5 *Product Flavors*

Product Flavors (продуктовые варианты) представляют собой функциональность, которая позволяет создавать различные версии одного и того же приложения из одного исходного кода. Это особенно полезно для создания разных версий приложения для различных рынков, брендов или устройств, сохраняя при этом общую структуру кода.

3.6 Views

Понятие *View* (представление) относится к элементу пользовательского интерфейса, который отображает информацию пользователю и управляет взаимодействием с ним. В Android *view* является основой для всех UI-компонентов, таких как кнопки, текстовые поля, списки и т.д..

3.7 Controls

Термин *Controls* относится к элементам управления пользовательским интерфейсом (UI controls). Это компоненты, которые позволяют пользователю взаимодействовать с приложением. Они могут включать в себя различные элементы, такие как кнопка, текстовые поля, списки, переключатели и так далее.

ЗАКЛЮЧЕНИЕ

В процессе выполнения данной лабораторной работы были изучены и успешно применены основные подходы к разработке Android-приложений с использованием Kotlin Multiplatform. Главной задачей проекта стало создание приложения «Unit/Currency Converter», предназначенного для конвертации различных единиц измерения и валют. Приложение демонстрирует ключевые принципы разработки Android-приложений, включая поддержку различных ориентаций экрана и создание удобного пользовательского интерфейса.

Приложение включает три категории единиц измерения, каждая из которых содержит не менее трех типов единиц. Для ввода данных была разработана специальная цифровая клавиатура, которая блокирует использование стандартной системной клавиатуры для редактирования полей. Интерфейс был разделен на два фрагмента: для данных и для клавиатуры, что позволило гибко настраивать расположение элементов в зависимости от ориентации устройства. В портретной ориентации фрагменты располагаются вертикально, а в ландшафтной — горизонтально.

Дополнительные функции приложения включают кнопку для обмена исходных и конвертированных значений, а также возможность копирования данных в буфер обмена с помощью специальных кнопок. Для реализации премиальной версии приложения использовались механизмы source sets и build flavors, что позволило разделить код и ресурсы для разных вариантов сборки.

Проект был разработан с использованием современных Android-компонентов, таких как Activity и Fragment, с корректным управлением их жизненным циклом, что обеспечило сохранение данных при изменении ориентации экрана. Применение архитектурных принципов разделения логики, интерфейса и данных повысило устойчивость и масштабируемость приложения.

Таким образом, выполнение лабораторной работы позволило освоить основы разработки Android-приложений с использованием Kotlin Multiplatform, создание адаптивных интерфейсов и управление состоянием приложения, что является важной базой для разработки более сложных и функциональных приложений в будущем.