

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 №3  
**«ЛАБОРАТОРНАЯ РАБОТА 3»**

Выполнил:

студент гр. 253501  
Щур М.А.

Проверил:

старший преподаватель  
кафедры информатики  
Владымцев В.Д.

## **1 ЦЕЛЬ РАБОТЫ**

Цель данной лабораторной работы является разработка мобильного приложения-калькулятора, предоставляющее два режима работы: стандартный и научный. Приложение должно обеспечивать возможность переключения между режимами через пользовательский интерфейс, а также автоматически переходить в научный режим при изменении ориентации устройства на ландшафтную. Для реализации разных версий приложения (демо и полной) планируется использовать build flavors. В ходе выполнения работы требуется изучить использование фрагментов, управление ориентацией экрана и настройку build variants.

## 2 ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

В ходе выполнения лабораторной работы было разработано мобильное приложение-калькулятор с двумя режимами: обычным и научным. Для реализации функциональности переключения между режимами был создан пользовательский интерфейс, позволяющий вручную изменять режим работы приложения. Кроме того, реализована автоматическая смена режима на научный при повороте устройства в ландшафтную ориентацию.

Для разделения функциональности между режимами были использованы фрагменты, что позволило структурировать приложение и упростить управление интерфейсом. В процессе работы изучены принципы работы с фрагментами, их жизненный цикл и взаимодействие между ними.

Для реализации двух версий приложения — демо и полной — были настроены build flavors, что дало возможность управлять сборками приложения с разными наборами функций и конфигураций. Также была изучена настройка build variants для упрощения разработки и тестирования различных версий.

Таким образом, в ходе лабораторной работы были достигнуты поставленные цели, а разработанное приложение отвечает заданным требованиям.

### 3 ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

Data Storage относится к способам хранения данных внутри приложения. Для этого могут использоваться локальные базы данных, такие как SQLite, которые предназначены для работы со структурированными данными, или файловые системы для хранения неструктурированных данных, таких как изображения или текстовые файлы. Кроме того, популярным решением является использование библиотек для сериализации данных, например, GSON, позволяющих сохранять и загружать объекты в формате JSON. Например:

```
val sharedPreferences=getSharedPreferences("My», Context.MODE_PRIVATE)
val editor = sharedPreferences.edit()

// Сохранение данных
editor.putString("username", "JohnDoe")
editor.putInt("age", 30)
editor.apply()

// Чтение данных
val username = sharedPreferences.getString("username", null)
val age = sharedPreferences.getInt("age", 0)
```

App Architecture определяет организацию структуры приложения и взаимодействие между его компонентами. Использование продуманной архитектуры, например, MVVM (Model-View-ViewModel), способствует разделению логики приложения на отдельные слои, что упрощает поддержку, тестирование и масштабирование. Архитектура также учитывает управление жизненным циклом компонентов, таких как Activity и Fragment, что помогает избежать утечек памяти и обеспечивает стабильную работу приложения:

- Model: Содержит бизнес-логику и данные приложения.
- View: Отвечает за отображение данных и взаимодействие с пользователем.
- ViewModel: Посредник между моделью и представлением. ViewModel подготавливает данные для отображения и обрабатывает пользовательские действия.

Styling отвечает за визуальное оформление приложения, включая использование цветов, шрифтов, отступов и анимаций. В Android для стилизации применяются стили и темы, которые позволяют централизованно управлять внешним видом приложения. Это помогает добиться единообразного, эстетичного интерфейса, соответствующего бренду и ожиданиям пользователей.

Services — это компоненты Android, предназначенные для выполнения длительных операций в фоновом режиме, таких как воспроизведение музыки или отслеживание местоположения. В приложении-таймере службы могут использоваться для управления таймером и отправки уведомлений, даже если приложение свернуто. Ключевой особенностью является оптимизация работы

служб для снижения расхода батареи и соблюдение ограничений операционной системы, накладываемых на фоновые процессы. Например:

```
class MyService : Service() {  
    override fun onBind(intent: Intent?): IBinder? {  
        return null // В данном случае сервис не привязывается  
    }  
  
    override fun onStartCommand(intent: Intent?, flags: Int, startId:  
Int): Int {  
        // Выполнение фоновой работы  
        return START_STICKY  
    }  
  
    override fun onDestroy() {  
        super.onDestroy()  
        // Очистка ресурсов  
    }  
}
```

Preferences используются для хранения пользовательских настроек, таких как выбор темы, языка или параметров таймера. Библиотека AndroidX Preference предоставляет удобные инструменты для создания и управления экранами настроек. Эти настройки позволяют персонализировать приложение в соответствии с предпочтениями каждого пользователя. Особенностью является возможность синхронизации настроек между устройствами, если приложение поддерживает такую функцию, что делает использование приложения более удобным. Например:

```
Class MainActivity : AppCompatActivity() {  
  
    private lateinit var sharedPreferences: SharedPreferences  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        sharedPreferences = getSharedPreferences("MyPreferences",  
Context.MODE_PRIVATE)  
  
        // Сохранение данных  
        val editor = sharedPreferences.edit()  
        editor.putString("username", "JohnDoe")  
        editor.apply()  
  
        // Чтение данных  
        val username = sharedPreferences.getString("username",  
"default_user")  
        Toast.makeText(this, "Username: $username",  
Toast.LENGTH_SHORT).show()  
    }  
}
```

## **ВЫВОДЫ**

В ходе лабораторной работы были успешно реализованы поставленные задачи по разработке мобильного приложения-калькулятора с двумя режимами: обычным и научным. Был создан удобный пользовательский интерфейс для ручного изменения режима работы приложения. Также внедрена автоматическая смена на научный режим при изменении ориентации устройства на ландшафтную, что улучшает пользовательский опыт. Применение фрагментов позволило разделить функциональность приложения на логические части, обеспечив удобное управление интерфейсом и его структурированность. В процессе работы изучены основные принципы работы с фрагментами, их жизненный цикл и способы взаимодействия между ними. Для реализации двух версий приложения (демо и полной) были настроены build flavors. Это дало возможность создавать разные варианты приложения с различным функционалом, что упростило процесс разработки и тестирования. Также была изучена настройка build variants, что является важным навыком для управления конфигурациями проекта.