**Задание 1 – Создание фрагмента. Жизненный цикл фрагмент**

**Создание фрагментов вручную**

Создадим два фрагмента вручную. Для этого необходимо создать для них layout'ы и классы. Xml код данных фрагментов предоставлены на рисунках 1.1 и 1.2, на рисунках 1.3 и 1.4 предоставлен код реализации данных фрагметов.



Рисунок 1.1 – Файл fragment1.xml



Рисунок 1.2 – Файл fragment2.xml

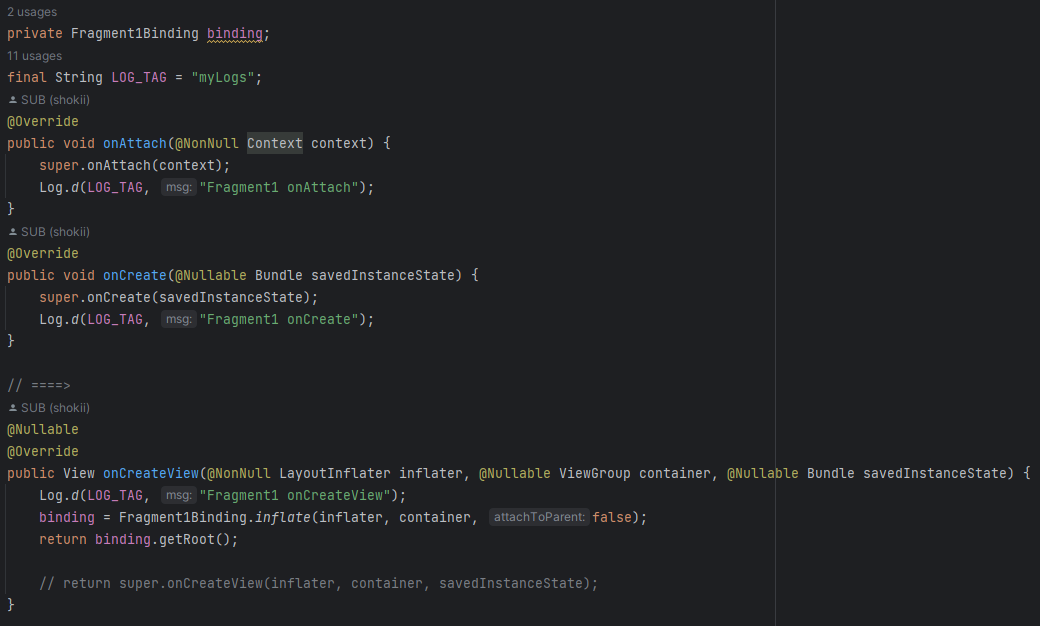


Рисунок 1.3 – Часть кода в файле Fragment1.java

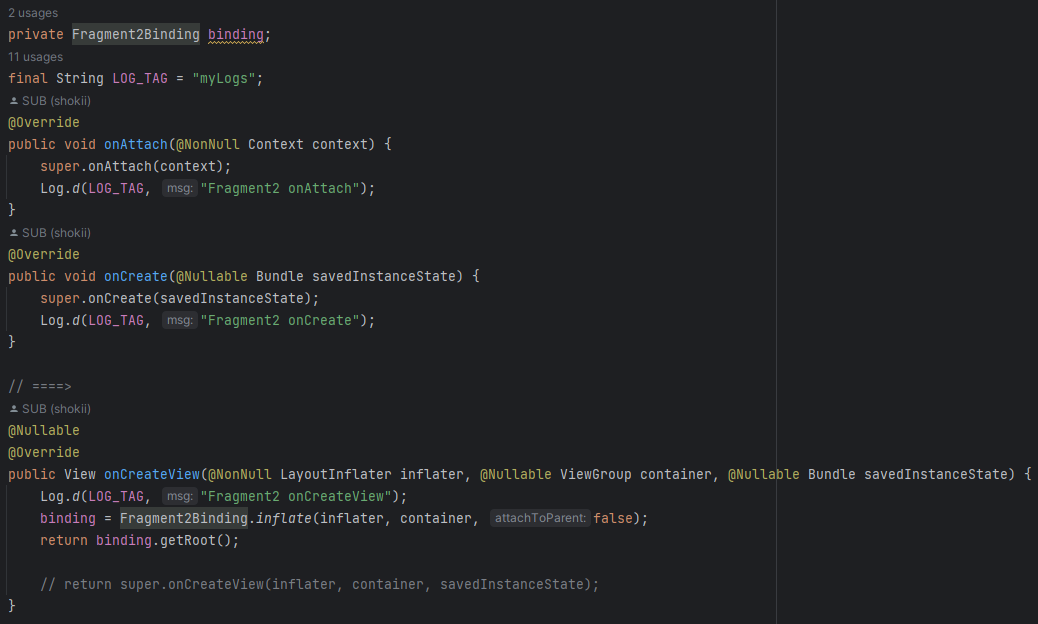


Рисунок 1.4 – Часть кода в файле Fragment2.java

Итак, всё (и layout, и классы) для фрагментов готово. Можем поместить их в основной layout-файл главного Activity. Для этого реализуем код который предоставлен на рисунке 1.5.

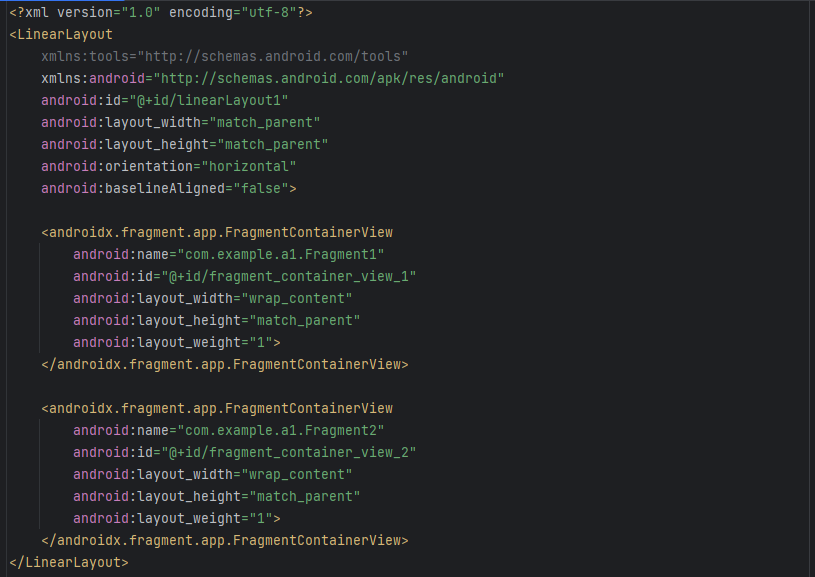


Рисунок 1.5 – Файл main\_activity.xml

Реализуем класс, который будет отвечать за главное activity, а также добавим запись в лог всех методов жизненного цикла Activity. Код реалзации предоставлен на рисунке 1.6.

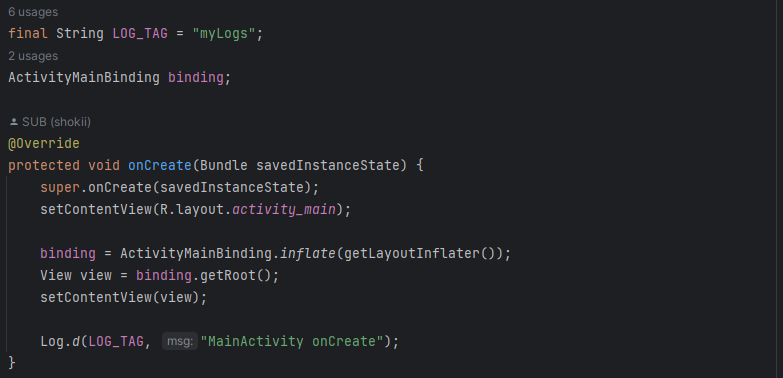


Рисунок 1.6 – Часть кода в файле MainActivity.java

Запустим наше приложение на виртуальном устройстве. Отображение на устройстве предоставлено на рисунке 1.7.

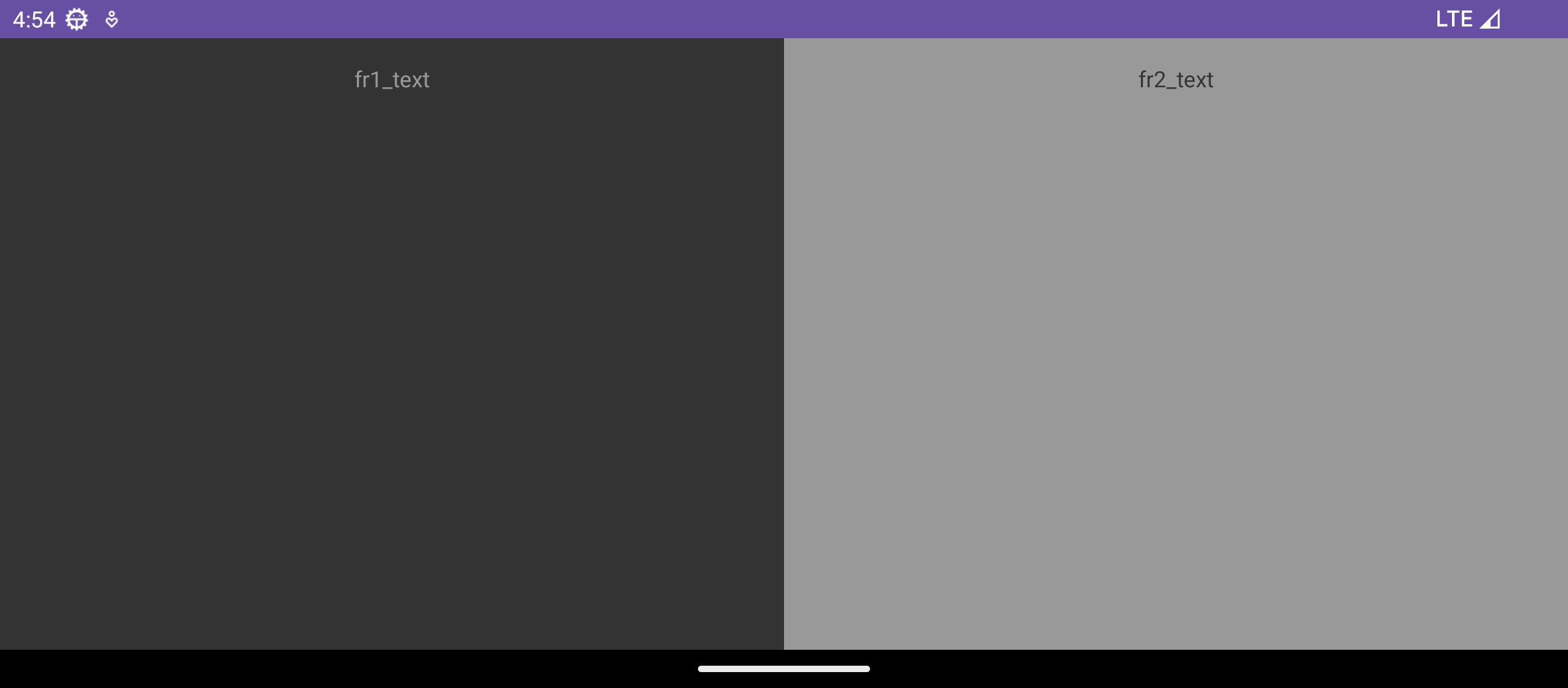


Рисунок 1.7 – Горизонтальное отображение фрагментов

Логи которые вывела наша программа предоставлена на рисунке 1.8.

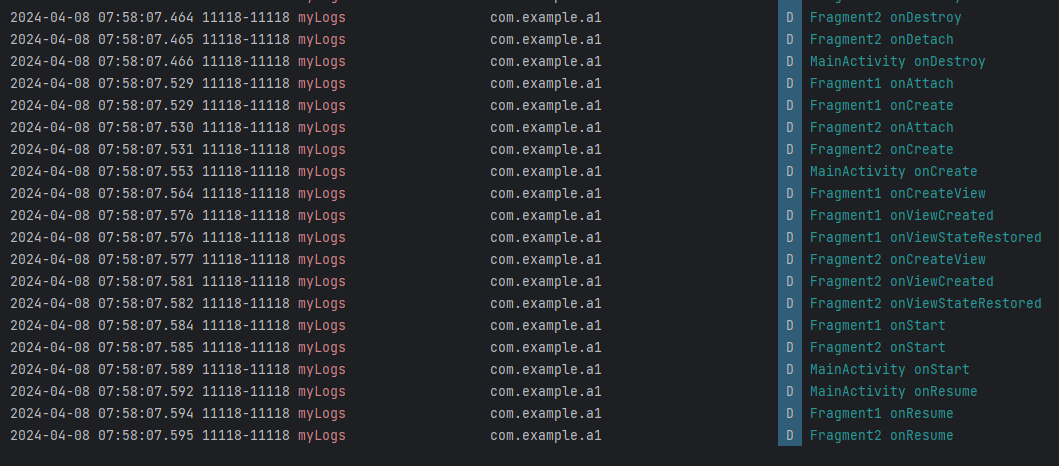


Рисунок 1.8 – Логи приложения

**Создание фрагмента с помощью мастера**

По выбранному шаблону будут добавлены в проект класс, layout и строковый ресурс. Однако, при запуске программы фрагмент не будет отображаться, т.к. он не прописан в layout главного Activity. Добавьте в проект с помощью мастера ещё один фрагмент по шаблону Fragment(Blank) Откройте layout-файл главного Activity и добавьте в него код предоставленный на рисунке 2.1.

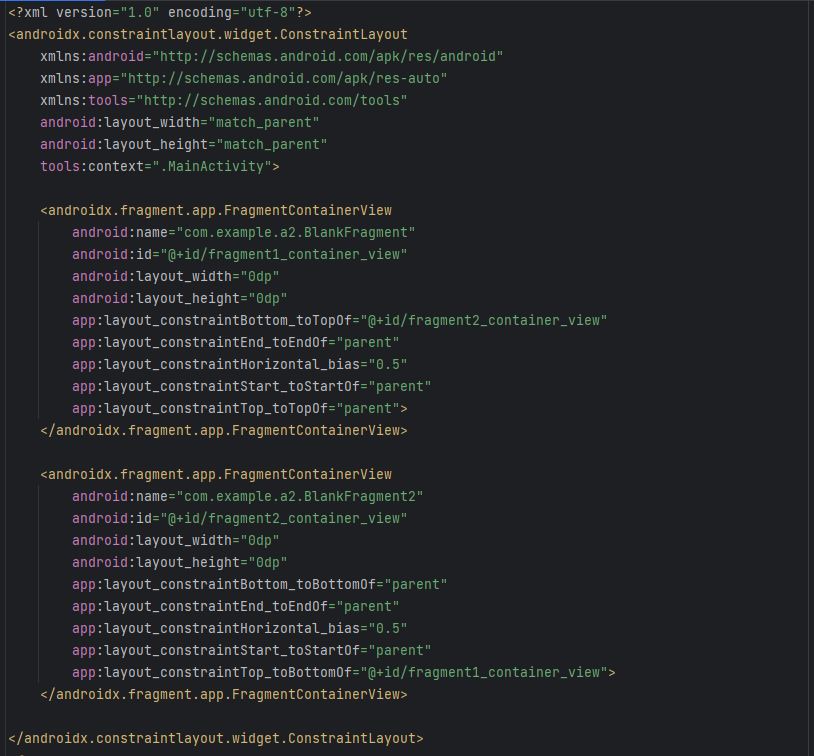


Рисунок 2.1 – Файл main\_activity.xml

Xml код который был автоматически создан мастером предоставлен на рисунке 2.2.

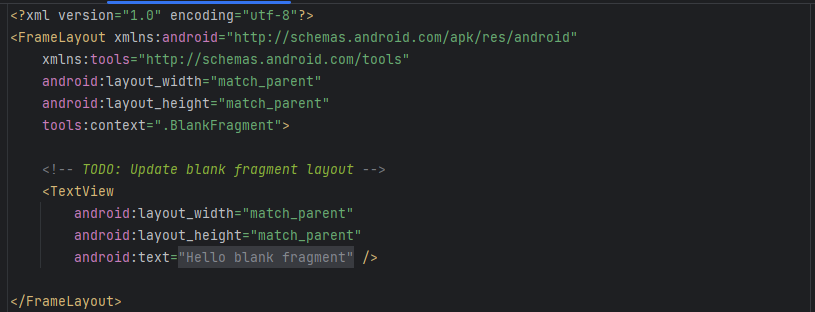


Рисунок 2.2 – Файл fragment\_blank.xml

Запустим наше приложение на виртуальном устройстве. Отображение на устройстве предоставлено на рисунке 2.3.

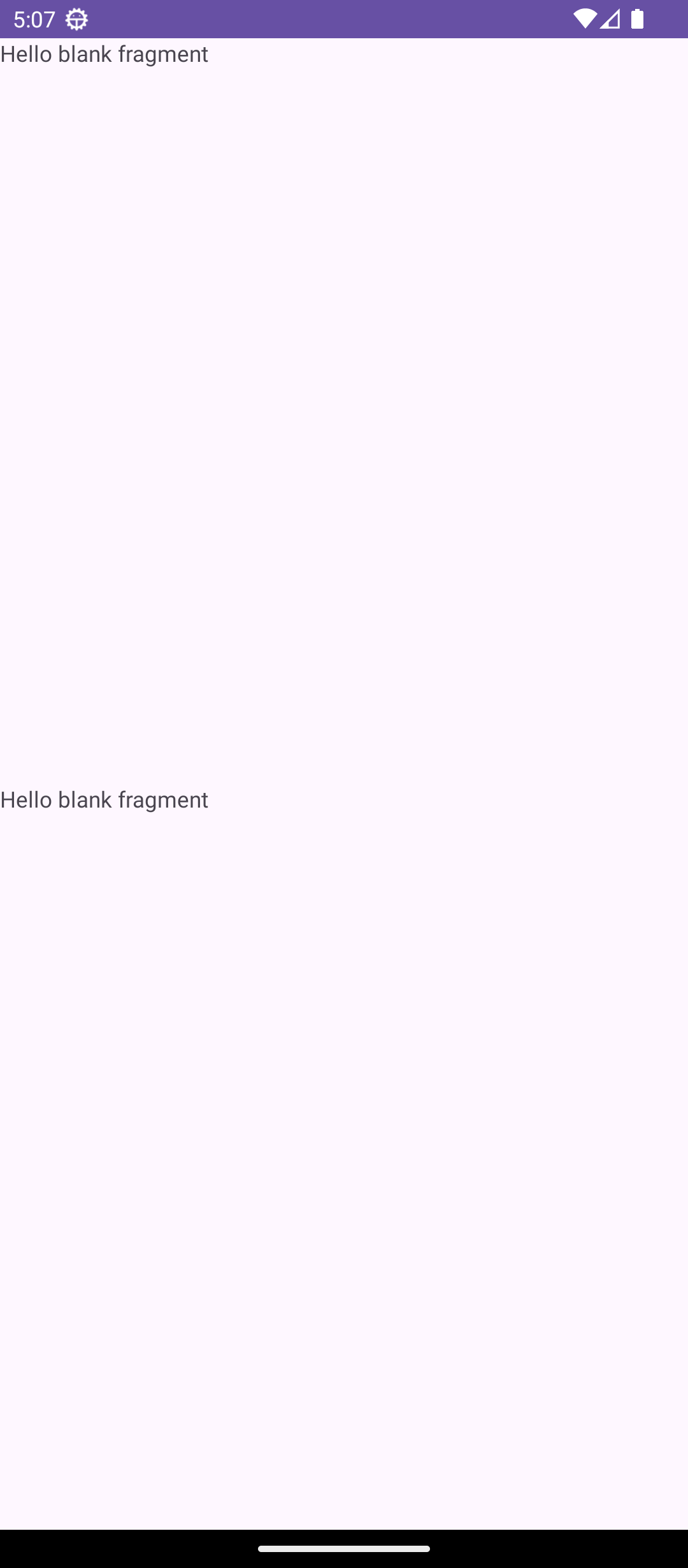


Рисунок 2.3 – Приложение запущенное на устройстве

**Задание 2 – Динамическая работа с фрагментами**

В strings.xml добавим строковые ресурсы, предоставленные на рисунке 3.1.

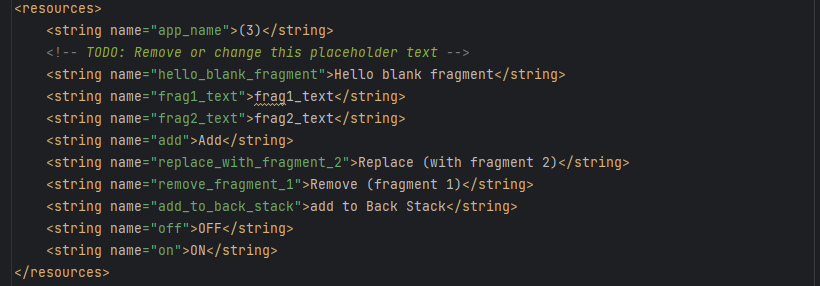


Рисунок 3.1 – Файл strings.xml

Добавим в проект два фрагмента fragment1 и fragment2 любым описанным выше способом. При этом должны быть созданы layout-файлы и классы, наследующие android.fragment.app.Fragment. Xml код данных фрагментов предоставлена на рисунках 3.2 и 3.3.

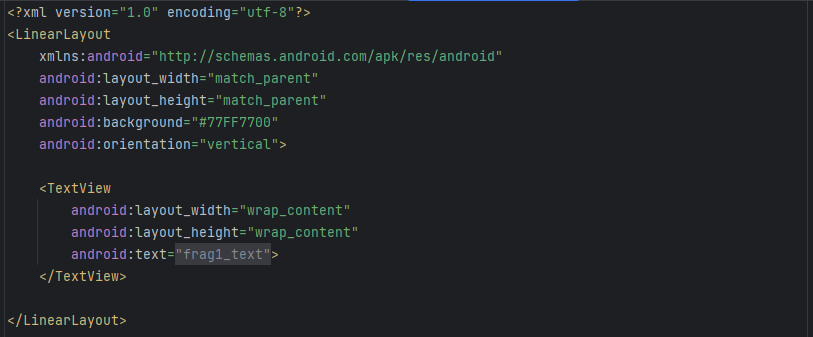


Рисунок 3.2 – Файл fragment\_fist.xml

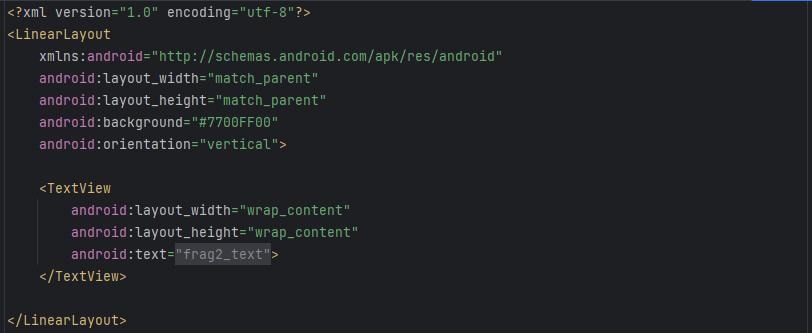


Рисунок 3.4 – Файл fragment\_second.xml

Реализованные классы наших фрагментов предоставлена на рисунках 3.5 и 3.6.

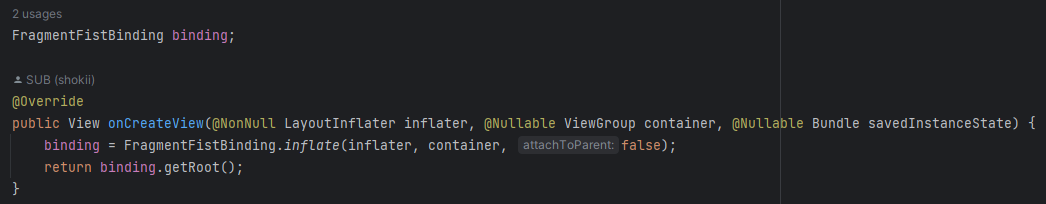


Рисунок 3.5 – Файл FirstFragment.java

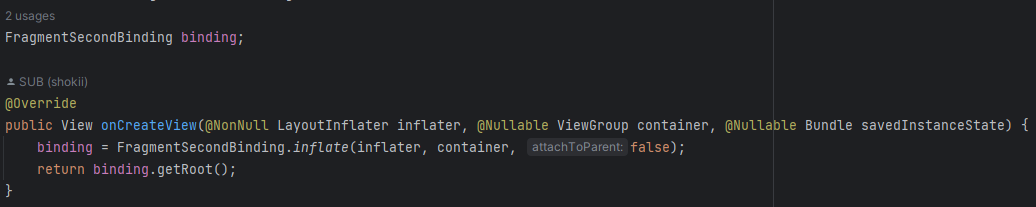


Рисунок 3.6 – Файл SecondFragment.java

В главном Activity будут:

* три кнопки для добавления, удаления и замены фрагментов;
* чекбокс для включения использования BackStack;
* FrameLayout – это контейнер типа ViewGroup, в котором будет происходить вся работа с фрагментами.

Реализация данных элементов предоставлена на рисунке 3.7.

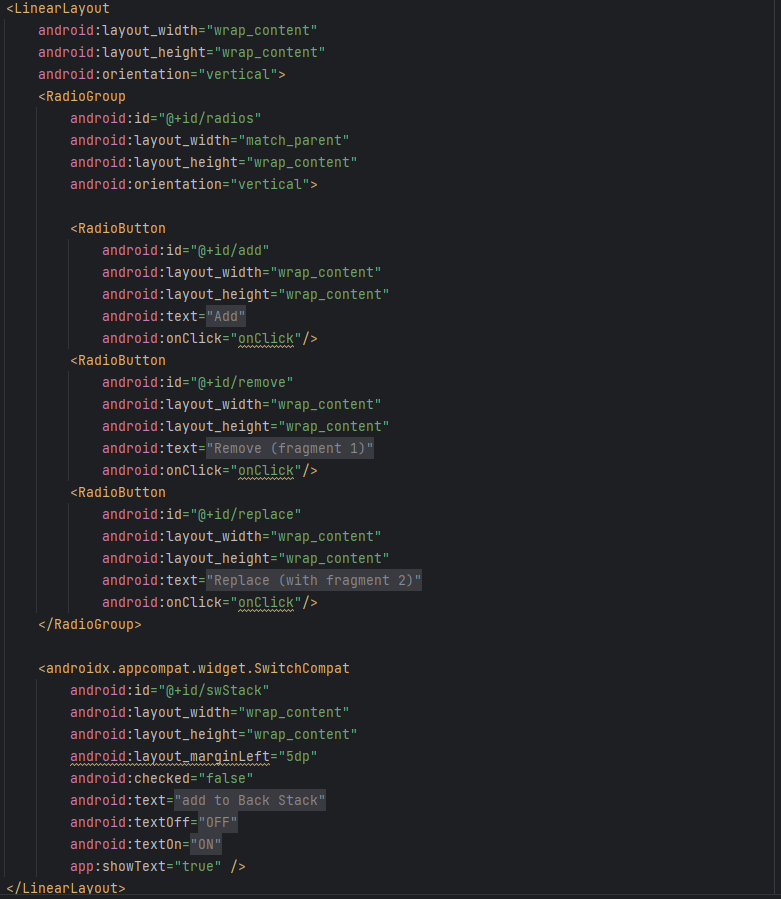


Рисунок 3.7 – Файл activity\_main.xml

Код реализации нашего класса для главного activity предоставлен на рисунке 3.8.



Рисунок 3.8 – Часть кода файла MainActivity.java

Запустим наше приложение на виртуальном устройстве. Отображение на устройстве предоставлено на рисунке 3.9.



Рисунок 3.9 – Приложение запущенное на устройстве

**Задание 3 – Взаимодействие с Activity**

В strings.xml добавьте строковые ресурсы, предоставленные на рисунке 4.1.

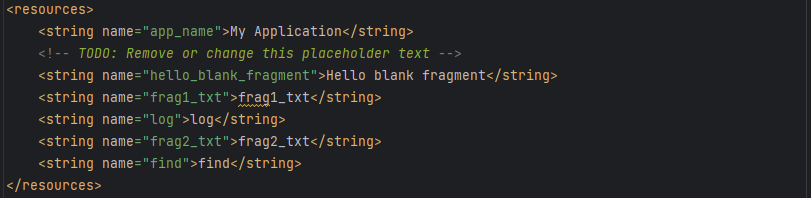


Рисунок 4.1 – Файл stings.xml

Добавим в проект два фрагмента fragment1 и fragment2 любым описанным выше способом. При этом должны быть созданы layout-файлы и классы, наследующие android.fragment.app.Fragment. Xml код данных фрагментов предоставлена на рисунках 4.2 и 4.3.

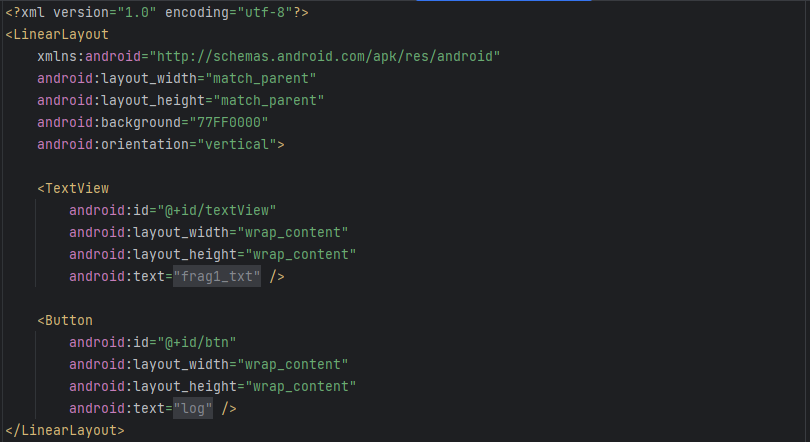


Рисунок 4.2 – Файл fragment\_fist.xml



Рисунок 4.3 – Файл fragment\_second.xml

Реализованные классы наших фрагментов предоставлена на рисунках 4.5 и 4.5.



Рисунок 4.4 – Файл FirstActivity.java

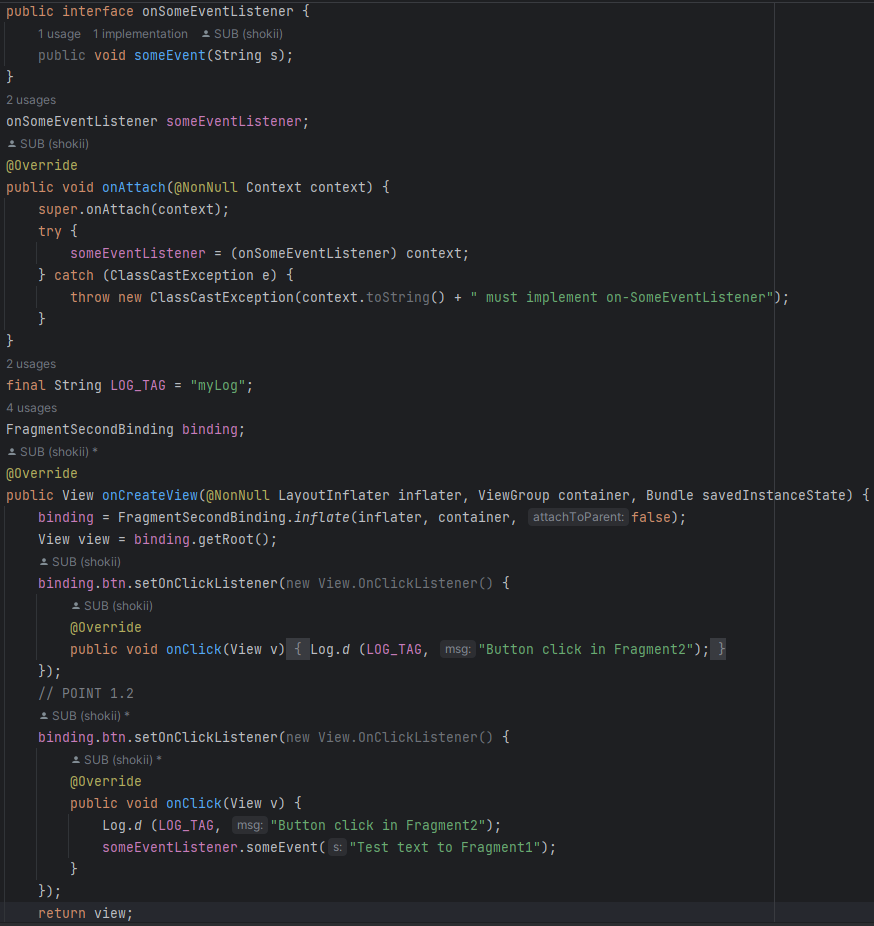


Рисунок 4.5 – Файл SecondActivity.java

Настроем основное Activity в соответствии с заданием. Xml код предоставлен на рисунке 4.6.

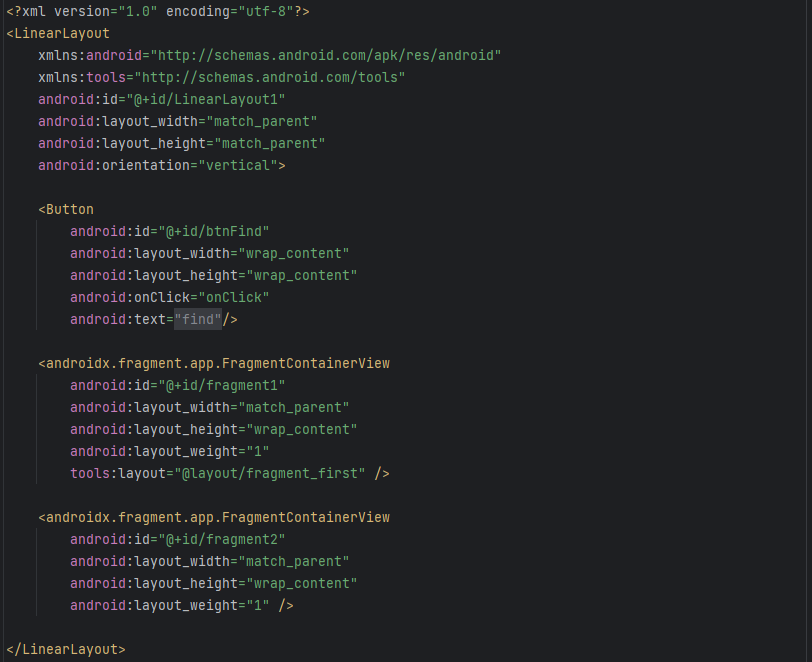


Рисунок 4.6 – Файл activity\_main.xml

Код реализации нашего класса для главного activity предоставлен на рисунке 4.7.

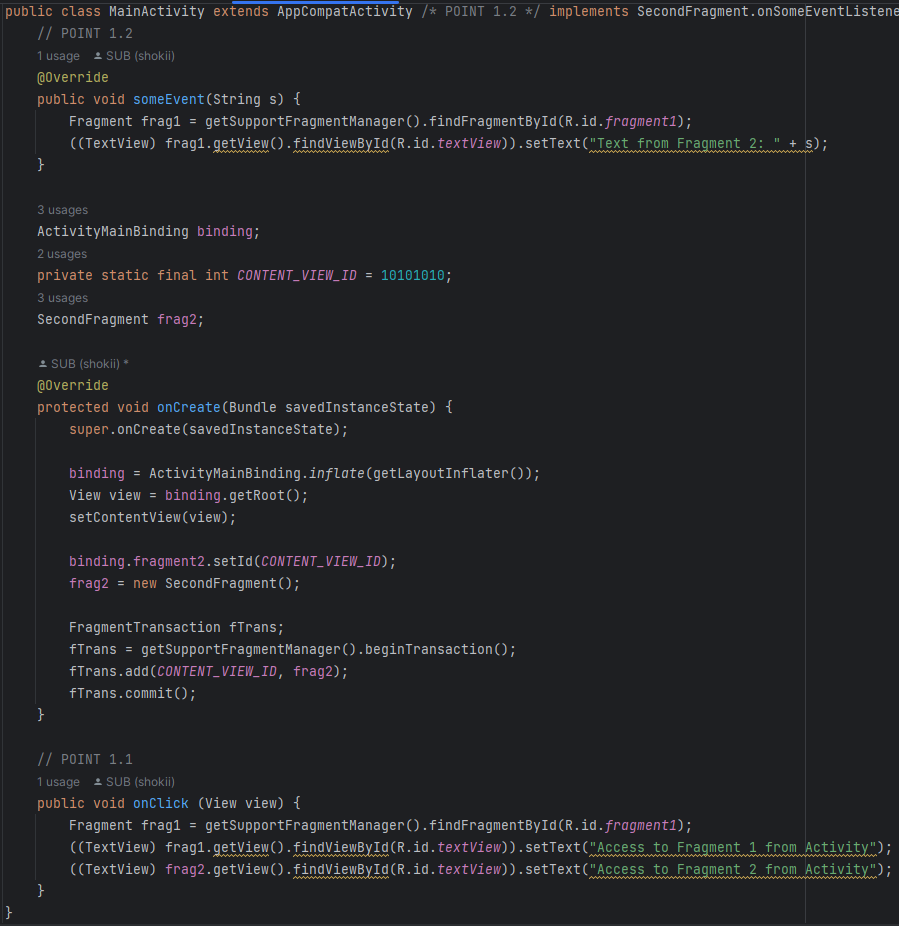


Рисунок 4.7 – Файл MainActivity.java

Запустим наше приложение на виртуальном устройстве. Отображение на устройстве предоставлено на рисунке 4.8.

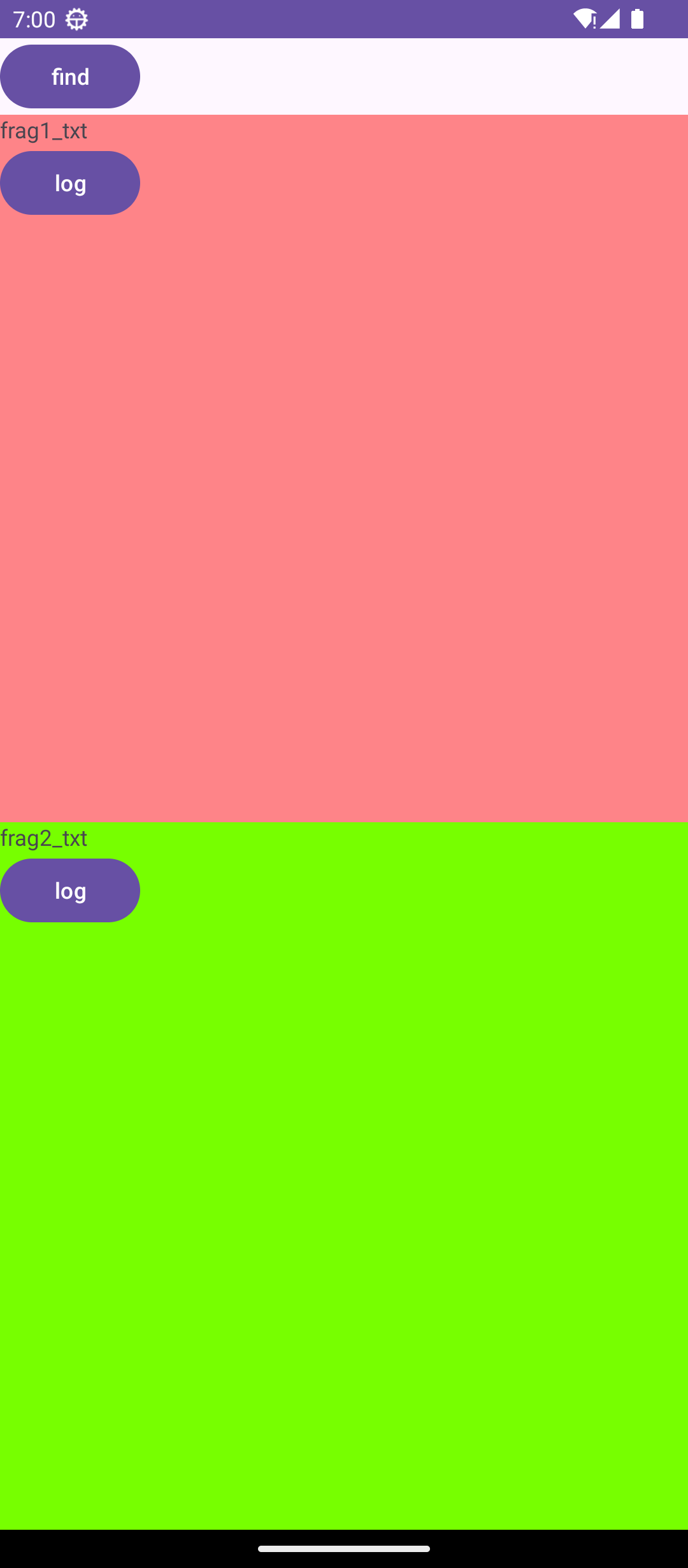


Рисунок 4.8 – Приложение запущенное на устройстве

Логи которые вывела наша программа предоставлена на рисунке 1.8.

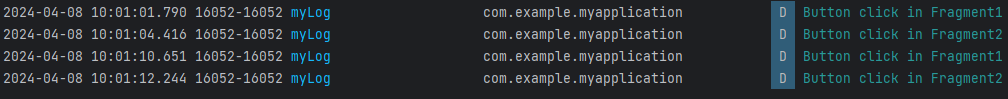


Рисунок 4.9 – Логи приложения

**Практическое задание для самостоятельной разработки**

**Задание 1**

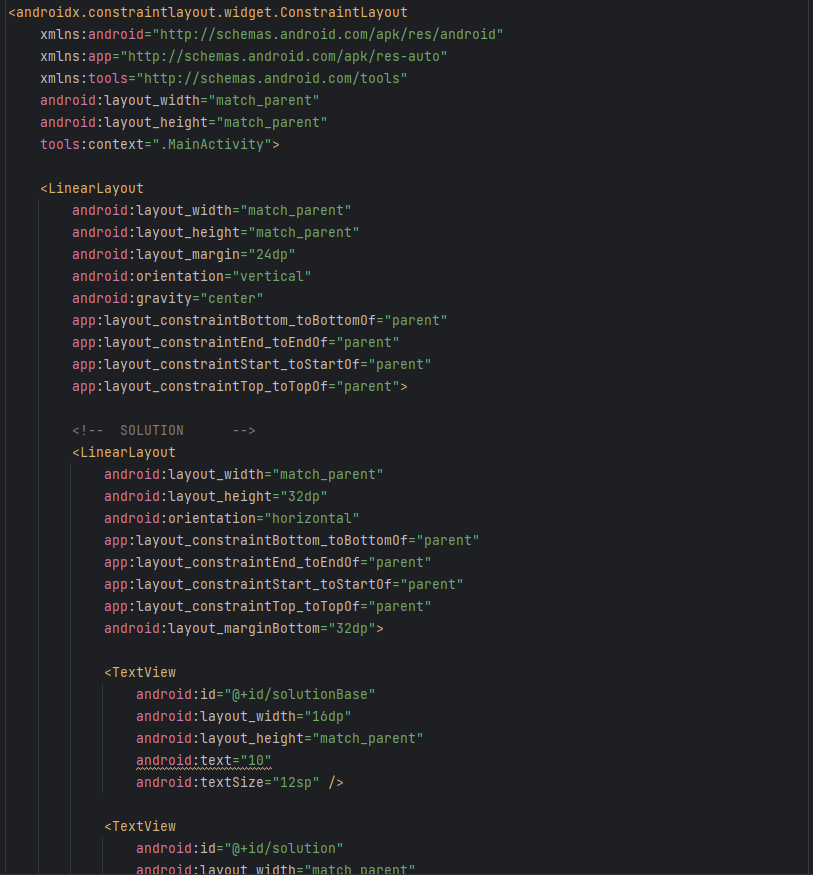


Рисунок 5.1 – Часть кода файла activity\_main.xml



Рисунок 5.2 – Часть кода файла MainACtivity.java

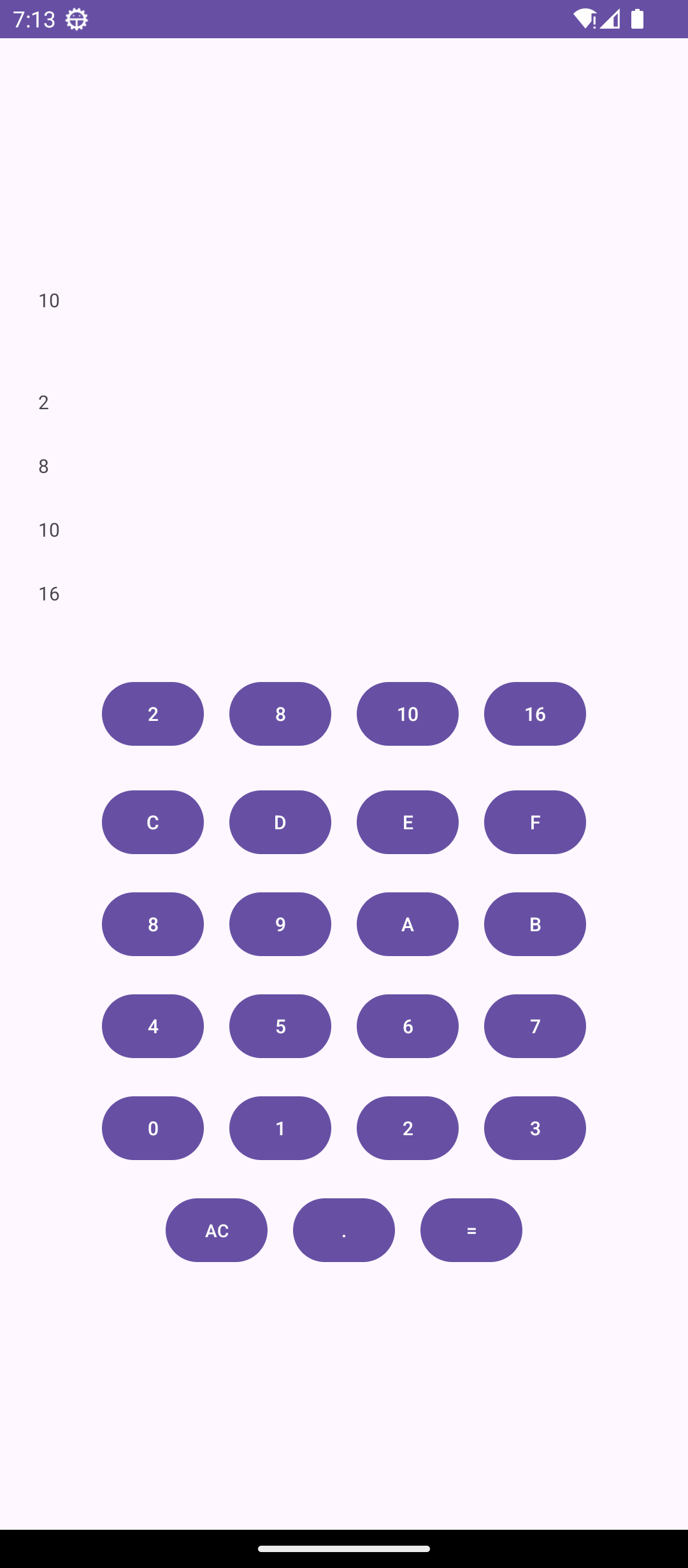


Рисунок 5.3 – Приложение запущенное на устройстве

**Задание 2**



Рисунок 6.1 – Файл activity\_main.xml

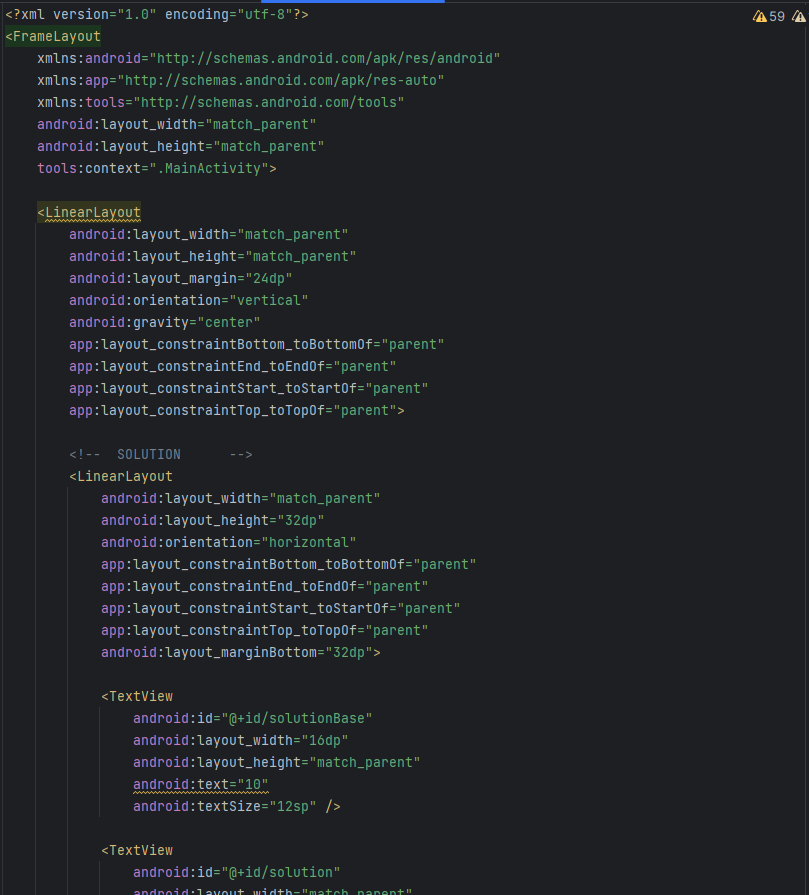


Рисунок 6.2 – Часть файла fragment\_blank.xml

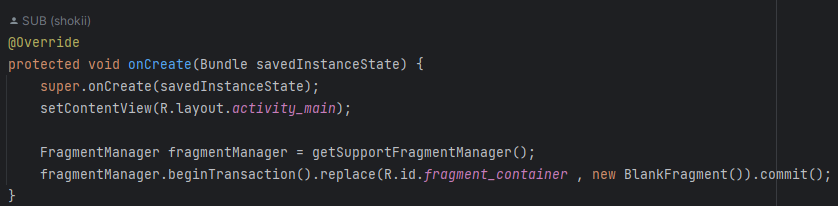


Рисунок 6.3 – Файл MainActivity.java

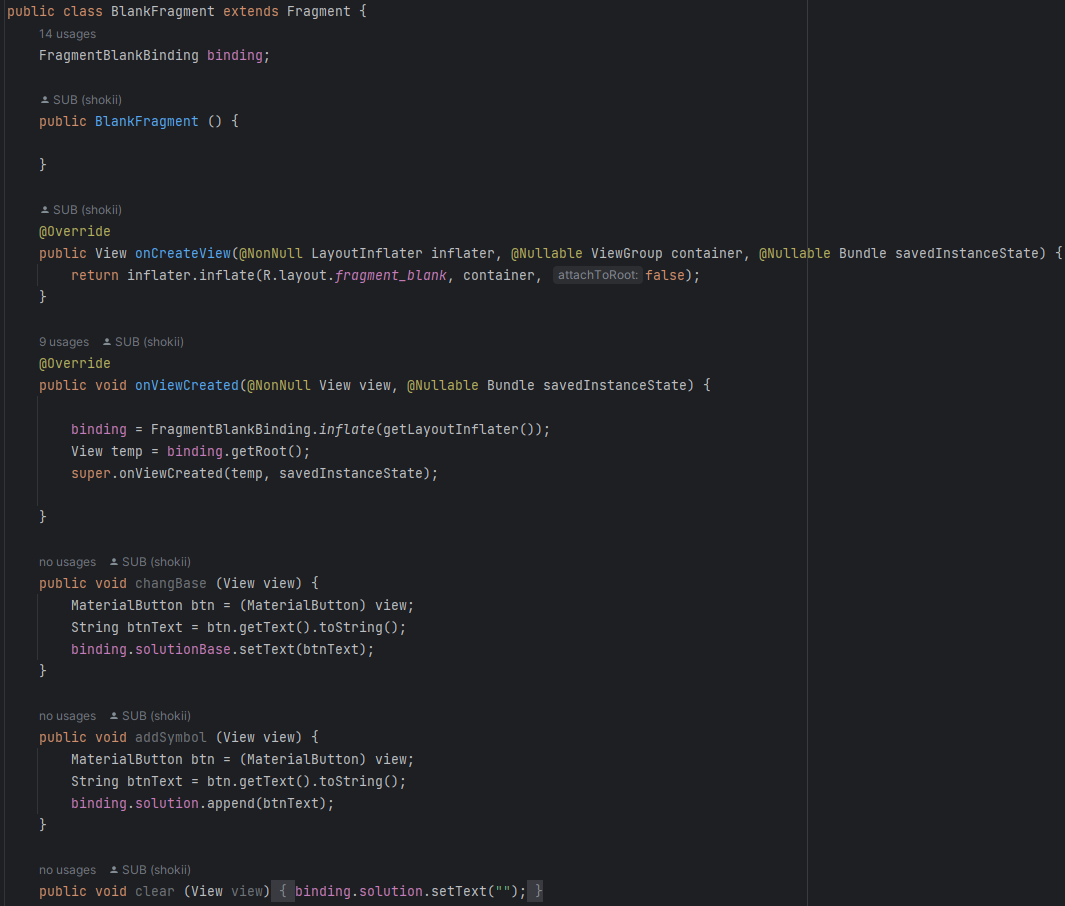


Рисунок 6.4 – Часть кода файла BlankFragment.java

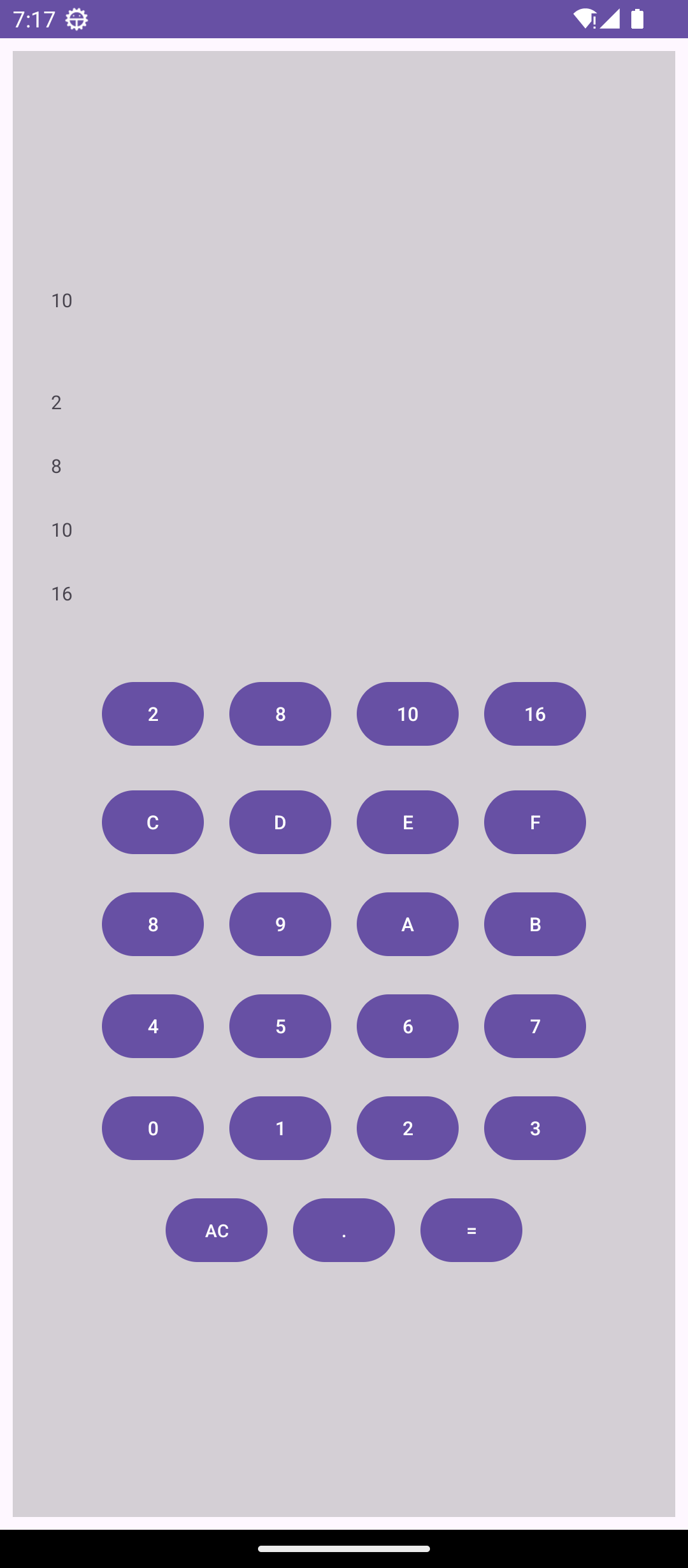


Рисунок 6.5 – Приложение запущенное на устройстве

**Вывод**

В процессе выполнения практической работы №5 были освоены базовые навыки по работе с фрагментами в Android-приложениях.

В ходе работы были рассмотрены основные концепции и методы как можно проводить взаимодействие между activity и фрагментами, а также способы передачи данных между ними.

Получено умение создавать и работать с фрагментами внутри activity, что позволяет создавать более гибкие и масштабируемые пользовательские интерфейсы.