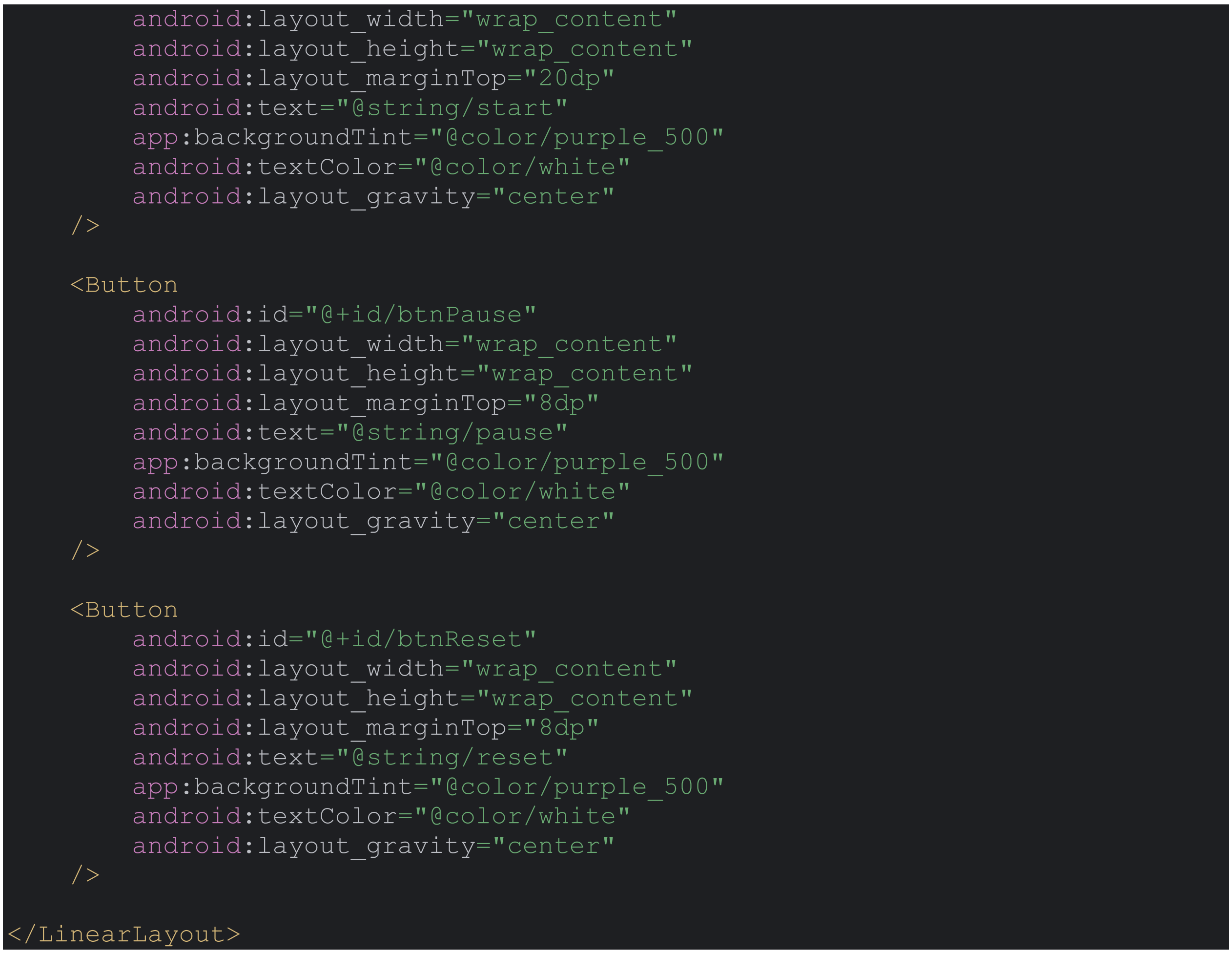
Практическая 6

MainActivity.kt:

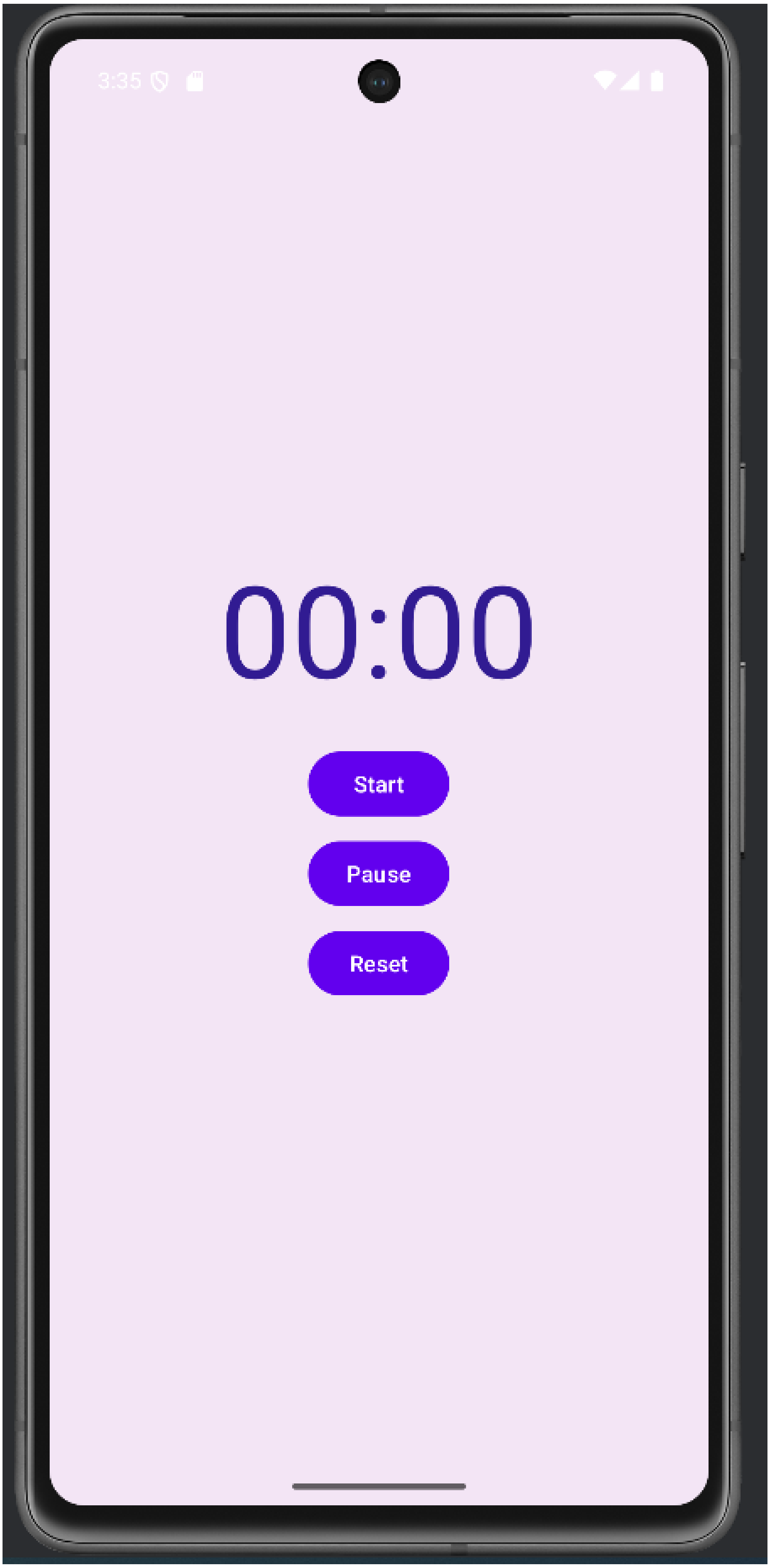
```
package com.example.kekw3

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.SystemClock
import android.widget.Button
import android.widget.Chronometer

class MainActivity : AppCompatActivity() {

    // Объявляем переменные для виджетов и состояния
    lateinit var chronometr: Chronometer // Сам секундомер (Chronometer)
    var running: Boolean = false        // Флаг, работает ли секундомер
    var offset: Long = 0                // Смещение времени для паузы

    // Ключи для сохранения состояния в Bundle
    val OFFSET_KEY = "offset"
    val RUNNING_KEY = "running"
    val BASE_KEY = "base_key"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Получаем ссылки на виджеты из макета
        chronometr = findViewById(R.id.textTime)
        val btnStart: Button = findViewById(R.id.btnStart)
        val btnPause: Button = findViewById(R.id.btnPause)
        val btnReset: Button = findViewById(R.id.btnReset)

        // Восстанавливаем состояние, если оно было сохранено (например,
после поворота экрана)
        if (savedInstanceState != null) {
            offset = savedInstanceState.getLong(OFFSET_KEY)
            running = savedInstanceState.getBoolean(RUNNING_KEY)

            if (running) {
                // Если секундомер работал, восстанавливаем его базовое время
и запускаем
                chronometr.base = savedInstanceState.getLong(BASE_KEY)
                chronometr.start()
            } else {
                // Если был на паузе, просто устанавливаем время
                setBaseTime()
            }
        }

        // Обработчик нажатия на кнопку "Start"
        btnStart.setOnClickListener {
            if (!running) {
                setBaseTime()
                chronometr.start()
                running = true
            }
        }

        // Обработчик нажатия на кнопку "Pause"
        btnPause.setOnClickListener {
            if (running) {
                saveOffset()
                chronometr.stop()
                running = false
            }
        }

        // Обработчик нажатия на кнопку "Reset"
        btnReset.setOnClickListener {
            offset = 0
            setBaseTime()
            chronometr.stop() // Останавливаем на случай, если он работал
            running = false
        }
    }

    // Метод для сохранения состояния перед уничтожением Activity (например,
при повороте)
    override fun onSaveInstanceState(outState: Bundle) {
        // Сохраняем все важные переменные
        outState.putLong(OFFSET_KEY, offset)
        outState.putBoolean(RUNNING_KEY, running)
        outState.putLong(BASE_KEY, chronometr.base)
        super.onSaveInstanceState(outState)
    }

    // Устанавливает базовое время для секундомера с учетом паузы
    private fun setBaseTime() {
        chronometr.base = SystemClock.elapsedRealtime() - offset
    }

    // Сохраняет прошедшее время (смещение) при постановке на паузу
    private fun saveOffset() {
        offset = SystemClock.elapsedRealtime() - chronometr.base
    }
}
```

activity_main.xml:

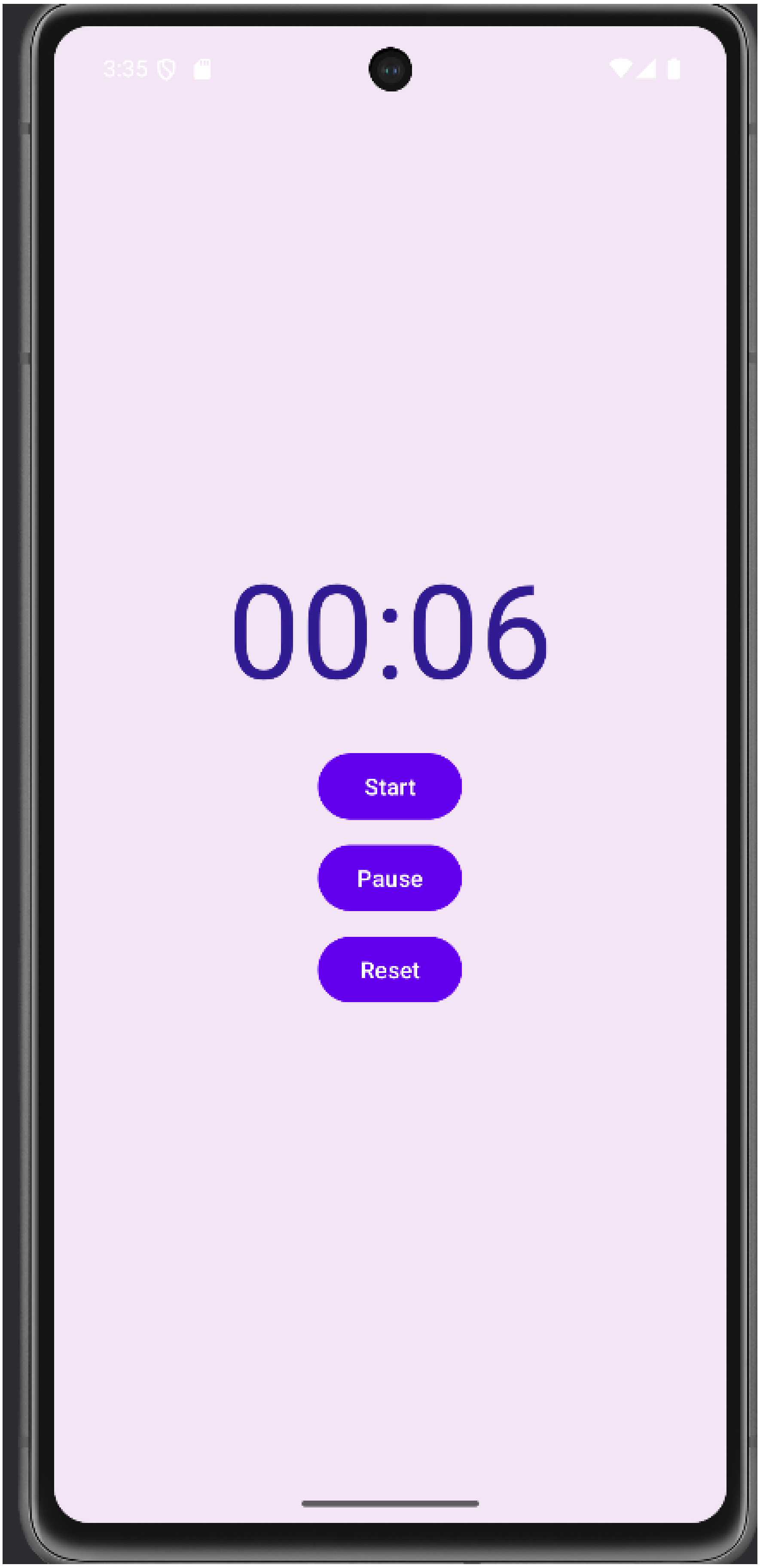
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/kekw3"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp"
    tools:context=".MainActivity"
    android:background="#F3E5F5">

    <Chronometer
        android:id="@+id/textTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="80sp"
        android:textColor="#311B92"
        android:layout_gravity="center"
    />

    <Button
        android:id="@+id/btnStart"
```



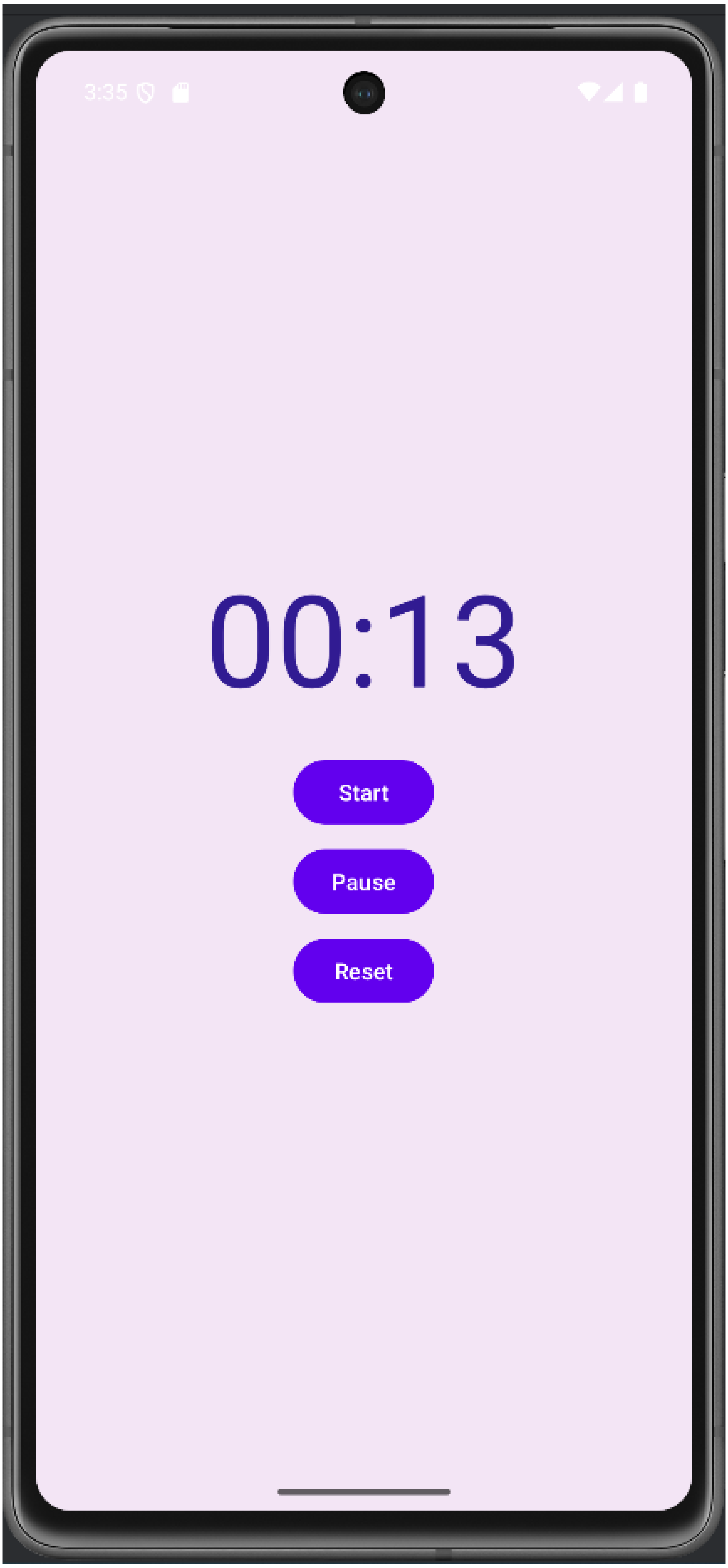
```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="@string/start"
        app:backgroundTint="@color/purple_500"
        android:textColor="@color/white"
        android:layout_gravity="center"
    />

    <Button
        android:id="@+id/btnPause"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="@string/pause"
        app:backgroundTint="@color/purple_500"
        android:textColor="@color/white"
        android:layout_gravity="center"
    />

    <Button
        android:id="@+id/btnReset"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="@string/reset"
        app:backgroundTint="@color/purple_500"
        android:textColor="@color/white"
        android:layout_gravity="center"
    />

</LinearLayout>
```



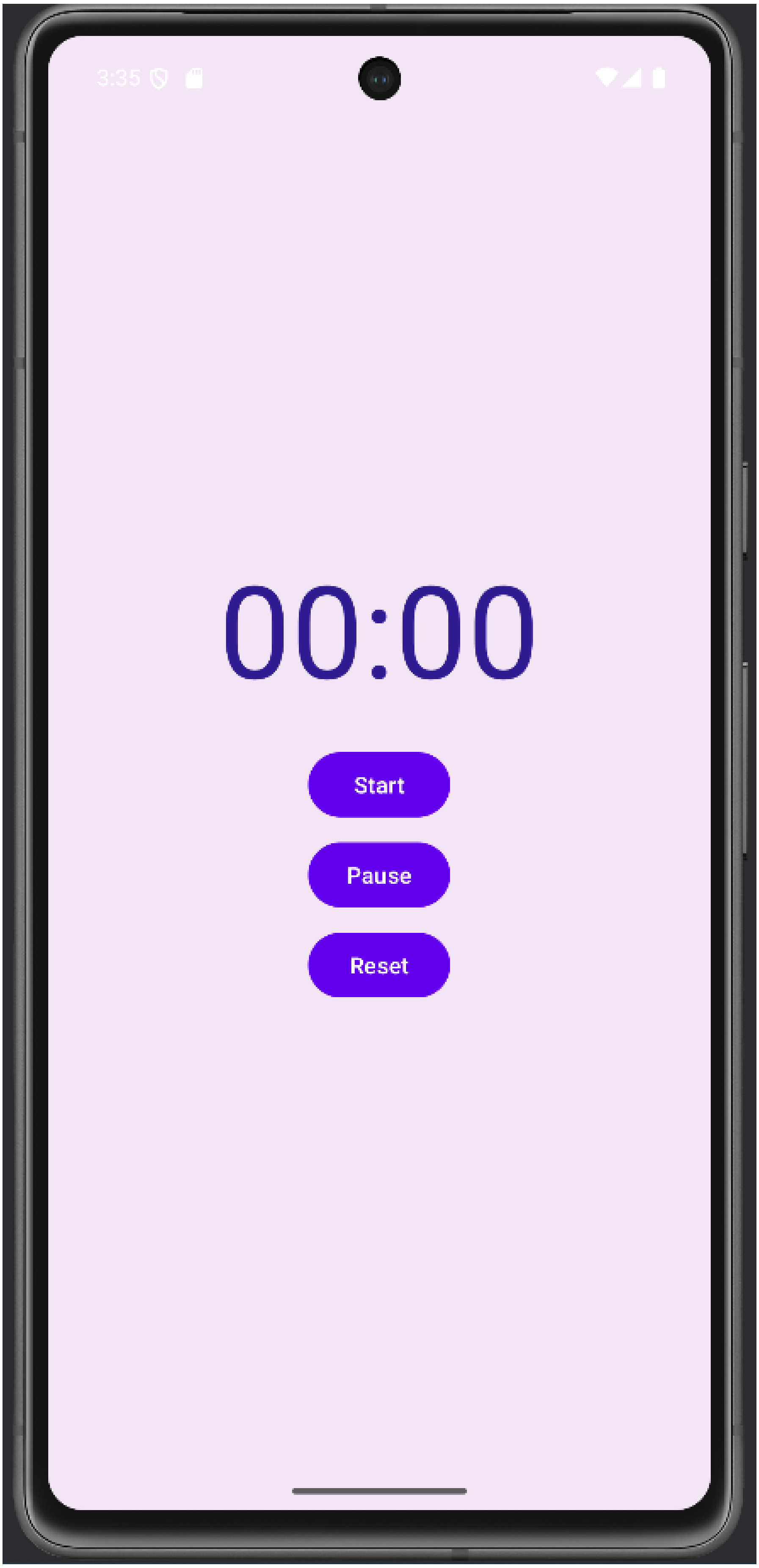
Start



3:35
00:06
Start
Pause
Reset



Pause



Reset