Практическая работа №6

1. Создание таймера main activity.xml

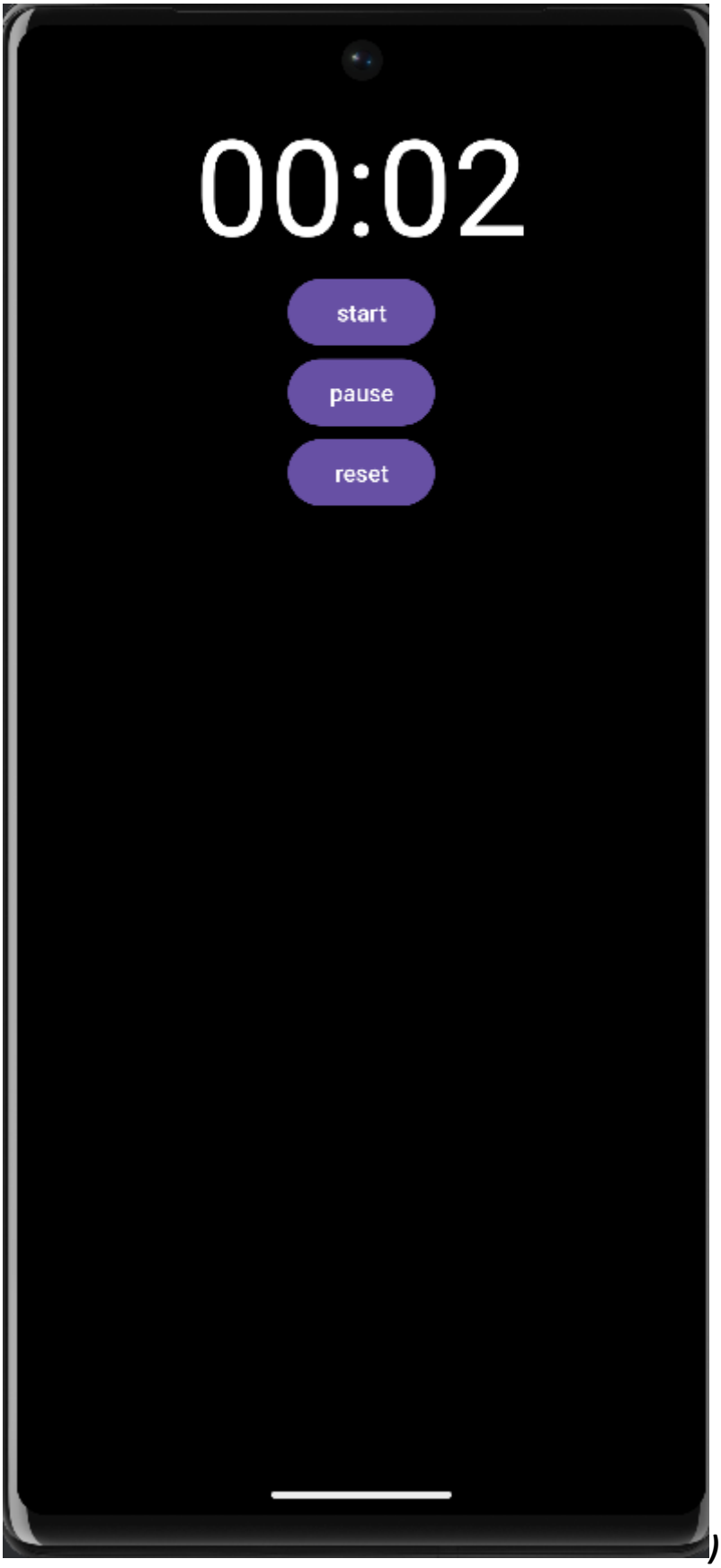
```
<LinearLayout
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/black"
    tools:context=".MainActivity">

    <Chronometer
        android:id="@+id/textTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="80sp"
        android:textColor="@color/white"
        android:layout_gravity="center"
        />
    <Button
        android:id="@+id/btnStart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/s"
        android:layout_gravity="center"
        />
    <Button
        android:id="@+id/btnPause"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/p"
        android:layout_gravity="center"
        />
    <Button
        android:id="@+id/btnReset"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/r"
        android:layout_gravity="center"
        />
</LinearLayout>
```

2.main activity.kt

```
class MainActivity : AppCompatActivity() {
    lateinit var chronometer: Chronometer
    var running: Boolean = false
    var offset: Long = 0
    val OFFSET_KEY = "offset"
    val RUNNING_KEY = "running"
    val BASE_KEY = "base_key"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }
        chronometer = findViewById(R.id.textTime)
        val btnStart = findViewById<Button>(R.id.btnStart)
        val btnPause = findViewById<Button>(R.id.btnPause)
        val btnReset = findViewById<Button>(R.id.btnReset)
        if (savedInstanceState != null) {

            offset = savedInstanceState.getLong(OFFSET_KEY)
            running = savedInstanceState.getBoolean(RUNNING_KEY)
            if (running) {
                chronometer.base = savedInstanceState.getLong(BASE_KEY)
                chronometer.start()
            } else setBaseTime()
        }
```

```
        btnStart.setOnClickListener {
            if (!running) {
                setBaseTime()
                chronometer.start()
                running = true
            }
        }
        btnPause.setOnClickListener {
            if (!running) {
                saveOffset()
                chronometer.stop()
                running = false
            }
        }
        btnReset.setOnClickListener {
            if (!running) {
                offset = 0
                setBaseTime()
                running = false
            }
        }


    }

    private fun saveOffset() {
        offset = SystemClock.elapsedRealtime() - chronometer.base
    }

    private fun setBaseTime() {
        chronometer.base = SystemClock.elapsedRealtime() - offset
    }

    override fun onSaveInstanceState(savedInstanceState: Bundle) {
        savedInstanceState.putLong("offset", offset)
        savedInstanceState.putBoolean("running", running)
        savedInstanceState.putLong("base_key", chronometer.base)
        super.onSaveInstanceState(savedInstanceState)
```

3.результат



)