

## Практическая работа (Приложение секундомер)

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/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp"
    tools:context=".MainActivity"
    android:background="@drawable/mine">

    <Chronometer
        android:id="@+id/textTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="60sp"
        android:textColor="@color/green"
        android:layout_gravity="center"/>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <Button
            android:id="@+id/btnStart"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/start"
            android:backgroundTint="@color/green"
            android:layout_margin="14dp"/>

        <Button
            android:id="@+id/btnPause"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/pause"
            android:backgroundTint="@color/green"
            android:layout_margin="14dp"/>

        <Button
            android:id="@+id/btnReset"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/reset"
            android:backgroundTint="@color/green"
            android:layout_margin="14dp"/>
    </LinearLayout>
</LinearLayout>
```

Colors.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
```

```
<color name="black">#FF000000</color>
<color name="white">#FFFFFFF</color>
<color name="green">#248A1C</color>
</resources>
```

Strings.xml:

```
<resources>
    <string name="app_name">sekundomer</string>
    <string name="start">Start</string>
    <string name="pause">Pause</string>
    <string name="reset">Reset</string>
</resources>
```

MainActivity.kt:

```
package com.example.sekundomer_mironenko
import android.os.Bundle
import android.os.SystemClock
import android.widget.Button
import android.widget.Chronometer
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    private lateinit var chronometr: Chronometer
    private var running = false
    private var offset: Long = 0

    // Ключи для Bundle
    private val OFFSET_KEY = "offset"
    private val RUNNING_KEY = "running"
    private val BASE_KEY = "base"

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

        chronometr = 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) {
                chronometr.base = savedInstanceState.getLong(BASE_KEY)
                chronometr.start()
            } else {
                setBaseTime()
            }
        } else {
            setBaseTime()
        }

        btnStart.setOnClickListener {
            if (!running) {
                setBaseTime()
                chronometr.start()
                running = true
            }
        }
    }
}
```

```

    }

    btnPause.setOnClickListener {
        if (running) {
            saveOffset()
            chronometr.stop()
            running = false
        }
    }

    btnReset.setOnClickListener {
        offset = 0
        setBaseTime()
        running = false
    }
}

override fun onSaveInstanceState(outState: Bundle) {
    outState.putLong(OFFSET_KEY, offset)
    outState.putBoolean(RUNNING_KEY, running)
    outState.putLong(BASE_KEY, chronometr.base)
    super.onSaveInstanceState(outState)
}

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

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

```

