

Лаба 3

Дорошенко

Секундомер

Время до



Время после



Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    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"
    android:background="@drawable/qwer"
    tools:context=".MainActivity">

    <Chronometer
        android:id="@+id/textTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="60sp"
        android:layout_gravity="center"
        android:padding="20dp"
        android:background="#80FFFFFF"
```

```

        android:textColor="#000000"
        android:text="00:00"
    />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="20dp">

        <Button
            android:id="@+id/btnStart"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/start"
            android:layout_marginEnd="10dp"
            android:minWidth="120dp"
            android:backgroundTint="#4CAF50"
            android:textColor="#FFFFFF"
        />

        <Button
            android:id="@+id/btnPause"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/pause"
            android:layout_marginEnd="10dp"
            android:minWidth="120dp"
            android:backgroundTint="#000000"
            android:textColor="#FFFFFF"
        />

        <Button
            android:id="@+id/btnReset"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/reset"
            android:minWidth="120dp"
            android:backgroundTint="#F44336"
            android:textColor="#FFFFFF"
        />

    </LinearLayout>
</LinearLayout>

```

MainActivity.kt

```

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

class MainActivity : AppCompatActivity() {

    private lateinit var chronometer: Chronometer
    private var isRunning: Boolean = false
    private var offset: Long = 0

    companion object {
        private const val OFFSET_KEY = "offset"
        private const val RUNNING_KEY = "running"
    }
}

```

```

        private const val BASE_KEY = "base"
    }

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

        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)

        // Восстановление состояния при повороте экрана
        if (savedInstanceState != null) {
            offset = savedInstanceState.getLong(OFFSET_KEY)
            isRunning = savedInstanceState.getBoolean(RUNNING_KEY)

            if (isRunning) {
                chronometer.base = savedInstanceState.getLong(BASE_KEY)
                chronometer.start()
            } else {
                setBaseTime()
            }
        }

        btnStart.setOnClickListener {
            if (!isRunning) {
                setBaseTime()
                chronometer.start()
                isRunning = true
            }
        }

        btnPause.setOnClickListener {
            if (isRunning) {
                saveOffset()
                chronometer.stop()
                isRunning = false
            }
        }

        btnReset.setOnClickListener {
            offset = 0
            setBaseTime()
            chronometer.stop()
            isRunning = false
        }
    }

    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        if (isRunning) {
            saveOffset()
        }
        outState.putLong(OFFSET_KEY, offset)
        outState.putBoolean(RUNNING_KEY, isRunning)
        outState.putLong(BASE_KEY, chronometer.base)
    }

    override fun onPause() {
        super.onPause()
        if (isRunning) {
            saveOffset()
            chronometer.stop()
        }
    }
}

```

```

override fun onResume() {
    super.onResume()
    if (isRunning) {
        setBaseTime()
        chronometer.start()
    }
}
private fun saveOffset() {
    offset = SystemClock.elapsedRealtime() - chronometer.base
}
private fun setBaseTime() {
    chronometer.base = SystemClock.elapsedRealtime() - offset
}
}

```

strings.xml

```

<resources>
    <string name="app_name">Secundomer_gleb</string> <string
name="start">Start</string>
    <string name="pause">pause</string>
    <string name="reset">Reset</string>
</resources>

```