

## Практика №6

Был написан секундомер

```
activity_main.xml  strings.xml  MainActivity.kt
1 package com.example.myapplication
2
3 import android.os.Bundle
4 import android.os.SystemClock
5 import android.widget.Button
6 import android.widget.Chronometer
7 import androidx.activity.enableEdgeToEdge
8 import androidx.appcompat.app.AppCompatActivity
9 import androidx.core.view.ViewCompat
10 import androidx.core.view.WindowInsetsCompat
11
12 class MainActivity : AppCompatActivity() {
13     lateinit var chronometer: Chronometer
14     var running = Boolean = false
15     var offset : Long = 0
16     override fun onCreate(savedInstanceState: Bundle?) {
17         super.onCreate(savedInstanceState)
18         enableEdgeToEdge()
19         setContentView(R.layout.activity_main)
20         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
21             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
22             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
23             insets
24         }
25         chronometer = findViewById(R.id.textTime)
26         val btnStart = findViewById<Button>(R.id.btnStart)
27         val btnPause = findViewById<Button>(R.id.btnPause)
28         val btnReset = findViewById<Button>(R.id.btnReset)
29
30         btnStart.setOnClickListener {
31             if (!running) {
32                 saveOffset()
33                 chronometer.start()
34                 running = true
35             }
36         }
37         btnPause.setOnClickListener {
38             if (running) {
39                 saveOffset()
40             }
41         }
42     }
43
44     fun saveOffset() {
45         offset = chronometer.elapsedRealtime()
46     }
47 }
```

```
activity_main.xml  strings.xml  MainActivity.kt
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10     <Button
11         android:id="@+id/btnStart"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:layout_gravity="center"
15         android:text="Start" />
16
17     <Chronometer
18         android:id="@+id/textTime"
19         android:layout_width="wrap_content"
20         android:layout_height="60dp"
21         android:text="01:10" />
22
23     <Button
24         android:id="@+id/btnPause"
25         android:layout_width="wrap_content"
26         android:layout_height="wrap_content"
27         android:layout_gravity="center"
28         android:text="pause" />
29
30     <Button
31         android:id="@+id/btnReset"
32         android:layout_width="wrap_content"
33         android:layout_height="wrap_content"
34         android:layout_gravity="center"
35         android:text="Reset" />
36
37 </LinearLayout>
```

Formats: dimension, enum  
Values: fill\_parent, match\_parent, wrap\_content

Specifies the basic height of the view. This is a required attribute for any view inside of a containing layout manager. Its value may be a dimension (such as "12dip") for a constant height or one of the special

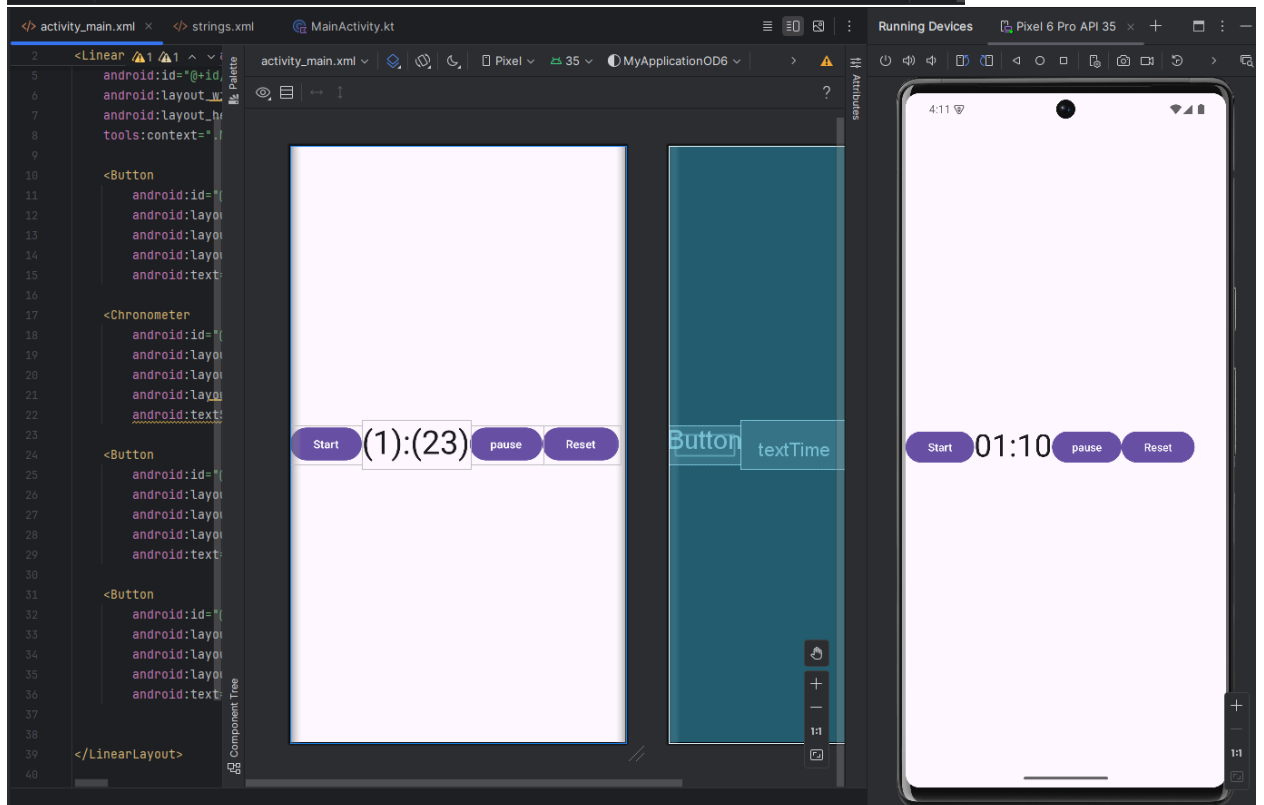
Running Devices | Pixel 6 Pro API 35

4:10

Start 01:10 pause Reset

```
activity_main.xml | strings.xml | MainActivity.kt
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        btnStart.setOnClickListener {
            saveOffset()
            chronometer.start()
            running = true
        }
        btnPause.setOnClickListener {
            if (running) {
                saveOffset()
            }
        }
        btnReset.setOnClickListener {
            offset = 0
            setBaseTime()
            running = false
        }
    }

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



Честно я в шоке, что он даже работает