**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное

учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Отчет по лабораторной работе №10 по дисциплине

**«Разработка программного обеспечения для ОС Android»**

Выполнил:

Студент группы 8И7А \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Леонов

Проверил:

Ст. преп. ОИТ ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Дорофеев

Томск 2020

# Задание

Доработайте приложение из лабораторной работы, дав возможность управлять интервалом времени, через который осуществляется увеличение счётчика, а также сброс или задание начального значения счётчика.

# Текст программы

<https://github.com/8ait/android-course>

**MainActivity.kt**

package com.leonovalexandr.lab10

import android.content.\*

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.os.IBinder

import android.transition.Slide

import android.view.View

import android.widget.EditText

import android.widget.SeekBar

import android.widget.TextView

const val BROADCAST\_TIME\_EVENT = "com.leonovalexandr.lab10.timeevent"

class MainActivity : AppCompatActivity() {

var receiver: BroadcastReceiver? = null

val filter = IntentFilter(BROADCAST\_TIME\_EVENT)

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val seekBar = findViewById<SeekBar>(R.id.seekBar)

val intervalView = findViewById<TextView>(R.id.intervalValue)

seekBar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener{

override fun onProgressChanged(seekBar: SeekBar?, progress: Int, fromUser: Boolean) {

intervalView.text = "Интервал: $progress"

}

override fun onStartTrackingTouch(seekBar: SeekBar?) { }

override fun onStopTrackingTouch(seekBar: SeekBar?) { }

})

receiver = object : BroadcastReceiver() {

// Получено широковещательное сообщение

override fun onReceive(context: Context?, intent: Intent) {

val counter = intent.getIntExtra("counter", 0)

val textCounter = findViewById<TextView>(R.id.counter\_view)

textCounter.text = counter.toString()

}

}

}

override fun onDestroy() {

super.onDestroy()

}

fun buttonStartService(view: View) {

val intent = Intent(this, TimeService::class.java)

val slider = findViewById<SeekBar>(R.id.seekBar)

val intervalValue = slider.progress

val editStart = findViewById<EditText>(R.id.startValue)

val startValue = editStart.text.toString().toIntOrNull()

intent.putExtra("interval", intervalValue)

intent.putExtra("counter", startValue ?: 0)

startService(intent)

bindService(intent, myConnection, Context.BIND\_AUTO\_CREATE)

registerReceiver(receiver, filter)

}

fun buttonStopService(view: View) {

unregisterReceiver(receiver)

unbindService(myConnection)

stopService(Intent(this, TimeService::class.java))

}

var myService: TimeService? = null

var isBound = false

val myConnection = object : ServiceConnection {

override fun onServiceConnected(className: ComponentName, service: IBinder) {

val binder = service as TimeService.MyBinder

myService = binder.getService()

isBound = true

}

override fun onServiceDisconnected(name: ComponentName) {

isBound = false

}

}

}

**TimeService.kt**

package com.leonovalexandr.lab10

import android.app.Service

import android.content.Intent

import android.os.Binder

import android.os.IBinder

import android.util.Log

import kotlinx.coroutines.GlobalScope

import kotlinx.coroutines.Job

import kotlinx.coroutines.delay

import kotlinx.coroutines.launch

class TimeService : Service() {

inner class MyBinder: Binder() {

fun getService() : TimeService {

return this@TimeService

}

}

private var counter = 0

private lateinit var job: Job

private val myBinder = MyBinder()

private var interval:Int = 1

override fun onBind(intent: Intent?): IBinder? {

return myBinder

}

override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {

counter = intent?.getIntExtra("counter", 0) ?: 0

interval = intent?.getIntExtra("interval", 1) ?: 1

job = GlobalScope.launch {

while (true) {

delay(interval \* 1000L)

Log.d("SERVICE", "Timer Is Ticking: " + counter)

counter++

val intent = Intent(BROADCAST\_TIME\_EVENT)

intent.putExtra("counter", counter)

sendBroadcast(intent);

}

}

return super.onStartCommand(intent, flags, startId)

}

override fun onDestroy() {

Log.d("SERVICE", "onDestroy")

job.cancel()

super.onDestroy()

}

}

**activity\_main.xml**

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_height="match\_parent"

android:layout\_width="match\_parent"

android:orientation="vertical"

android:layout\_margin="10dp">

<TextView

android:id="@+id/counter\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:textSize="36dp"

android:text="0"

android:layout\_margin="25dp"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="20dp"

android:text="Начальное значение:"/>

<EditText

android:id="@+id/startValue"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:inputType="number"/>

</LinearLayout>

<TextView

android:id="@+id/intervalValue"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="20dp"

android:text="Интервал: "/>

<SeekBar

android:id="@+id/seekBar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:max="11"

android:min="1"

android:progress="1"

android:layout\_margin="15dp"/>

<Button

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:onClick="buttonStartService"

android:text="Запустить сервис"/>

<Button

android:layout\_width="match\_parent"

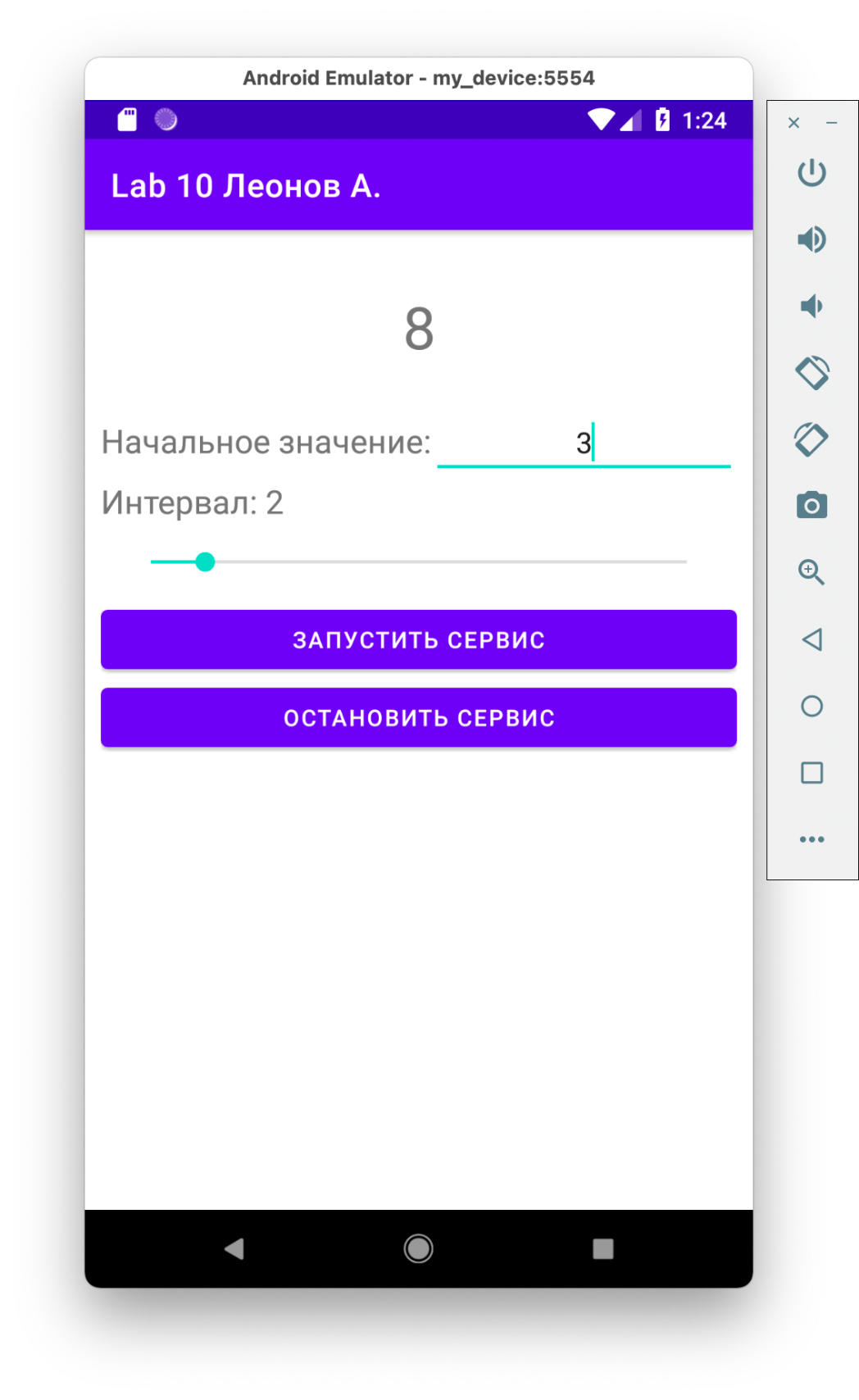
android:layout\_height="wrap\_content"

android:onClick="buttonStopService"

android:text="Остановить сервис"/>

</LinearLayout>

# Результаты работы



# Выводы

В результате выполнения лабораторной работы я познакомился с технологиями организации сервисов в приложениях Android.