|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_***ИУК «Информатика и управление»*\_\_**\_\_\_\_\_\_\_\_\_\_\_\_

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №8**

**ДИСЦИПЛИНА: «Разработка мобильного программного обеспечения»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-62Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Калашников А.С.)  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Шаматрин А.Г.)  (Подпись) (Ф.И.О.) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |
|  | | |

Калуга, 2023

**Целью** выполнения лабораторной работы является формирование практических навыков создания различных android-служб.

**Основными** **задачами** выполнения лабораторной работы являются:

1. Научиться создавать различные службы для мобильного устройства.

2. Уметь понимать схемы взаимодействия службы с другими элементами платформы Android.

3. Разработать эффективные приложения с учетом аппаратных ограничений мобильных устройств

**Вариант №7**

**Задание:**

Создать службу контроля подсветки экрана устройства. Приложение должно иметь возможность установки интенсивности подсветки и установки таймера, по событию которого интенсивность подсветки должна измениться. Также необходимо предусмотреть возможность включения и отключения службы

**Листинг реализованной программы:**

**MyService.kt**

package com.example.myapplication  
  
import android.app.PendingIntent.getActivity  
import android.app.Service  
import android.content.Context  
import android.content.Intent  
import android.net.Uri  
import android.os.CountDownTimer  
import android.os.IBinder  
import android.provider.Settings  
import android.view.View  
import android.view.WindowManager  
import android.widget.Toast  
  
  
class MyService : Service() {  
  
 override fun onBind(intent: Intent): IBinder? {  
 return null  
 }  
  
 override fun onCreate() {  
 super.onCreate()  
 Toast.makeText(this, "Служба создана", Toast.LENGTH\_SHORT).show()  
 }  
  
 override fun onStartCommand(intent: Intent, flags: Int, startId: Int): Int {  
 val time = intent.getStringExtra("time")  
 val sysBackLightValue = intent.getStringExtra("progress")  
 Task((time!!.toInt()\*1000).toLong(),sysBackLightValue.toString())  
  
 System.out.println((time!!.toInt()\*1000).toLong())  
 Toast.makeText(this, sysBackLightValue, Toast.LENGTH\_SHORT).show()  
 return START\_STICKY  
  
 }  
 override fun onDestroy() {  
 super.onDestroy()  
 Toast.makeText(this, "Служба остановлена", Toast.LENGTH\_SHORT).show()  
 }  
  
 fun Task(time: Long,brightness: String){  
 var cTimer: CountDownTimer? = null  
  
 val brightnessInt = ((brightness.toFloat() \* 255)-20).toInt()  
 Toast.makeText(this, "Запущено задание", Toast.LENGTH\_SHORT).show()  
  
 cTimer = object : CountDownTimer(time, 1000) {  
 override fun onTick(millisUntilFinished: Long) {  
 System.out.println((millisUntilFinished / 1000).toString())  
 }  
 override fun onFinish() {  
 System.out.println("done")  
 System.out.println(brightnessInt.toString())  
 val layoutParams = WindowManager.LayoutParams()  
 val windowManager = getSystemService(Context.WINDOW\_SERVICE) as WindowManager  
 val currentBrightness = Settings.System.getInt(contentResolver, Settings.System.SCREEN\_BRIGHTNESS)  
  
  
  
 }  
 }  
 cTimer.start()  
  
  
 }  
  
}

**AndroidManifest.xml**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <uses-permission android:name="android.permission.WRITE\_SETTINGS" />  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MyApplication"  
 tools:targetApi="31">  
 <service  
 android:name=".MyService"  
 android:enabled="true"  
 android:exported="true"></service>  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

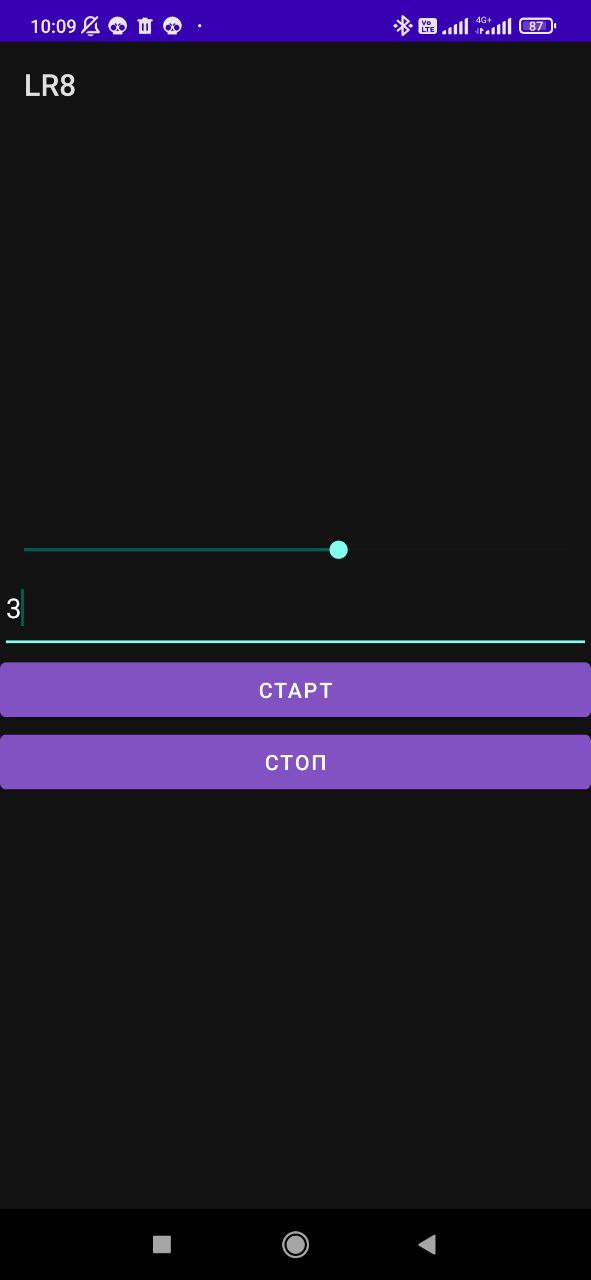
**MainActivity.kt**

package com.example.myapplication  
  
import android.content.Intent  
import android.os.Bundle  
import android.os.CountDownTimer  
import android.view.View  
import android.widget.Button  
import android.widget.EditText  
import android.widget.SeekBar  
import androidx.appcompat.app.AppCompatActivity  
import java.util.\*  
  
  
class MainActivity : AppCompatActivity() {  
  
 var timer: Timer = Timer()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 setContentView(R.layout.activity\_main)  
 title = "SwipeRefreshLayout"  
 val btnStart = findViewById<Button>(R.id.button\_start)  
 val btnStop = findViewById<Button>(R.id.button\_stop)  
 val seekBar: SeekBar = findViewById(R.id.seekBar)  
  
 val editText: EditText = findViewById(R.id.editText)  
 seekBar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener{  
 override fun onProgressChanged(seekBar: SeekBar?, progress: Int, fromUser: Boolean) {  
 }  
 override fun onStartTrackingTouch(seekBar: SeekBar?) { }  
 override fun onStopTrackingTouch(seekBar: SeekBar?) { }  
 })  
 // запуск службы  
 btnStart.setOnClickListener(object: View.OnClickListener {  
 override fun onClick(view: View) {  
 var cTimer: CountDownTimer? = null  
 val serviceIntent: Intent = Intent(this@MainActivity, MyService::class.java)  
  
 serviceIntent.putExtra("time", editText.text.toString());  
 serviceIntent.putExtra("progress", (seekBar.progress.toFloat()/100).toString());  
  
 val lp = window.attributes  
 lp.screenBrightness = (seekBar.progress.toFloat()/100).toFloat()  
 window.attributes = lp  
  
   
 // используем явный вызов службы  
 startService(serviceIntent)  
 }  
 })  
 // остановка службы  
 btnStop.setOnClickListener(object: View.OnClickListener {  
 override fun onClick(view: View) {  
 stopService(Intent(this@MainActivity, MyService::class.java))  
 }  
 })  
 }  
  
  
}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <SeekBar  
 android:id="@+id/seekBar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
 <EditText  
 android:id="@+id/editText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="60dp"/>  
 <Button  
 android:id="@+id/button\_start"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Старт" />  
 <Button  
 android:id="@+id/button\_stop"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Стоп" />  
  
 </LinearLayout>  
  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Результат выполнения работы:**



**Рис. 1.** Работа программы

**Вывод:** в ходе выполнения данной лабораторной работы сформированы практические навыки создания различных android-служб.