|  |  |
| --- | --- |
| 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»***

**ДОМАШНЯЯ РАБОТА**

**«Сервисы»**

**ДИСЦИПЛИНА: «Разработка мобильного ПО»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-52Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_Боков А.А.\_\_\_\_\_)  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_Прудяк П.Н.\_\_\_\_\_)  (Подпись) (Ф.И.О.) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |
| Калуга, 2024 г. | | |

**Цель:** формирование практических навыков создания различных android-служб.

**Задачи:**

1. Научиться создавать различные службы для мобильного

устройства.

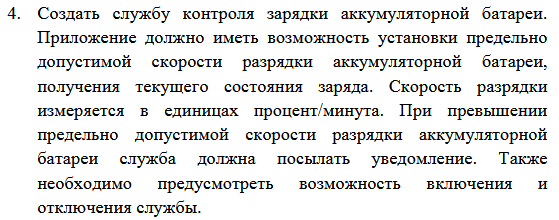
2. Уметь понимать схемы взаимодействия службы с другими

элементами платформы Android.

3. Разработать эффективные приложения с учетом аппаратных

ограничений мобильных устройств.

**Вариант 4**



**Листинг программы:**

***MainActivity.java:***

package com.example.android\_dev\_hw;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.os.BatteryManager;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 private EditText maxDischargeRateInput;  
 private Button startServiceButton, stopServiceButton;  
 private TextView batteryLevelText;  
 private BroadcastReceiver batteryReceiver;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 maxDischargeRateInput = findViewById(R.id.*edit\_max\_rate*);  
 startServiceButton = findViewById(R.id.*button\_start\_service*);  
 stopServiceButton = findViewById(R.id.*button\_stop\_service*);  
 batteryLevelText = findViewById(R.id.*text\_battery\_level*);  
  
 startServiceButton.setOnClickListener(v -> {  
 String rateInput = maxDischargeRateInput.getText().toString();  
 if (!rateInput.isEmpty()) {  
 float maxRate = Float.*parseFloat*(rateInput);  
 Intent intent = new Intent(this, BatteryService.class);  
 intent.putExtra("maxRate", maxRate);  
 startService(intent);  
 Toast.*makeText*(this, "Service Started", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(this, "Please enter a max discharge rate", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 stopServiceButton.setOnClickListener(v -> {  
 stopService(new Intent(this, BatteryService.class));  
 Toast.*makeText*(this, "Service Stopped", Toast.*LENGTH\_SHORT*).show();  
 });  
  
 batteryReceiver = new BroadcastReceiver() {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 int level = intent.getIntExtra(BatteryManager.*EXTRA\_LEVEL*, -1);  
 int scale = intent.getIntExtra(BatteryManager.*EXTRA\_SCALE*, -1);  
 if (level != -1 && scale != -1) {  
 int batteryPct = (int) ((level / (float) scale) \* 100);  
 batteryLevelText.setText("Current Battery Level: " + batteryPct + "%");  
 }  
 }  
 };  
  
 registerReceiver(batteryReceiver, new IntentFilter(Intent.*ACTION\_BATTERY\_CHANGED*));  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 unregisterReceiver(batteryReceiver);  
 }  
}

***BatteryService.java:***

package com.example.android\_dev\_hw;  
  
import android.annotation.SuppressLint;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.Service;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.os.BatteryManager;  
import android.os.Build;  
import android.os.Handler;  
import android.os.IBinder;  
import android.util.Log;  
  
import androidx.core.app.NotificationCompat;  
  
public class BatteryService extends Service {  
  
 private static final String *CHANNEL\_ID* = "BatteryServiceChannel";  
 private float maxDischargeRate = 0;  
 private Handler handler = new Handler();  
 private int lastBatteryLevel = -1;  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
 createNotificationChannel();  
 }  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 if (intent != null && intent.hasExtra("maxRate")) {  
 maxDischargeRate = intent.getFloatExtra("maxRate", 0);  
 }  
  
 // Переводим сервис в передний план  
 startForegroundService();  
  
 monitorBattery();  
 return *START\_STICKY*;  
 }  
  
 @SuppressLint("ForegroundServiceType")  
 private void startForegroundService() {  
 String channelId = "BatteryForegroundService";  
 String channelName = "Battery Monitoring Service";  
  
 if (android.os.Build.VERSION.*SDK\_INT* >= android.os.Build.VERSION\_CODES.*O*) {  
 NotificationChannel channel = new NotificationChannel(  
 channelId,  
 channelName,  
 NotificationManager.*IMPORTANCE\_LOW* );  
 NotificationManager manager = getSystemService(NotificationManager.class);  
 if (manager != null) {  
 manager.createNotificationChannel(channel);  
 }  
 }  
  
 NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(this, channelId)  
 .setSmallIcon(R.drawable.*ic\_battery\_monitoring*) // Ваш значок  
 .setContentTitle("Battery Monitoring")  
 .setContentText("Monitoring battery discharge rate...")  
 .setPriority(NotificationCompat.*PRIORITY\_LOW*)  
 .setOngoing(true);  
  
 startForeground(1, notificationBuilder.build());  
 }  
  
  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 handler.removeCallbacksAndMessages(null);  
 }  
  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
  
 private void monitorBattery() {  
 handler.postDelayed(() -> {  
 int currentBatteryLevel = getBatteryLevel();  
 if (lastBatteryLevel != -1 && currentBatteryLevel < lastBatteryLevel) {  
 int dischargeAmount = lastBatteryLevel - currentBatteryLevel;  
 float dischargeRate = dischargeAmount / 1.0f; // % per minute  
  
 if (dischargeRate > maxDischargeRate) {  
 sendNotification(dischargeRate);  
 }  
 }  
 lastBatteryLevel = currentBatteryLevel;  
 monitorBattery();  
 }, 60000); // Проверяем каждую минуту  
 }  
  
 private int getBatteryLevel() {  
 IntentFilter filter = new IntentFilter(Intent.*ACTION\_BATTERY\_CHANGED*);  
 Intent batteryStatus = registerReceiver(null, filter);  
 if (batteryStatus != null) {  
 return batteryStatus.getIntExtra(BatteryManager.*EXTRA\_LEVEL*, -1);  
 }  
 return -1;  
 }  
  
 private void sendNotification(float dischargeRate) {  
 NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(R.drawable.*ic\_battery\_warning*) // Замените на ваш ресурс  
 .setContentTitle("Battery Discharge Alert")  
 .setContentText("Discharge rate exceeded: " + dischargeRate + "%/min")  
 .setPriority(NotificationCompat.*PRIORITY\_HIGH*)  
 .setAutoCancel(true);  
  
 NotificationManager notificationManager = (NotificationManager) getSystemService(Context.*NOTIFICATION\_SERVICE*);  
 if (notificationManager != null) {  
 notificationManager.notify(1, notificationBuilder.build());  
 }  
 }  
  
 private void createNotificationChannel() {  
 NotificationChannel channel = null;  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 channel = new NotificationChannel(  
 *CHANNEL\_ID*,  
 "Battery Service Channel",  
 NotificationManager.*IMPORTANCE\_HIGH* );  
 }  
 NotificationManager manager = getSystemService(NotificationManager.class);  
 if (manager != null) {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 manager.createNotificationChannel(channel);  
 }  
 }  
 }  
}

***activity\_main.xml***

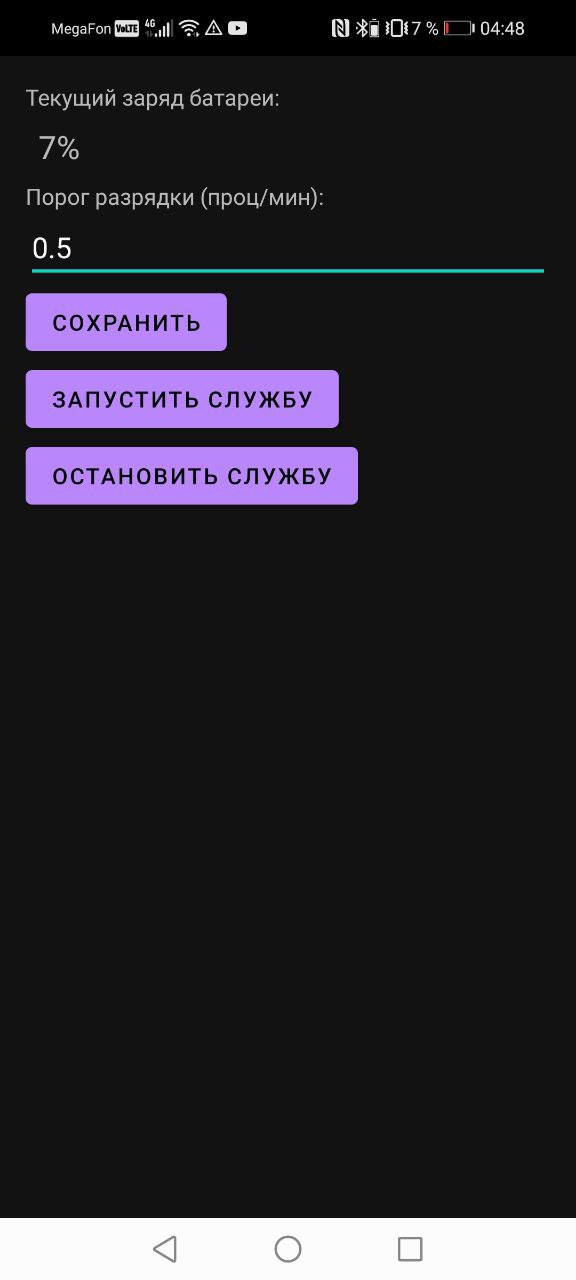
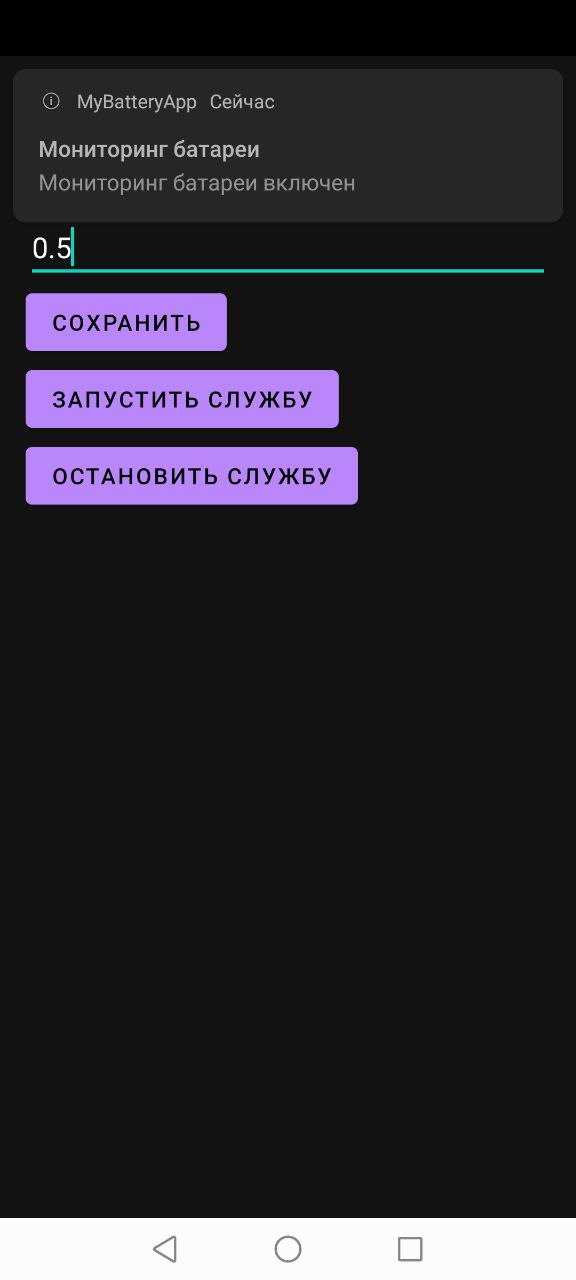
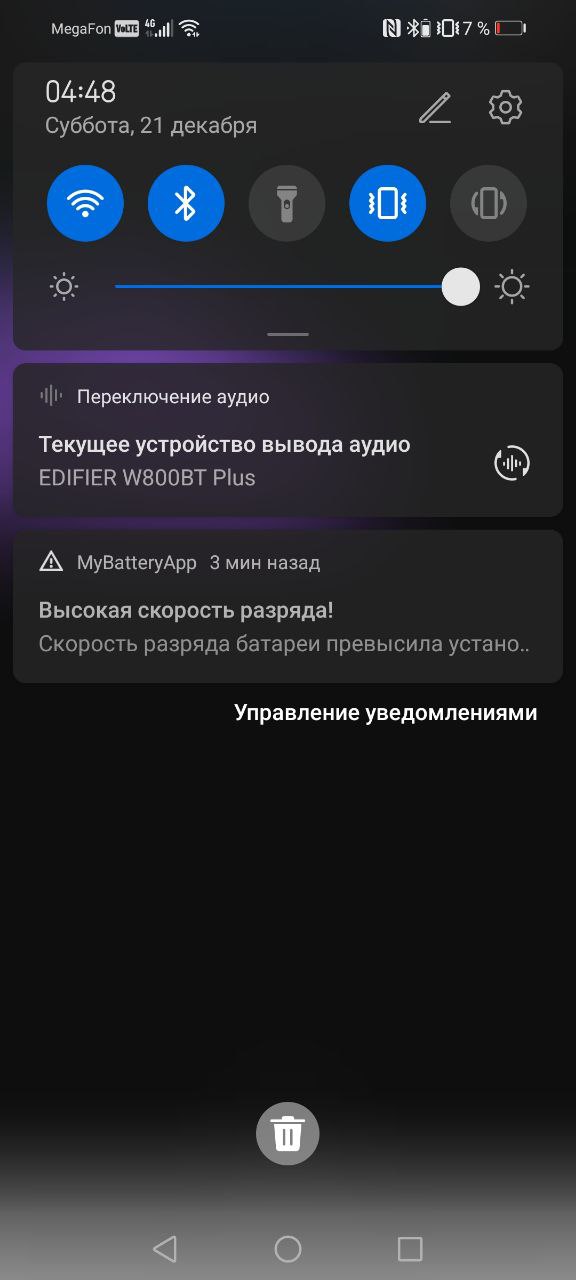
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/text\_battery\_level"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Current Battery Level: --%"  
 android:textSize="18sp"  
 android:layout\_marginBottom="16dp" />  
  
 <EditText  
 android:id="@+id/edit\_max\_rate"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter max discharge rate (%/min)"  
 android:inputType="numberDecimal" />  
  
 <Button  
 android:id="@+id/button\_start\_service"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Start Service"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/button\_stop\_service"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Stop Service"  
 android:layout\_marginTop="8dp" />  
</LinearLayout>

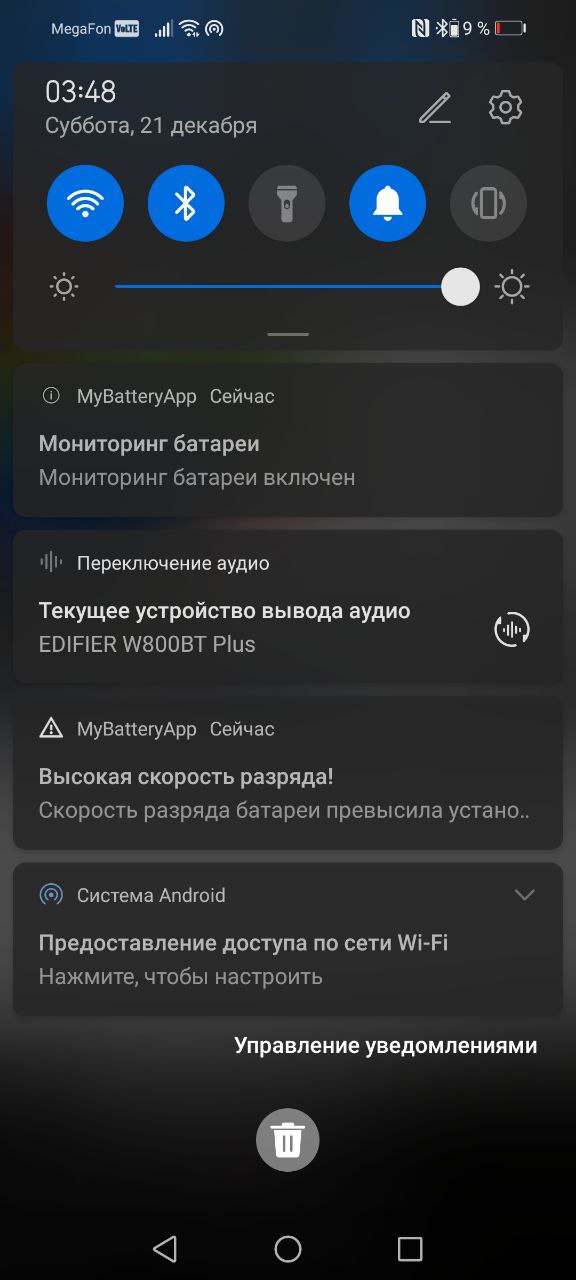
***AndroidManifest.xml:***

***]***

<?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.FOREGROUND\_SERVICE" />  
 <uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />  
 <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:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Android\_dev\_hw"  
 tools:targetApi="31">  
 <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>  
 <service android:name=".BatteryService" />  
 </application>  
  
  
  
</manifest>

**Результаты работы:**

****

****

**Вывод:** в ходе лабораторной работы был реализован сервис для контроля скорости разрядки батареи и пользовательский интерфейс для него.