



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

4. Создать службу контроля зарядки аккумуляторной батареи. Приложение должно иметь возможность установки предельно допустимой скорости разрядки аккумуляторной батареи, получения текущего состояния заряда. Скорость разрядки измеряется в единицах процент/минута. При превышении предельно допустимой скорости разрядки аккумуляторной батареи служба должна посылать уведомление. Также необходимо предусмотреть возможность включения и отключения службы.

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
        .setTitle("Battery Monitoring")
        .setText("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
        .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"
```



```

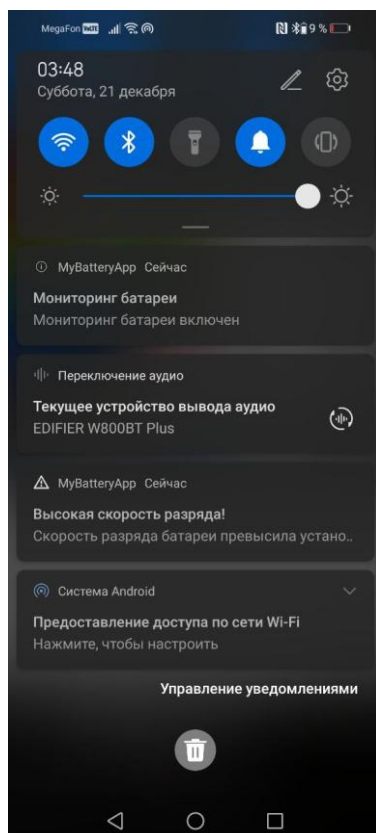
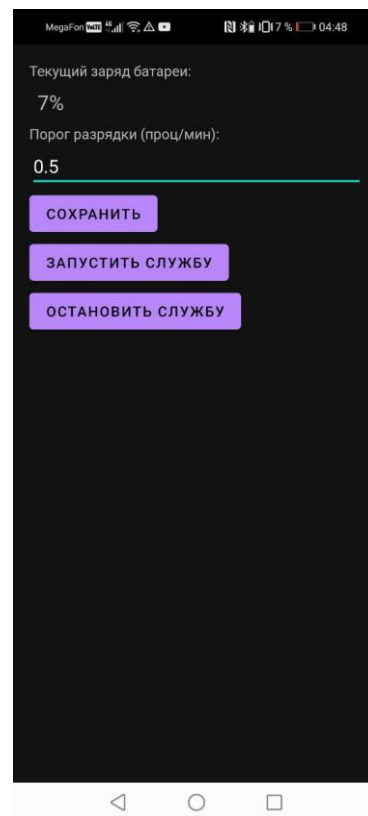
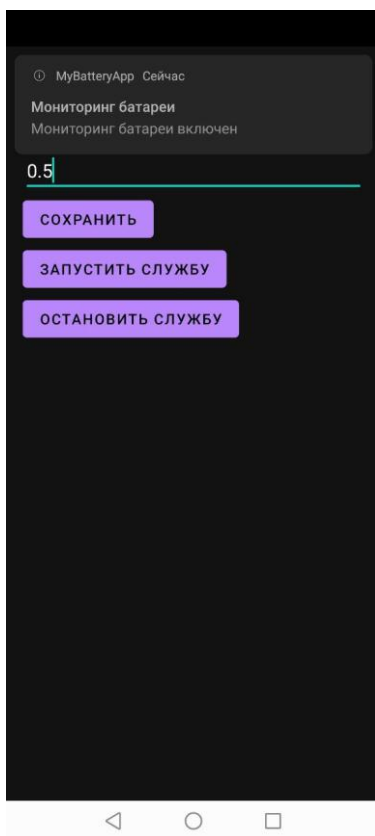
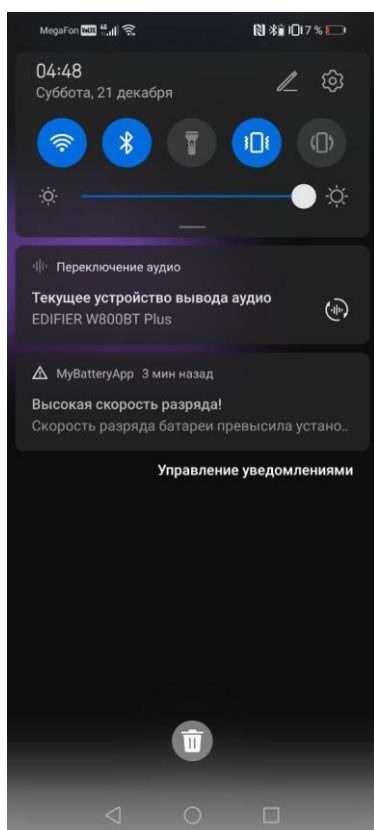
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Android_dev_hw"
    tools:targetApi="31">
    <activity
        android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
            <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.