|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №11**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  студент группыИКБО-28-22 | Некрасов Г.А. |
| **Проверил:**  к.э.н доцент | Петросян Л.Э. |

Москва 2024 г.

**СОДЕРЖАНИЕ**

[Задание 3](#_Toc167627613)

[Выполнение работы 3](#_Toc167627614)

[Тестирование работы 12](#_Toc167627615)

# Задание

* Реализовать в приложении просмотр любой веб-страницы через WebView.
* Реализовать проигрывание музыки из интернета.
* Реализовать различные анимации элементов UI.
* Реализовать отправку уведомлений: обычную отправку и отложенную отправку

# Выполнение работы

Листинг 1. MainActivity.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_11pract;  import android.os.Bundle; import android.content.Intent; import android.webkit.WebView; import android.webkit.WebViewClient; import android.os.Handler; import androidx.appcompat.app.AppCompatActivity;  public class MainActivity extends AppCompatActivity {  private static final long DELAY\_TIME\_MILLIS = 5000; // Время задержки в миллисекундах (в данном случае, 5 секунд)  private WebView webView;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);  webView = (WebView) findViewById(R.id.*webView*);  webView.setWebViewClient(new WebViewClient()); // Помогает приложению открывать ссылки внутри WebView, а не во внешнем браузере  webView.getSettings().setJavaScriptEnabled(true); // Включаем поддержку JavaScript   webView.loadUrl("https://online-edu.mirea.ru "); // Загрузка страницы   // Создаем объект Handler  new Handler().postDelayed(new Runnable() {  @Override  public void run() {  // После заданной задержки запускаем вторую активность  startActivity(new Intent(MainActivity.this, MusicPlayerActivity.class));  // Закрываем текущую активность (опционально)  finish();  }  }, DELAY\_TIME\_MILLIS);  } } |

Листинг 2. Разметка activity\_main.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">   <WebView  android:id="@+id/webView"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent" />  </RelativeLayout> |

Листинг 3. MusicPlayerActivity.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_11pract;  import android.animation.ArgbEvaluator; import android.animation.ObjectAnimator; import android.app.AlarmManager; import android.app.NotificationChannel; import android.app.NotificationManager; import android.app.PendingIntent; import android.content.Context; import android.content.Intent; import android.content.pm.PackageManager; import android.graphics.Color; import android.media.MediaPlayer; import android.os.Build; import android.os.Bundle; import android.util.Log; import android.view.MotionEvent; import android.view.View; import android.view.accessibility.AccessibilityNodeInfo; import android.view.animation.Animation; import android.view.animation.AnimationUtils; import android.widget.Button; import android.widget.TextView; import androidx.appcompat.app.AppCompatActivity; import androidx.core.app.ActivityCompat; import androidx.core.app.NotificationCompat; import androidx.core.content.ContextCompat;  public class MusicPlayerActivity extends AppCompatActivity {   public static final String *CHANNEL\_ID* = "music\_player\_channel";  public static final String *CHANNEL\_ID\_SCHEDULE* = "delayed\_channel";  public static final int *NOTIFICATION\_ID* = 1;   private static final int *PERMISSION\_REQUEST\_CODE* = 100;   private Button buttonPlay;  private MediaPlayer mediaPlayer;  private TextView textSongTitle;  private Animation moveRotateAnimation;   private float dX, dY;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_music\_player*);   buttonPlay = findViewById(R.id.*buttonPlay*);  mediaPlayer = MediaPlayer.*create*(this, R.raw.*ladygaga\_judas*); // замените на URL вашей музыки   textSongTitle = findViewById(R.id.*textSongTitle*);   // Создание канала уведомлений  createNotificationChannel();   // Анимация перемещения и вращения изображения judas  moveRotateAnimation = AnimationUtils.*loadAnimation*(this, R.anim.*move\_rotate\_anim*);  moveRotateAnimation.setRepeatCount(Animation.*INFINITE*); // Бесконечное повторение анимации  findViewById(R.id.*imageView*).startAnimation(moveRotateAnimation);   // Анимация изменения размера и цвета текста названия песни  final ObjectAnimator animatorSize = ObjectAnimator.*ofFloat*(textSongTitle, "scaleX", 0.5f, 2f);  animatorSize.setDuration(3000);  animatorSize.setRepeatCount(ObjectAnimator.*INFINITE*);  animatorSize.setRepeatMode(ObjectAnimator.*REVERSE*);  animatorSize.start();   final ObjectAnimator animatorColor = ObjectAnimator.*ofObject*(textSongTitle, "textColor",  new ArgbEvaluator(), Color.*RED*, Color.*BLUE*);  animatorColor.setDuration(3000);  animatorColor.setRepeatCount(ObjectAnimator.*INFINITE*);  animatorColor.setRepeatMode(ObjectAnimator.*REVERSE*);  animatorColor.start();   // Остановить анимации при уничтожении активности  textSongTitle.addOnAttachStateChangeListener(new View.OnAttachStateChangeListener() {  @Override  public void onViewAttachedToWindow(View view) {}   @Override  public void onViewDetachedFromWindow(View view) {  animatorSize.cancel();  animatorColor.cancel();  }  });   buttonPlay.setOnTouchListener(new View.OnTouchListener() {  @Override  public boolean onTouch(View v, MotionEvent event) {  switch (event.getAction()) {  case MotionEvent.*ACTION\_DOWN*:  dX = v.getX() - event.getRawX();  dY = v.getY() - event.getRawY();  break;  case MotionEvent.*ACTION\_MOVE*:  v.setX(event.getRawX() + dX);  v.setY(event.getRawY() + dY);  break;  case MotionEvent.*ACTION\_UP*:  v.performClick(); // Вызываем performClick() при отпускании кнопки  break;  default:  return false;  }  return true;  }  });   buttonPlay.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (mediaPlayer.isPlaying()) {  mediaPlayer.pause();  buttonPlay.setText("Play");  } else {  mediaPlayer.start();  buttonPlay.setText("Pause");   // Отправка уведомления при начале воспроизведения  sendNotification("Now playing", "Lady Gaga - Judas");   // Добавим вывод в лог для проверки  Log.*d*("MusicPlayerActivity", "Button clicked");  }  }  });   // Для обеспечения правильного обработки нажатий кнопки в режиме доступности  buttonPlay.setAccessibilityDelegate(new View.AccessibilityDelegate() {  @Override  public boolean performAccessibilityAction(View host, int action, Bundle args) {  if (action == AccessibilityNodeInfo.*ACTION\_CLICK*) {  buttonPlay.performClick();  return true;  }  return super.performAccessibilityAction(host, action, args);  }  });   // Обработка нажатия на кнопку для отложенного уведомления  Button btnScheduleNotification = findViewById(R.id.*btnScheduleNotification*);  btnScheduleNotification.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (ContextCompat.*checkSelfPermission*(MusicPlayerActivity.this, android.Manifest.permission.*SET\_ALARM*) != PackageManager.*PERMISSION\_GRANTED*) {  ActivityCompat.*requestPermissions*(MusicPlayerActivity.this, new String[]{android.Manifest.permission.*SET\_ALARM*}, *PERMISSION\_REQUEST\_CODE*);  } else {  scheduleNotification(5000);  }  }  });  }   private void scheduleNotification(long delay) {  Log.*d*("MusicPlayerActivity", "Scheduling notification with delay: " + delay);   Intent notificationIntent = new Intent(this, AlarmReceiver.class);  PendingIntent pendingIntent = PendingIntent.*getBroadcast*(this, 0, notificationIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE*);   AlarmManager alarmManager = (AlarmManager) getSystemService(*ALARM\_SERVICE*);  long futureInMillis = System.*currentTimeMillis*() + delay;   try {  if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M*) {  alarmManager.setExactAndAllowWhileIdle(AlarmManager.*RTC\_WAKEUP*, futureInMillis, pendingIntent);  } else if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*KITKAT*) {  alarmManager.setExact(AlarmManager.*RTC\_WAKEUP*, futureInMillis, pendingIntent);  } else {  alarmManager.set(AlarmManager.*RTC\_WAKEUP*, futureInMillis, pendingIntent);  }  } catch (SecurityException e) {  Log.*e*("MusicPlayerActivity", "Permission denied for setting alarms", e);  }  }   private void createNotificationChannel() {  if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  CharSequence name = "Scheduled Notification Channel";  String description = "Channel for scheduled notifications";  int importance = NotificationManager.*IMPORTANCE\_DEFAULT*;  NotificationChannel channel = new NotificationChannel(*CHANNEL\_ID\_SCHEDULE*, name, importance);  channel.setDescription(description);   NotificationManager notificationManager = getSystemService(NotificationManager.class);  if (notificationManager != null) {  notificationManager.createNotificationChannel(channel);  }  }  }   private void sendNotification(String title, String message) {  // Создание интента для запуска активности при нажатии на уведомление  Intent intent = new Intent(this, MusicPlayerActivity.class);  PendingIntent pendingIntent = PendingIntent.*getActivity*(this, 0, intent, PendingIntent.*FLAG\_IMMUTABLE*);   // Создание уведомления  NotificationCompat.Builder builder = new NotificationCompat.Builder(this, *CHANNEL\_ID\_SCHEDULE*)  .setSmallIcon(R.drawable.*ic\_launcher\_background*)  .setContentTitle(title)  .setContentText(message)  .setContentIntent(pendingIntent)  .setAutoCancel(true);   // Отправка уведомления  NotificationManager notificationManager = (NotificationManager) getSystemService(Context.*NOTIFICATION\_SERVICE*);  notificationManager.notify(*NOTIFICATION\_ID*, builder.build());  }   private void cancelNotification(Context context) {  Intent notificationIntent = new Intent(context, AlarmReceiver.class);  PendingIntent pendingIntent = PendingIntent.*getBroadcast*(context, 0, notificationIntent, PendingIntent.*FLAG\_IMMUTABLE*);   AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.*ALARM\_SERVICE*);  if (alarmManager != null) {  alarmManager.cancel(pendingIntent);  }  }   @Override  protected void onDestroy() {  super.onDestroy();  if (mediaPlayer != null) {  mediaPlayer.stop(); // Остановить воспроизведение музыки  mediaPlayer.release();  mediaPlayer = null;  }  cancelNotification(this); // Отменить запланированное уведомление  }   @Override  public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {  super.onRequestPermissionsResult(requestCode, permissions, grantResults);  if (requestCode == *PERMISSION\_REQUEST\_CODE*) {  if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  // Разрешение было предоставлено, запланируем уведомление  scheduleNotification(5000);  } else {  // Разрешение не было предоставлено, уведомляем пользователя  Log.*e*("MusicPlayerActivity", "Permission denied to set alarms");  }  }  } } |

Листинг 4. Разметка activity\_music\_player.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MusicPlayerActivity">   <ImageView  android:id="@+id/imageView"  android:layout\_width="200dp"  android:layout\_height="200dp"  android:src="@drawable/judas"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="16dp" />   <TextView  android:id="@+id/textSongTitle"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Lady Gaga - Judas"  android:textSize="18sp"  android:layout\_below="@id/imageView"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="8dp" />   <Button  android:id="@+id/buttonPlay"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Play"  android:layout\_below="@id/textSongTitle"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="16dp" />   <Button  android:id="@+id/btnScheduleNotification"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Schedule Notification"  android:layout\_below="@id/buttonPlay"  android:layout\_marginTop="16dp"  android:layout\_centerHorizontal="true"/>  </RelativeLayout> |

Листинг 5. AlarmReceiver.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_11pract;  import android.content.BroadcastReceiver; import android.content.Context; import android.content.Intent; import androidx.core.app.NotificationCompat; import androidx.core.app.NotificationManagerCompat;  public class AlarmReceiver extends BroadcastReceiver {  @Override  public void onReceive(Context context, Intent intent) {  NotificationCompat.Builder builder = new NotificationCompat.Builder(context, MusicPlayerActivity.*CHANNEL\_ID\_SCHEDULE*)  .setSmallIcon(R.drawable.*ic\_launcher\_background*)  .setContentTitle("Delayed Notification")  .setContentText("This is your scheduled notification.")  .setPriority(NotificationCompat.*PRIORITY\_DEFAULT*)  .setAutoCancel(true);   NotificationManagerCompat notificationManager = NotificationManagerCompat.*from*(context);  notificationManager.notify(1, builder.build());  } } |

Листинг 6. Анимация move\_rotate\_anim.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <set xmlns:android="http://schemas.android.com/apk/res/android">  <translate  android:fromXDelta="0%"  android:toXDelta="100%"  android:fromYDelta="0%"  android:toYDelta="100%"  android:duration="3000"  android:repeatCount="infinite" />  <rotate  android:fromDegrees="0"  android:toDegrees="360"  android:pivotX="50%"  android:pivotY="50%"  android:duration="3000"  android:repeatCount="infinite" /> </set> |

# Тестирование работы

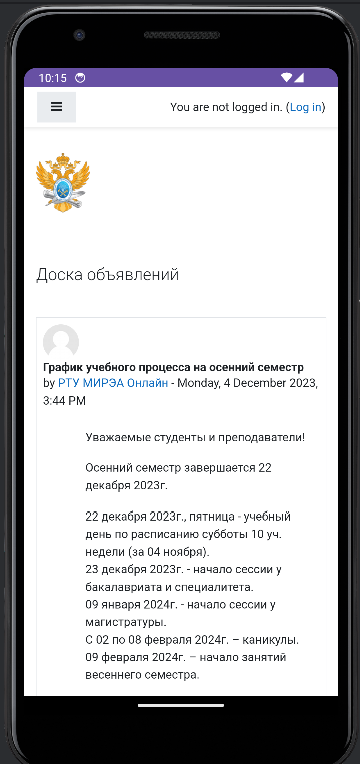


Рисунок 1. Проверка веб-активности

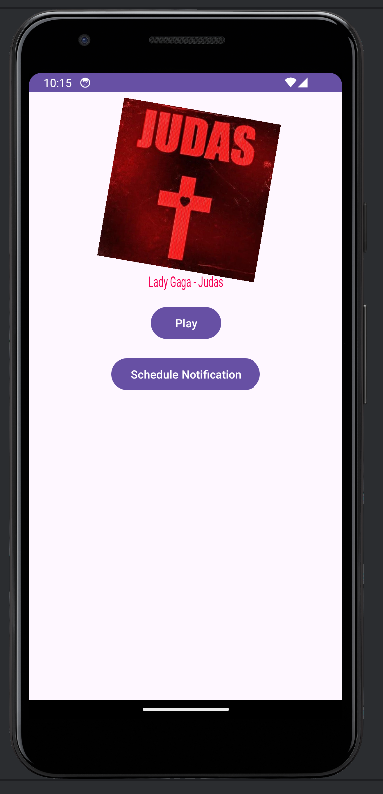


Рисунок 2. Проверка анимаций

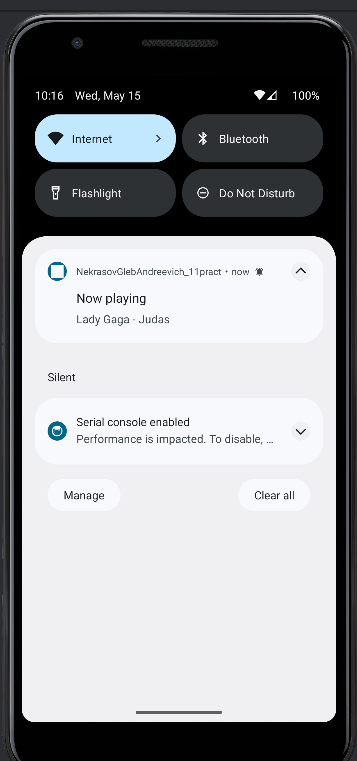


Рисунок 3. Проверка прямого уведомления

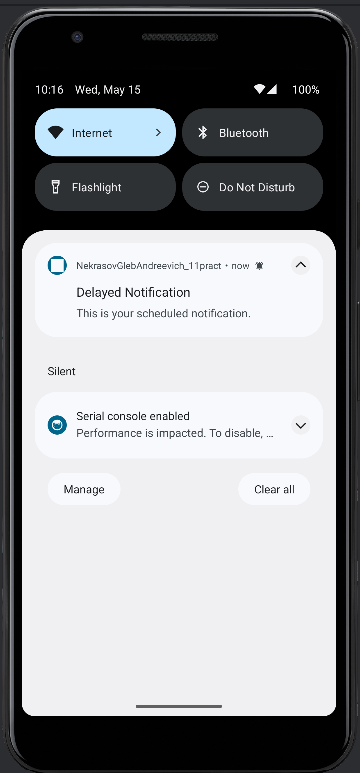


Рисунок 4. Проверка отложенного уведомления