**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 22**

по дисциплине «Разработка мобильных приложений»

**Тема: Unit 3 Работа в фоновом режиме “Уведомления”**

Студент группы РИЗ-300028у: А.С. Севостьянов

Преподаватель: С.И.Тимошенко,

доц., к.т.н.

**Екатеринбург 2022**

содержание

[1 Постановка задачи 3](#_Toc118041527)

[2 Результаты проделанной работы 4](#_Toc118041528)

[3 Анализ полученных результатов 5](#_Toc118041529)

## 1 Постановка задачи

Создать приложение, которое отправляет уведомление, когда пользователь нажимает кнопку в приложении.

Обновить уведомление с помощью кнопки в приложении и кнопки действия, которая находится внутри уведомления.

.

## Результаты проделанной работы

Программный код проекта, файл MainActivity.

package com.example.a81;  
  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.NotificationCompat;  
  
*/\*\*  
 Содержит три кнопки, которые доставляют,  
 обновляют и отменяют уведомление.  
 \*/*public class MainActivity extends AppCompatActivity {  
  
 // Константы для кнопок действий с уведомлениями. private static final String *ACTION\_UPDATE\_NOTIFICATION* =  
 "com.android.example.notifyme.ACTION\_UPDATE\_NOTIFICATION";  
 // Идентификатор канала уведомления.  
 private static final String *PRIMARY\_CHANNEL\_ID* =  
 "primary\_notification\_channel";  
 // Идентификатор уведомления.  
 private static final int *NOTIFICATION\_ID* = 0;  
  
 private Button button\_notify;  
 private Button button\_cancel;  
 private Button button\_update;  
  
 private NotificationManager mNotifyManager;  
  
 private NotificationReceiver mReceiver = new NotificationReceiver();  
  
  
 @Override  
 protected void onCreate(Bundle ) {  
 super.onCreate(savedInstanceState);  
 setContentViewsetContentView(R.layout.*activity\_main*);  
  
 // Создайте канал уведомлений.  
 createNotificationChannel();  
  
// Зарегистрируйте широковещательный приемник для получения действия обновления от  
// уведомление.  
 registerReceiver(mReceiver,  
 new IntentFilter(*ACTION\_UPDATE\_NOTIFICATION*));  
  
 // Добавьте обработчики onClick ко всем кнопкам.  
 button\_notify = findViewById(R.id.*notify*);  
 button\_notify.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 // Отправить уведомление  
 sendNotification();  
 }  
 });  
  
 button\_update = (Button) findViewById(R.id.*update*);  
 button\_update.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 // Обновите уведомление.  
 updateNotification ();  
 }  
 });  
  
 button\_cancel = (Button) findViewById(R.id.*cancel*);  
 button\_cancel.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 // Отменить уведомление.  
 cancelNotification();  
 }  
 });  
  
// Сбросьте состояние кнопки. Включить только кнопку уведомления и отключить  
// кнопки обновления и отмены.  
 setNotificationButtonState(true, false, false);  
 }  
  
 */\*\*  
 Отмените регистрацию получателя, когда приложение будет уничтожено.  
 \*/* @Override  
 protected void onDestroy() {  
 unregisterReceiver(mReceiver);  
 super.onDestroy();  
 }  
  
 */\*\*  
 Создает канал уведомлений для OREO и выше.  
 \*/* public void createNotificationChannel() {  
  
// Создайте объект диспетчера уведомлений.  
 mNotifyManager =  
 (NotificationManager) getSystemService(*NOTIFICATION\_SERVICE*);  
  
// Каналы уведомлений доступны только в OREO и выше.  
// Итак, добавьте проверку версии SDK.  
 if (android.os.Build.VERSION.*SDK\_INT* >=  
 android.os.Build.VERSION\_CODES.*O*) {  
  
// Создайте канал уведомлений со всеми параметрами.  
 NotificationChannel notificationChannel = new NotificationChannel  
 (*PRIMARY\_CHANNEL\_ID*,  
 getString(R.string.notification\_channel\_name),  
 NotificationManager.*IMPORTANCE\_HIGH*);  
  
 notificationChannel.enableLights(true);  
 notificationChannel.setLightColor(Color.*RED*);  
 notificationChannel.enableVibration(true);  
 notificationChannel.setDescription  
 (getString(R.string.notification\_channel\_description));  
  
 mNotifyManager.createNotificationChannel(notificationChannel);  
 }  
 }  
  
 */\*\*  
 \* Метод OnClick для кнопки "Уведомить меня!".  
 \* Создает и доставляет простое уведомление.  
 \*/* public void sendNotification() {  
  
// Устанавливает ожидающее намерение обновить уведомление.  
// Соответствует нажатию кнопки Обновить меня!.  
 Intent updateIntent = new Intent(*ACTION\_UPDATE\_NOTIFICATION*);  
 PendingIntent updatePendingIntent = PendingIntent.*getBroadcast*(this,  
 *NOTIFICATION\_ID*, updateIntent, PendingIntent.*FLAG\_ONE\_SHOT*);  
  
// Создайте уведомление со всеми параметрами с помощью помощника  
// способ.  
 NotificationCompat.Builder notifyBuilder = getNotificationBuilder();  
  
// Добавьте кнопку действия, используя ожидающее намерение.  
 notifyBuilder.addAction(R.drawable.ic\_update,  
 getString(R.string.update), updatePendingIntent);  
  
// Доставить уведомление.  
 mNotifyManager.notify(*NOTIFICATION\_ID*, notifyBuilder.build());  
  
// Включает кнопки обновления и отмены, но отключает "Уведомлять  
// Я!" кнопка.  
 setNotificationButtonState(false, true, true);  
 }  
  
 */\*\*  
 \* Вспомогательный метод, который создает уведомление.  
 \*/* private NotificationCompat.Builder getNotificationBuilder() {  
  
// Настройте ожидающее намерение, которое доставляется при получении уведомления  
// нажата.  
 Intent notificationIntent = new Intent (this, MainActivity );  
 PendingIntent notificationPendingIntent = PendingIntent.*getActivity* (this, *NOTIFICATION\_ID*, notificationIntent,  
 PendingIntent.*FLAG\_UPDATE\_CURRENT*);  
  
// Создайте уведомление со всеми параметрами.  
 NotificationCompat.Builder notifyBuilder = new NotificationCompat  
 .Builder(this, *PRIMARY\_CHANNEL\_ID*)  
 .setContentTitle(getString(R.string.notification\_title))  
 .setContentText(getString(R.string.notification\_text))  
 .setSmallIcon(R.drawable.ic\_android)  
 .setAutoCancel(true).setContentIntent(notificationPendingIntent)  
 .setPriority(NotificationCompat.*PRIORITY\_HIGH*)  
 .setDefaults(NotificationCompat.*DEFAULT\_ALL*);  
 return notifyBuilder;  
 }  
  
 */\*\*  
 \* Метод OnClick для кнопки "Обновить меня!". Обновляет существующий  
 \* уведомление о показе картинки.  
 \*/* public void updateNotification() {  
  
// Загрузите доступный для рисования ресурс в растровое изображение.  
 Bitmap androidImage = BitmapFactory  
 .*decodeResource*(getResources(), R.drawable.mascot\_1);  
  
// Создайте уведомление со всеми параметрами с помощью помощника  
// способ.  
 NotificationCompat.Builder notifyBuilder = getNotificationBuilder();  
  
// Обновите стиль уведомления до BigPictureStyle.  
 notifyBuilder.setStyle(new NotificationCompat.BigPictureStyle()  
 .bigPicture(androidImage)  
 .setBigContentTitle(getString(R.string.notification\_updated)));  
  
// Доставить уведомление.  
 mNotifyManager.notify(*NOTIFICATION\_ID*, notifyBuilder.build());  
  
// Отключите кнопку обновления, оставив включенной только кнопку отмены.  
 setNotificationButtonState(false, false, true);  
 }  
  
 */\*\*  
 \* Метод OnClick для кнопки "Отменить меня!". Отменяет уведомление.  
 \*/* public void cancelNotification () {  
// Отменить уведомление.  
 mNotifyManager.cancel (*NOTIFICATION\_ID*);  
  
// Сбросьте кнопки.  
 setNotificationButtonState(true, false, false);  
 }  
  
 */\*\*  
 \* Вспомогательный метод для включения / выключения кнопок.  
 \*/* void setNotificationButtonState(Boolean isNotifyEnabled, Boolean  
 isUpdateEnabled, Boolean isCancelEnabled) {  
 button\_notify.setEnabled(isNotifyEnabled);  
 button\_update.setEnabled(isUpdateEnabled);  
 button\_cancel.setEnabled(isCancelEnabled);  
 }  
  
 */\*\*  
 \* Класс широковещательного приемника для уведомлений.  
 \* Отвечает на уведомление об обновлении в ожидании действия намерения.  
 \*/* public class NotificationReceiver NotificationReceiverextends BroadcastReceiver {  
  
 public NotificationReceiver() {  
 }  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 // Обновите уведомление. updateNotificationupdateNotification ();  
 }  
 }  
}

Отображение проекта.

<?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">  
  
 <Button  
 android:id="@+id/notify"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/notify\_me"  
 app:layout\_constraintBottom\_toTopOf="@+id/update"  
 app:layout\_constraintEnd\_toEndOf ="parent"  
 app:layout\_constraintStart\_toStartOf ="parent"  
 app:layout\_constraintTop\_toTopOf ="parent" />  
  
 <Button  
 android:id="@+id/update"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/update\_me"  
 app:layout\_constraintBottom\_toTopOf="@+id/cancel"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/notify" />  
  
 <Button  
 android:id= "@+id/cancel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/cancel\_me"  
 app:layout\_constraintBottom\_toBottomOf ="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf ="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/update" />  
</androidx.constraintlayout.widget.ConstraintLayout>

На рисунке 1 показано начальное состояние программы.

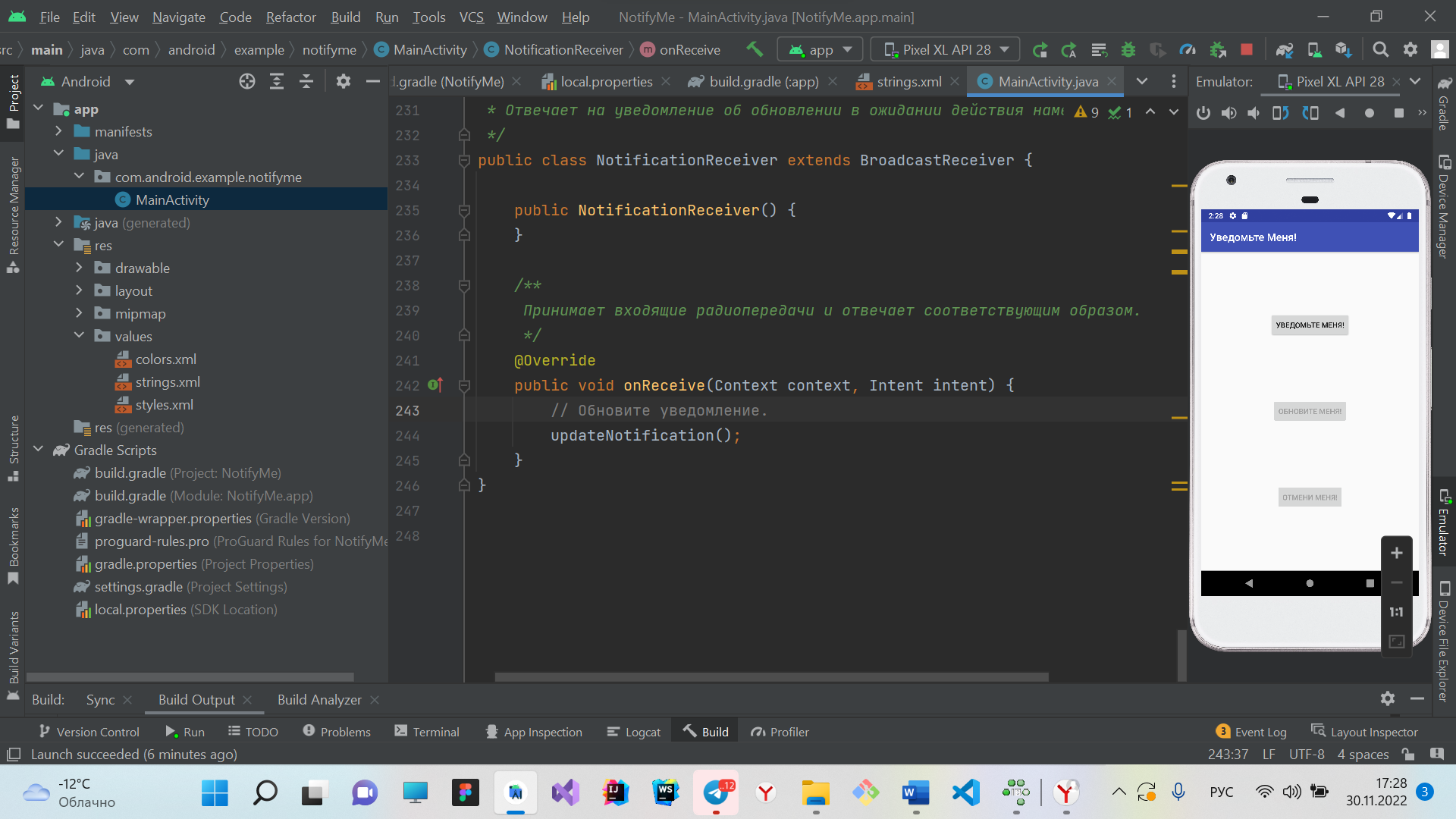


Рисунок 1 – Начальное состояние

На рисунке 2 показано состояние программы после того, как нажали на кнопку уведомить.

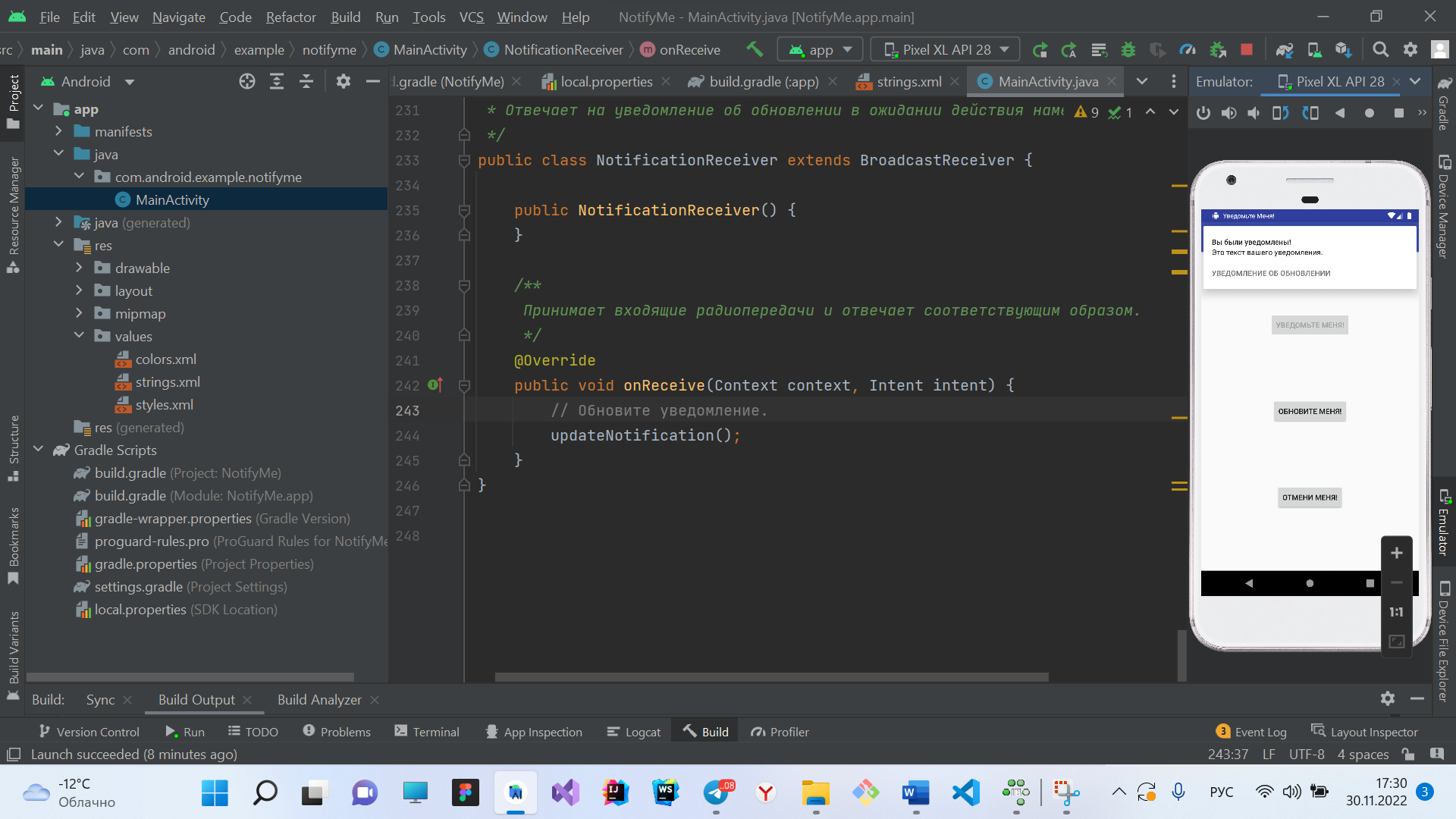


Рисунок 2 - Уведомление

## **Анализ полученных результатов**

В результате выполнения данной лабораторной работы мной был разработан проект, в котором я научился создавать уведомление с помощью конструктора уведомлений. Как использовать отложенные намерения для ответа на действия с уведомлениями. Как обновить или отменить существующие уведомления.

.