**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 23**

по дисциплине «Разработка мобильных приложений»

**Тема: Unit 3 Работа в фоновом режиме “Диспетчер аварийных сигналов”**

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

Установить повторяющийся будильник, чтобы уведомлять вас каждые 15 минут. Используйте ToggleButtonдля установки и отслеживания будильника. Используйте Toastсообщения для уведомления пользователя о включении или выключении будильника.

## Результаты проделанной работы

Программный код, файл MainActivity.

package com.android.fundamentals.standup;  
  
import android.app.AlarmManager;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Intent;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.os.SystemClock;  
import android.support.v7.app.AppCompatActivity;  
import android.widget.CompoundButton;  
import android.widget.Toast;  
import android.widget.ToggleButton;  
  
  
*/\*\*  
 Основное занятие для приложения Stand up!. Содержит кнопку переключения, которая  
 устанавливает будильник, который подает уведомление о вставании каждые 15 минут.  
 \*/*public class MainActivity extends AppCompatActivity {  
  
 // Идентификатор уведомления.  
 private static final int *NOTIFICATION\_ID* = 0;  
 // Идентификатор канала уведомления.  
 private static final String *PRIMARY\_CHANNEL\_ID* =  
 "primary\_notification\_channel";  
 private NotificationManager mNotificationManager;  
  
 */\*\*  
 Инициализирует действие.  
 \*/* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mNotificationManager = (NotificationManager)  
 getSystemService(*NOTIFICATION\_SERVICE*);  
  
 ToggleButton alarmToggle = findViewById(R.id.*alarmToggle*);  
  
// Настройте намерение широковещательной рассылки уведомлений.  
 Intent notifyIntent = new Intent(this, AlarmReceiver.class);  
  
 boolean alarmUp = (PendingIntent.*getBroadcast*(this, *NOTIFICATION\_ID*,  
 notifyIntent, PendingIntent.*FLAG\_NO\_CREATE*) != null);  
 alarmToggle.setChecked(alarmUp);  
  
 final PendingIntent notifyPendingIntent = PendingIntent.*getBroadcast* (this, *NOTIFICATION\_ID*, notifyIntent,  
 PendingIntent.*FLAG\_UPDATE\_CURRENT*);  
  
 final AlarmManager alarmManager = (AlarmManager) getSystemService  
 (*ALARM\_SERVICE*);  
  
 // Установите прослушиватель щелчков для кнопки переключения.  
 alarmToggle.setOnCheckedChangeListener  
 (new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged  
 (CompoundButton buttonView, boolean isChecked) {  
 String toastMessage;  
 if (isChecked) {  
  
 long repeatInterval = AlarmManager.*INTERVAL\_FIFTEEN\_MINUTES*;  
  
 long triggerTime = SystemClock.*elapsedRealtime*()  
 + repeatInterval;  
  
// Если переключатель включен, установите повторяющийся сигнал тревоги с  
 // 15-минутный интервал.  
 if (alarmManager != null) {  
 alarmManager.setInexactRepeating  
 (AlarmManager.*ELAPSED\_REALTIME\_WAKEUP*,  
 triggerTime, repeatInterval,  
 notifyPendingIntent);  
 }  
// Установите всплывающее сообщение для случая "включено".  
 toastMessage = getString(R.string.*alarm\_on\_toast*);  
  
 } else {  
// Отменить уведомление, если будильник выключен.  
 mNotificationManager.cancelAll();  
  
 if (alarmManager != null) {  
 alarmManager.cancel(notifyPendingIntent);  
 }  
// Установите всплывающее сообщение для случая "выкл.".  
 toastMessage = getString(R.string.*alarm\_off\_toast*);  
  
 }  
  
 // Покажите тост, чтобы сообщить, что будильник включен или выключен.  
 Toast.*makeText*(MainActivity.this, toastMessage,  
 Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 // Создайте канал уведомлений.  
 createNotificationChannel();  
 }  
  
  
 */\*\*  
 \* Создает канал уведомлений для OREO и выше.  
 \*/* public void createNotificationChannel() {  
  
// Создайте объект диспетчера уведомлений.  
 mNotificationManager =  
 (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*,  
 "Stand up notification",  
 NotificationManager.*IMPORTANCE\_HIGH*);  
  
 notificationChannel.enableLights(true);  
 notificationChannel.setLightColor(Color.*RED*);  
 notificationChannel.enableVibration(true);  
 notificationChannel.setDescription("Notifies every 15 minutes to " +  
 "stand up and walk");  
 mNotificationManager.createNotificationChannel(notificationChannel);  
 }  
 }  
}

Программный код, файл AlarmReceiver

package com.android.fundamentals.standup;  
  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.support.v4.app.NotificationCompat;  
  
*/\*\*  
 \* Широковещательный приемник сигнала тревоги, который доставляет уведомление.  
 \*/*public class AlarmReceiver extends BroadcastReceiver {  
  
 private NotificationManager mNotificationManager;  
 // Идентификатор уведомления.  
 private static final int *NOTIFICATION\_ID* = 0;  
 // Идентификатор канала уведомления.  
 private static final String *PRIMARY\_CHANNEL\_ID* =  
 "primary\_notification\_channel";  
  
 */\*\*  
 \* Вызывается, когда BroadcastReceiver получает широковещательную передачу намерения.  
 \*/* @Override  
 public void onReceive(Context context, Intent intent) {  
 mNotificationManager = (NotificationManager)  
 context.getSystemService(Context.*NOTIFICATION\_SERVICE*);  
  
 // Доставить уведомление.  
 deliverNotification(context);  
 }  
  
 */\*\*  
 \* Создает и доставляет уведомление.  
 \*/* private void deliverNotification(Context context) {  
// Создайте намерение содержимого для уведомления, которое запускает  
 // эта деятельность  
 Intent contentIntent = new Intent(context, MainActivity.class);  
  
 PendingIntent contentPendingIntent = PendingIntent.*getActivity* (context, *NOTIFICATION\_ID*, contentIntent, PendingIntent  
 .*FLAG\_UPDATE\_CURRENT*);  
// Создайте уведомление  
 NotificationCompat.Builder builder = new NotificationCompat.Builder  
 (context, *PRIMARY\_CHANNEL\_ID*)  
 .setSmallIcon(R.drawable.*ic\_stand\_up*)  
 .setContentTitle(context.getString(R.string.*notification\_title*))  
 .setContentText(context.getString(R.string.*notification\_text*))  
 .setContentIntent(contentPendingIntent)  
 .setPriority(NotificationCompat.*PRIORITY\_HIGH*)  
 .setAutoCancel(true)  
 .setDefaults(NotificationCompat.*DEFAULT\_ALL*);  
  
// Доставить уведомление  
 mNotificationManager.notify(*NOTIFICATION\_ID*, builder.build());  
 }  
}

На рисунке 1 показано начальное состояние программы.

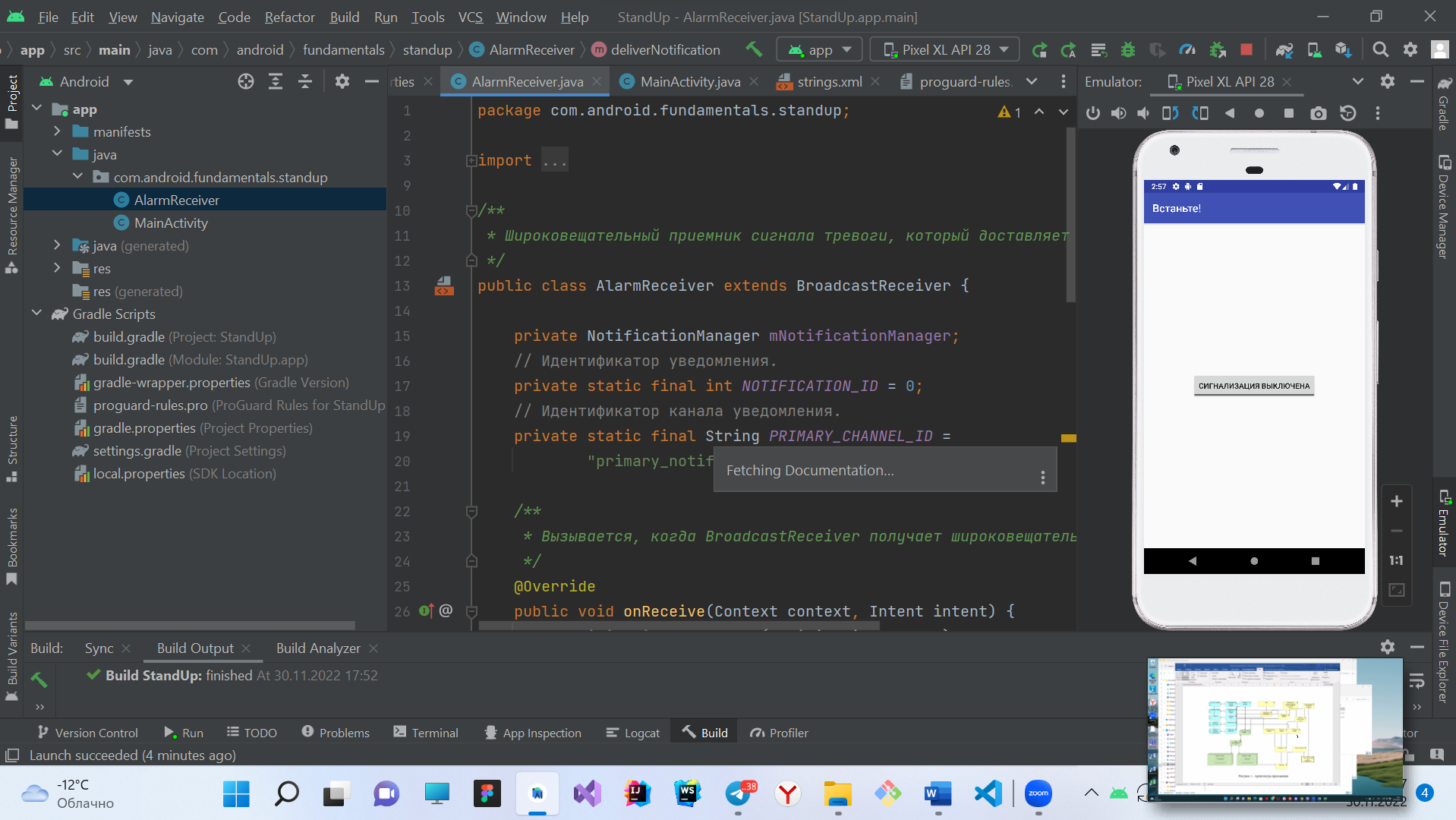


Рисунок 1 – Изначальное состояние программы

На рисунке 2 показано состояние программы после нажатия на кнопку.

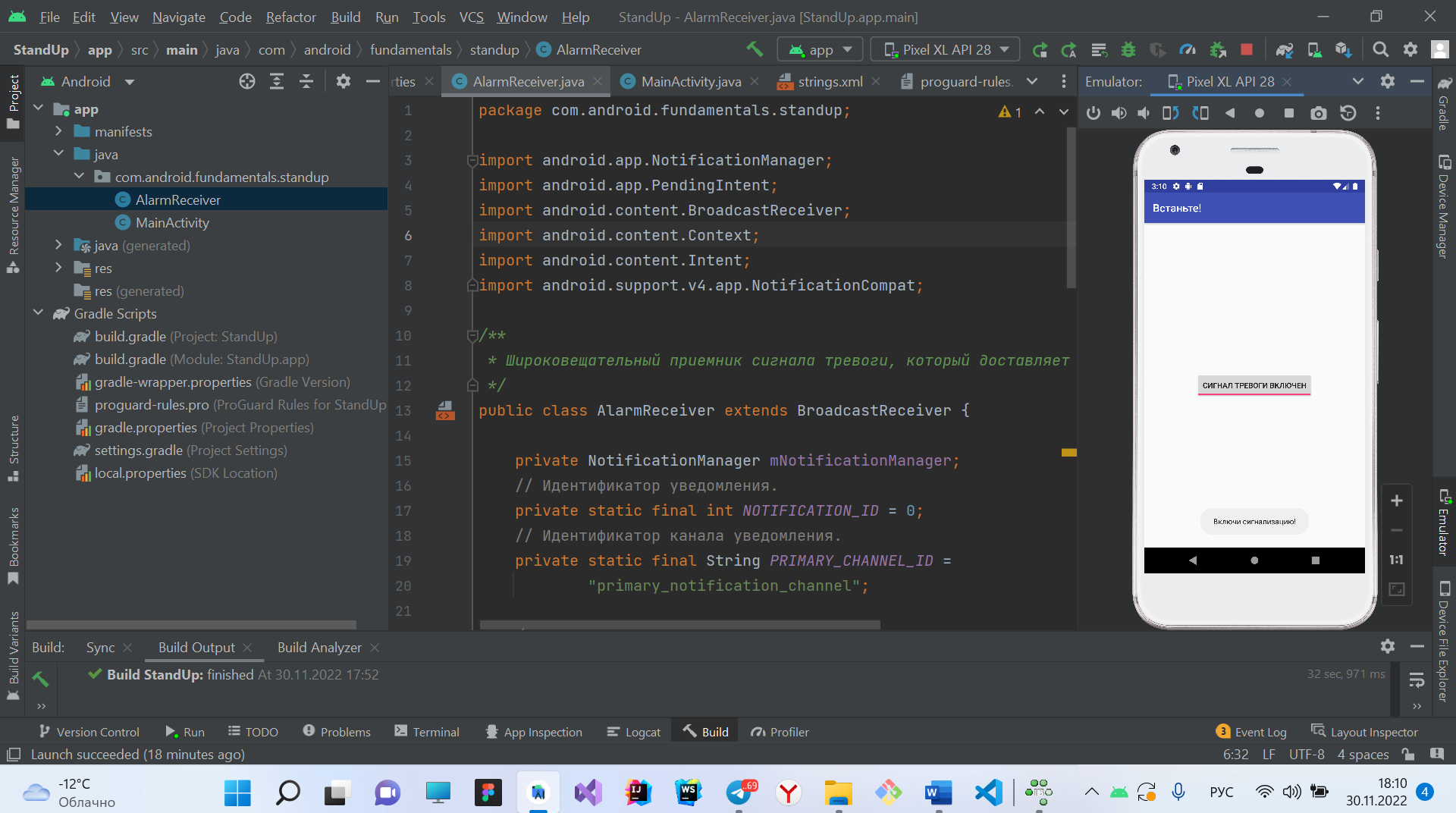


Рисунок 2 – Результат работы программы

## **Анализ полученных результатов**

В результате выполнения данной лабораторной работы мной был разработан проект, в котором я узнал, как запланировать повторяющиеся сигналы тревоги c AlarmManager помощью. Как проверить, установлен ли уже будильник. Как отменить повторяющийся сигнал тревоги.

.