|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** - **Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1** | |
| **по дисциплине** | |
| «Разработка программных приложений» | |
|  | |
| Выполнил студент группы ИКБО-02-18 | Юркевич Г.А. |
| Принял ассистент кафедры ИППО | Строганкова Н.В. |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «\_\_»\_\_\_\_\_\_\_2020 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2020 г. |  |

Москва 2020

**Содержание**

[1. Цель лабораторной работы 3](#_heading=h.26in1rg)

2. Общие требования [3](#_heading=h.2et92p0)

3. Задание [3](#_heading=h.3dy6vkm)

4. Поэтапное выполнение лабораторной работы [4](#_heading=h.4d34og8)

5. Результат работы [4](#_heading=h.lnxbz9)

6.Вывод [4](#_heading=h.17dp8vu)

[Список использованных источников 9](#_heading=h.35nkun2)

1. Цель лабораторной работы

Целью данной лабораторной работы является ознакомление со средой разработки программных приложений Android Studio, а также создание приложения моделирующего e-commerce решение.

**2. Общие требования**

Среда разработки по умолчанию - Android Studio. Язык программирования - Java.

По своему желанию, можно использовать Kotlin, а также XCode/Swift при наличии соответствующего оборудования.

Решения должны быть размещены в виде проекта в git репозитории (github/bitbucket).

Ссылку на git-репозиторий и отчет по выполненной работе прислать на почту: strogankova-mirea@yandex.ru.

Минимальная версия API – 19/

**3. Задание**

Реализовать widget, показывающий количество дней до установленной даты. По нажатию на сам виджет, отображать диалог для выбора даты. В указанный день, в 9:00, показывать уведомление о том что событие наступило.

**4. Поэтапное выполнение лабораторной работы**

Вначале был создан виджет, после чего был создан диалог для выбора даты, а также было налажено взаимодействие между ними. Для виджета был переписан метод onUpgrade в соответствии с требованиями задания. Также в этом методе было создано уведомление, которое появляется в 9 часов назначенной через диалог даты.

**5. Результат работы**

**MyWidget**

package com.example.clickwidget;

import java.text.SimpleDateFormat;

import java.time.LocalDate;

import java.time.Period;

import java.time.format.DateTimeFormatter;

import java.util.Calendar;

import java.util.Date;

import java.util.Locale;

import android.app.Notification;

import android.app.NotificationChannel;

import android.app.NotificationChannelGroup;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.appwidget.AppWidgetManager;

import android.appwidget.AppWidgetProvider;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.SharedPreferences.Editor;

import android.graphics.Color;

import android.os.Bundle;

import android.util.Log;

import android.widget.RemoteViews;

import androidx.core.app.NotificationCompat;

import static android.content.Context.NOTIFICATION\_SERVICE;

public class MyWidget extends AppWidgetProvider {

final String LOG\_TAG = "myLogs";

public SharedPreferences sp;

final static String ACTION\_CHANGE = "com.example.clickwidget.change\_count";

private NotificationManager mManager;

public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {

super.onUpdate(context, appWidgetManager, appWidgetIds);

Log.d(LOG\_TAG, "onUpdate " );

// обновляем все экземпляры

for (int i : appWidgetIds) {

Log.d(LOG\_TAG, "onUpdate start " );

updateWidget(context, appWidgetManager, i);

}

}

public void onDeleted(Context context, int[] appWidgetIds) {

super.onDeleted(context, appWidgetIds);

// Удаляем Preferences

Editor editor = context.getSharedPreferences(

ConfigActivity.WIDGET\_PREF, Context.MODE\_PRIVATE).edit();

for (int widgetID : appWidgetIds) {

editor.remove(ConfigActivity.WIDGET\_TIME + widgetID);

editor.remove(ConfigActivity.WIDGET\_COUNT + widgetID);

}

editor.commit();

}

static void updateWidget(Context ctx, AppWidgetManager appWidgetManager, int widgetID) {

SharedPreferences sp = ctx.getSharedPreferences(

ConfigActivity.WIDGET\_PREF, Context.MODE\_PRIVATE);

// Читаем формат времени и определяем текущее время

String time = sp.getString(ConfigActivity.WIDGET\_TIME + widgetID, null);

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd.MM.yyyy");

String currentDate = new SimpleDateFormat("dd.MM.yyyy", Locale.getDefault()).format(new Date());

String currentTime = new SimpleDateFormat("HH", Locale.getDefault()).format(new Date());

LocalDate startDate = LocalDate.parse(time, formatter);

LocalDate endDate = LocalDate.parse(currentDate, formatter);

Period period = Period.between(endDate,startDate );

if (time == null) return;

Log.d("myLogs", currentTime );

if ((currentTime).equals("09")&& period.getDays()==0 && period.getMonths()==0 && period.getYears()==0 ) {

Log.d("myLogs", "start " );

NotificationChannel androidChannel = new NotificationChannel("1",

"chanel 6", NotificationManager.IMPORTANCE\_DEFAULT);

// Sets whether notifications posted to this channel should display notification lights

androidChannel.enableLights(true);

// Sets whether notification posted to this channel should vibrate.

androidChannel.enableVibration(true);

// Sets the notification light color for notifications posted to this channel

androidChannel.setLightColor(Color.GREEN);

// Sets whether notifications posted to this channel appear on the lockscreen or not

androidChannel.setLockscreenVisibility(Notification.VISIBILITY\_PRIVATE);

NotificationCompat.Builder builder =

new NotificationCompat.Builder(ctx, "1")

.setSmallIcon(R.mipmap.ic\_launcher)

.setContentTitle("Время!")

.setContentText("Я ждал этого часа 10 000 лет!");

Notification notification = builder.build();

NotificationManager notificationManager =

(NotificationManager) ctx.getSystemService(NOTIFICATION\_SERVICE);

notificationManager.notify(1, notification);

Log.d("myLogs", "end" );

}

// Помещаем данные в текстовые поля

RemoteViews widgetView = new RemoteViews(ctx.getPackageName(),

R.layout.widget);

widgetView.setTextViewText(R.id.tvPressConfig, Integer.toString(period.getDays()));

// Конфигурационный экран (первая зона)

Intent configIntent = new Intent(ctx, ConfigActivity.class);

configIntent.setAction(AppWidgetManager.ACTION\_APPWIDGET\_CONFIGURE);

configIntent.putExtra(AppWidgetManager.EXTRA\_APPWIDGET\_ID, widgetID);

PendingIntent pIntent = PendingIntent.getActivity(ctx, widgetID,

configIntent, 0);

widgetView.setOnClickPendingIntent(R.id.tvPressConfig, pIntent);

// Обновление виджета (вторая зона)

Intent updateIntent = new Intent(ctx, MyWidget.class);

updateIntent.setAction(AppWidgetManager.ACTION\_APPWIDGET\_UPDATE);

updateIntent.putExtra(AppWidgetManager.EXTRA\_APPWIDGET\_IDS,

new int[] { widgetID });

pIntent = PendingIntent.getBroadcast(ctx, widgetID, updateIntent, 0);

widgetView.setOnClickPendingIntent(R.id.tvPressUpdate, pIntent);

// Обновляем виджет

appWidgetManager.updateAppWidget(widgetID, widgetView);

}

public void onReceive(Context context, Intent intent) {

super.onReceive(context, intent);

}

}

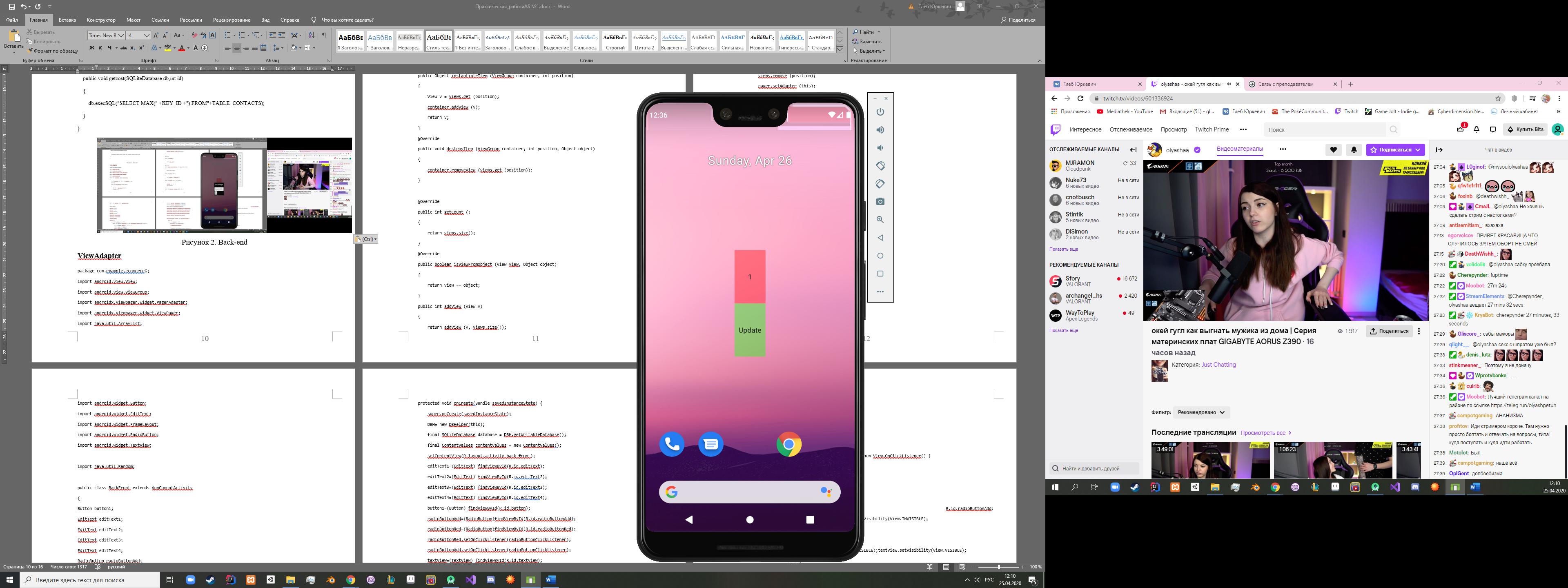


Рисунок 1. Виджет

**ConfigActivity**

package com.example.clickwidget;

import java.text.SimpleDateFormat;

import java.time.LocalDate;

import java.time.Period;

import java.time.format.DateTimeFormatter;

import java.util.Calendar;

import java.util.Date;

import java.util.Locale;

import android.app.Notification;

import android.app.NotificationChannel;

import android.app.NotificationChannelGroup;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.appwidget.AppWidgetManager;

import android.appwidget.AppWidgetProvider;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.SharedPreferences.Editor;

import android.graphics.Color;

import android.os.Bundle;

import android.util.Log;

import android.widget.RemoteViews;

import androidx.core.app.NotificationCompat;

import static android.content.Context.NOTIFICATION\_SERVICE;

public class MyWidget extends AppWidgetProvider {

final String LOG\_TAG = "myLogs";

public SharedPreferences sp;

final static String ACTION\_CHANGE = "com.example.clickwidget.change\_count";

private NotificationManager mManager;

public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {

super.onUpdate(context, appWidgetManager, appWidgetIds);

Log.d(LOG\_TAG, "onUpdate " );

// обновляем все экземпляры

for (int i : appWidgetIds) {

Log.d(LOG\_TAG, "onUpdate start " );

updateWidget(context, appWidgetManager, i);

}

}

public void onDeleted(Context context, int[] appWidgetIds) {

super.onDeleted(context, appWidgetIds);

// Удаляем Preferences

Editor editor = context.getSharedPreferences(

ConfigActivity.WIDGET\_PREF, Context.MODE\_PRIVATE).edit();

for (int widgetID : appWidgetIds) {

editor.remove(ConfigActivity.WIDGET\_TIME + widgetID);

editor.remove(ConfigActivity.WIDGET\_COUNT + widgetID);

}

editor.commit();

}

static void updateWidget(Context ctx, AppWidgetManager appWidgetManager, int widgetID) {

SharedPreferences sp = ctx.getSharedPreferences(

ConfigActivity.WIDGET\_PREF, Context.MODE\_PRIVATE);

// Читаем формат времени и определяем текущее время

String time = sp.getString(ConfigActivity.WIDGET\_TIME + widgetID, null);

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd.MM.yyyy");

String currentDate = new SimpleDateFormat("dd.MM.yyyy", Locale.getDefault()).format(new Date());

String currentTime = new SimpleDateFormat("HH", Locale.getDefault()).format(new Date());

LocalDate startDate = LocalDate.parse(time, formatter);

LocalDate endDate = LocalDate.parse(currentDate, formatter);

Period period = Period.between(endDate,startDate );

if (time == null) return;

Log.d("myLogs", currentTime );

if ((currentTime).equals("09")&& period.getDays()==0 && period.getMonths()==0 && period.getYears()==0 ) {

Log.d("myLogs", "start " );

NotificationChannel androidChannel = new NotificationChannel("1",

"chanel 6", NotificationManager.IMPORTANCE\_DEFAULT);

// Sets whether notifications posted to this channel should display notification lights

androidChannel.enableLights(true);

// Sets whether notification posted to this channel should vibrate.

androidChannel.enableVibration(true);

// Sets the notification light color for notifications posted to this channel

androidChannel.setLightColor(Color.GREEN);

// Sets whether notifications posted to this channel appear on the lockscreen or not

androidChannel.setLockscreenVisibility(Notification.VISIBILITY\_PRIVATE);

NotificationCompat.Builder builder =

new NotificationCompat.Builder(ctx, "1")

.setSmallIcon(R.mipmap.ic\_launcher)

.setContentTitle("Время!")

.setContentText("Я ждал этого часа 10 000 лет!");

Notification notification = builder.build();

NotificationManager notificationManager =

(NotificationManager) ctx.getSystemService(NOTIFICATION\_SERVICE);

notificationManager.notify(1, notification);

Log.d("myLogs", "end" );

}

// Помещаем данные в текстовые поля

RemoteViews widgetView = new RemoteViews(ctx.getPackageName(),

R.layout.widget);

widgetView.setTextViewText(R.id.tvPressConfig, Integer.toString(period.getDays()));

// Конфигурационный экран (первая зона)

Intent configIntent = new Intent(ctx, ConfigActivity.class);

configIntent.setAction(AppWidgetManager.ACTION\_APPWIDGET\_CONFIGURE);

configIntent.putExtra(AppWidgetManager.EXTRA\_APPWIDGET\_ID, widgetID);

PendingIntent pIntent = PendingIntent.getActivity(ctx, widgetID,

configIntent, 0);

widgetView.setOnClickPendingIntent(R.id.tvPressConfig, pIntent);

// Обновление виджета (вторая зона)

Intent updateIntent = new Intent(ctx, MyWidget.class);

updateIntent.setAction(AppWidgetManager.ACTION\_APPWIDGET\_UPDATE);

updateIntent.putExtra(AppWidgetManager.EXTRA\_APPWIDGET\_IDS,

new int[] { widgetID });

pIntent = PendingIntent.getBroadcast(ctx, widgetID, updateIntent, 0);

widgetView.setOnClickPendingIntent(R.id.tvPressUpdate, pIntent);

// Обновляем виджет

appWidgetManager.updateAppWidget(widgetID, widgetView);

}

public void onReceive(Context context, Intent intent) {

super.onReceive(context, intent);

}

}

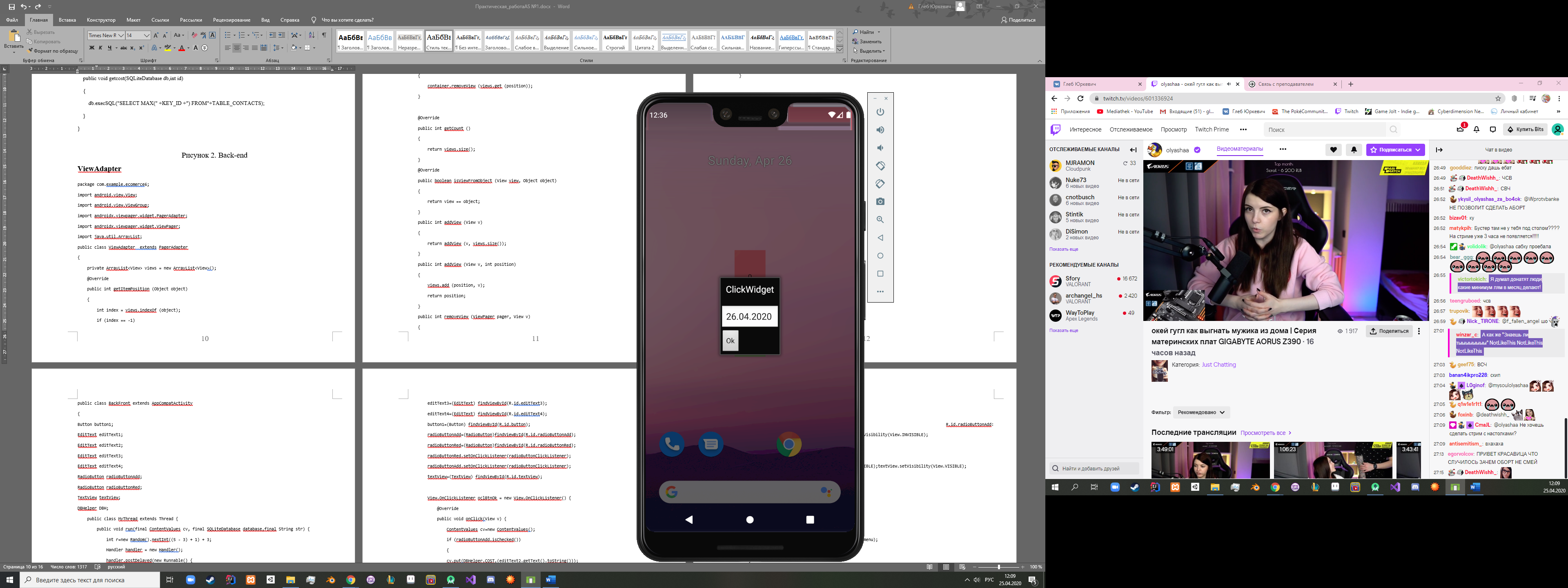


Рисунок 2. Config

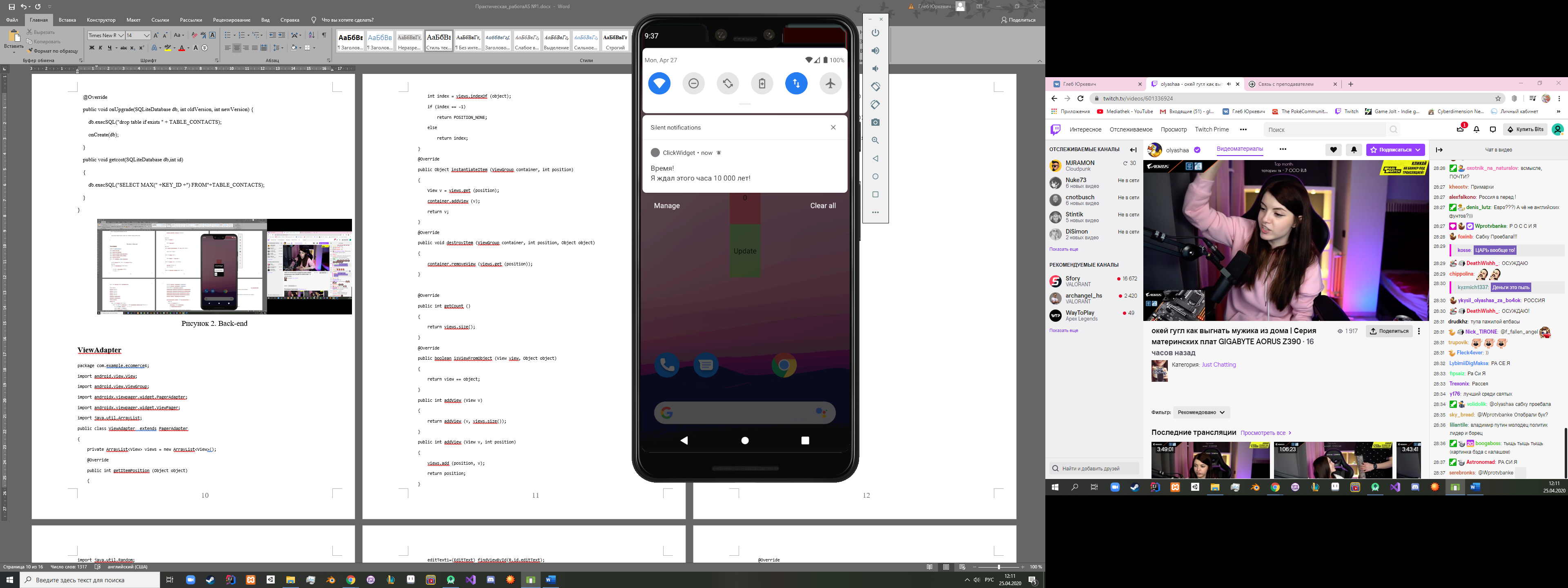


Рисунок 3. Уведомление

**5. Вывод**

В данной работе было разработан widget показывающий количество дней до установленной даты.

**Список использованных источников**

1. Главный сайт по Android: <https://developer.android.com/>
2. Канал Google на YouTube:

<http://www.youtube.com/channel/UC_x5XG1OV2P6uZZ5FSM9Ttw>

1. Онлайн-курсы по Android:

– Udacity:

<https://eu.udacity.com/course/new-android-fundamentals--ud851>

1. <https://stackoverflow.com/>
2. <https://github.com>
3. <https://google.com/>