МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  
 «Кемеровский государственный университет»**

**Институт фундаментальных наук**

**Кафедра ЮНЕСКО по ИВТ**

**ЛАБОРАТОРНАЯ РАБОТА №5**

**ПО ДИСЦИПЛИНЕ “РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ”**

студента 2 курса

**Крючкова Андрея Олеговича**

Направление 09.03.03 – Прикладная информатика в экономике

Преподаватель:

доцент

С.Ю. Завозкин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена:

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_202\_г.

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Кемерово 202\_

СОДЕРЖАНИЕ

[1. Постановка задач 2](#_Toc86597544)

[2. Реализация задач 3](#_Toc86597545)

[Заключение 9](#_Toc86597546)

# 1. Постановка задач

1. Познакомиться с особенностями работы библиотеки GSON
2. Познакомиться с процессом динамического обновления данных в RecyclerView.

# 2. Реализация задач

**1 Задание**

Создать приложение, содержащее в себе динамически наполняемый RecyclerView данными, полученными из Интернета с использованием библиотеки GSON;

1. Добавлен в Activity\_main.xml объект recyclerView с id “rView”

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/rView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
</LinearLayout>

Код Activity\_main.xml

1. Создана новая схема компановки для объектов recyclerView

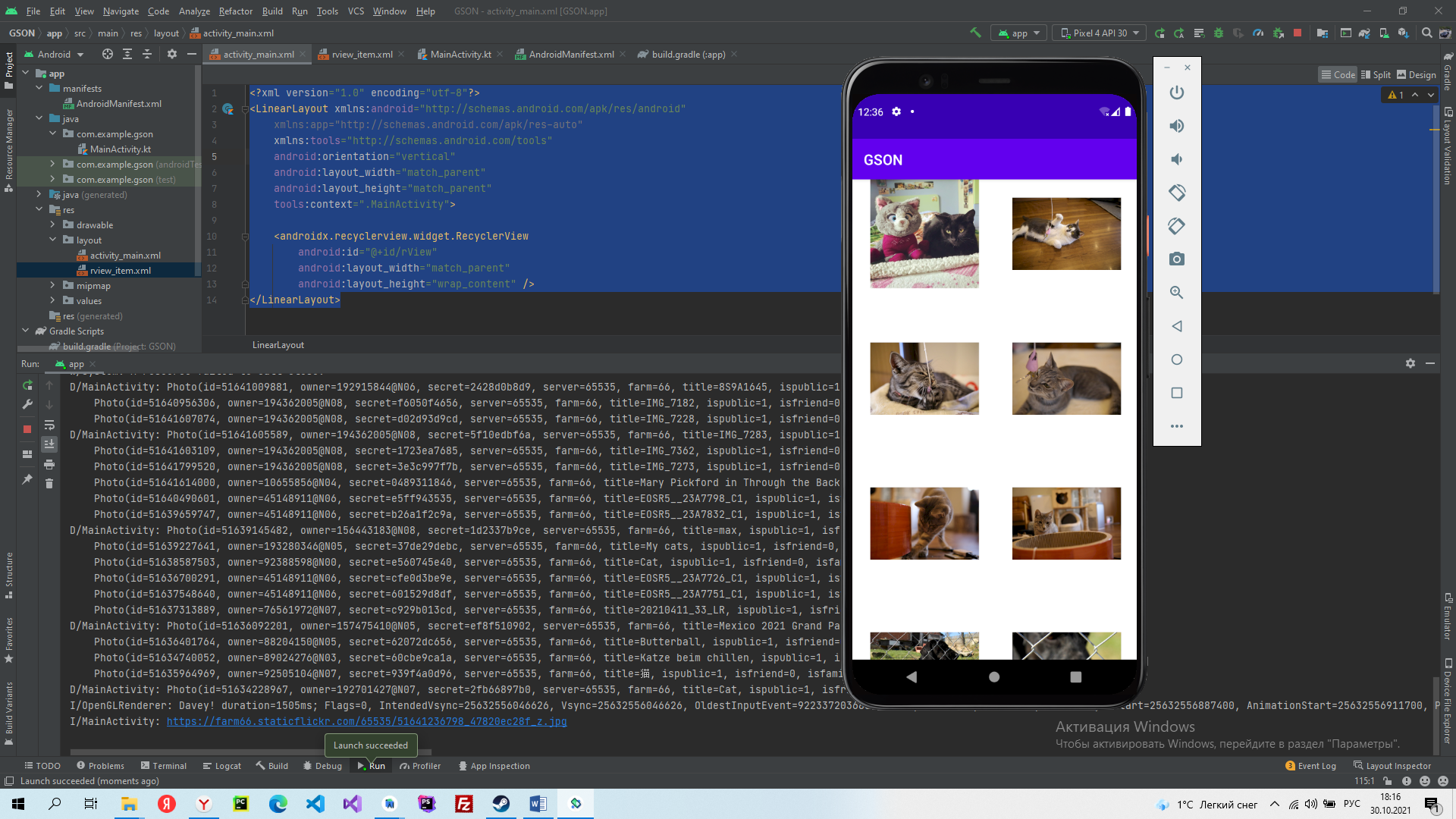
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/rview\_item"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 android:layout\_marginLeft="25dp"  
 android:layout\_marginRight="25dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
</LinearLayout>

Код rview\_item.xml

1. Реализованы: класс Адаптер, наполнение recyclerView объектами, копирование ссылки в буфер обмена и её вывод в логи при нажатии на элемент списка фото.

package com.example.gson  
  
import android.content.ClipData  
import android.content.ClipboardManager  
import android.content.Context  
import android.graphics.BitmapFactory  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.os.StrictMode  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ImageView  
import androidx.recyclerview.widget.GridLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import com.google.gson.Gson  
import com.google.gson.JsonArray  
import com.google.gson.JsonObject  
import okhttp3.\*  
import timber.log.Timber  
import java.io.InputStream  
import java.lang.Exception  
import java.net.HttpURLConnection  
import java.net.URL  
  
  
class MainActivity : AppCompatActivity() {  
  
 val url = "https://api.flickr.com/services/rest/?method=flickr.photos.search&api\_key=ff49fcd4d4a08aa6aafb6ea3de826464&tags=cat&format=json&nojsoncallback=1"  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 val policy = StrictMode.ThreadPolicy.Builder().permitAll().build()  
 StrictMode.setThreadPolicy(policy)  
 setContentView(R.layout.*activity\_main*)  
  
 val recyclerView: RecyclerView = findViewById(R.id.*rView*)  
 val gridLayoutManager = GridLayoutManager(this,2)  
  
 Timber.plant(Timber.DebugTree())  
 val listPhoto = *arrayListOf*<String>()  
  
 Thread **{** val connection = URL(url).openConnection() as HttpURLConnection  
 try {  
 val data = connection.*inputStream*.*bufferedReader*().*readText*()  
  
 val wrapper: Wrapper = Gson().fromJson(data, Wrapper::class.*java*)  
 val page: PhotoPage = Gson().fromJson(wrapper.photos, PhotoPage::class.*java*)  
 val photo = Gson().fromJson(page.photo, Array<Photo>::class.*java*).*toList*()  
  
 for (i in photo.*indices*) {  
 if (i % 5 == 4) {  
 Timber.d(photo[i].toString())  
  
 }  
 listPhoto.add("https://farm${photo[i].farm}.staticflickr.com/${photo[i].server}/${photo[i].id}\_${photo[i].secret}\_z.jpg")  
  
 }  
  
 runOnUiThread() **{** recyclerView.setHasFixedSize(true)  
 recyclerView.*layoutManager* = gridLayoutManager  
 recyclerView.*adapter* = Adapter(this, listPhoto, this)  
 **}** } catch (e: Exception) {  
 Timber.e(e.toString())  
 } finally {  
 connection.disconnect()  
 }  
  
  
 **}**.start()  
  
  
  
  
 }  
  
  
  
 fun onCellClickListener(data: String){  
 val clipboardmanager: ClipboardManager = getSystemService(*CLIPBOARD\_SERVICE*) as ClipboardManager  
 val clipBoard = ClipData.newPlainText("Link", data)  
 clipboardmanager.setPrimaryClip(clipBoard)  
 Timber.i(data)  
  
  
 }  
}  
  
data class Photo(  
 val id: String,  
 val owner: String,  
 val secret: String,  
 val server: String,  
 val farm: Int,  
 val title: String,  
 val ispublic: Int,  
 val isfriend: Int,  
 val isfamily: Int  
)  
  
data class PhotoPage(  
 val page: Int,  
 val pages: Int,  
 val perpages: Int,  
 val total: Int,  
 val photo: JsonArray  
  
)  
data class Wrapper(  
 val photos: JsonObject,  
 val stat: String  
)  
  
  
class Adapter(  
 private val context: Context,  
 private val list: ArrayList<String>,  
 private val cellClickListener: MainActivity  
): RecyclerView.Adapter<Adapter.ViewHolder>(){  
  
 class ViewHolder(view: View): RecyclerView.ViewHolder(view){  
 val image: ImageView = view.findViewById(R.id.*imageView*)  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
 val view = LayoutInflater.from(context).inflate(R.layout.*rview\_item*, parent, false)  
 return ViewHolder(view)  
 }  
  
 override fun getItemCount(): Int {  
 return list.*count*()  
 }  
  
 override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
 val rViewList = list[position]  
 val inptStream: InputStream = URL(rViewList).openStream()  
 val bmp = BitmapFactory.decodeStream(inptStream)  
 holder.image.setImageBitmap(bmp)  
  
 holder.itemView.setOnClickListener**{** cellClickListener.onCellClickListener(rViewList)  
 **}** }  
  
}

Код MainActivity.kt

4) 

Итоговое приложение

**2 Задание**

Реализовать приложение, содержащее в себе динамически обновляемый RecyclerView.

1. Создан основной макет приложения

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <EditText  
 android:id="@+id/et\_search"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/btn\_search"  
 android:layout\_width="210dp"  
 android:layout\_height="60dp"  
 android:layout\_marginTop="44dp"  
 android:text="Найти"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/rView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
</LinearLayout>

Код Activity\_main.xml

1. Создан макет элементов RecycleVIew

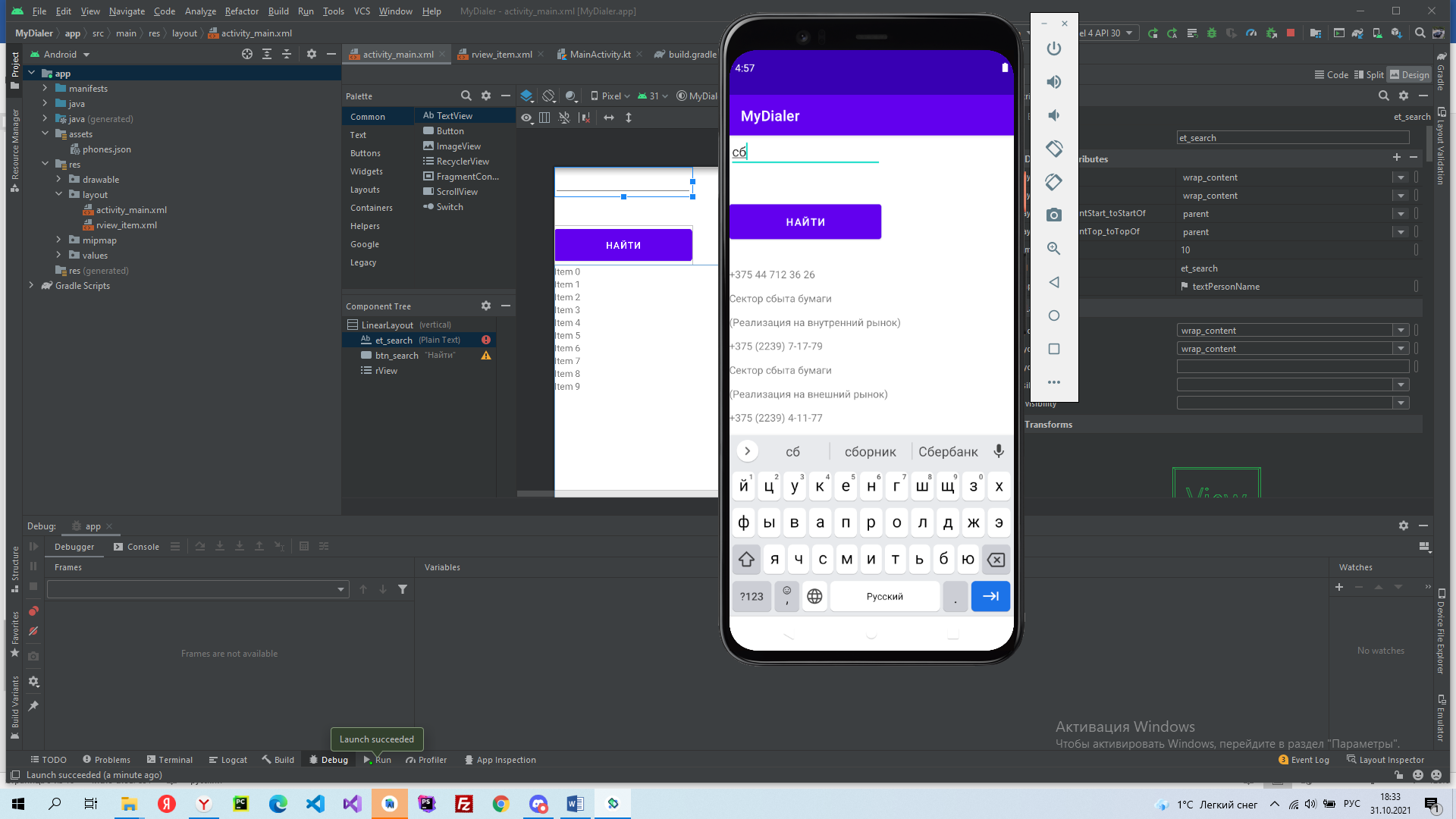
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/rview\_item"  
 android:layout\_width="match\_parent"  
 android:layout\_height="99dp"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/textName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="TextView" />  
  
 <TextView  
 android:id="@+id/textPhone"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="TextView" />  
  
 <TextView  
 android:id="@+id/textType"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="TextView" />  
</LinearLayout>

Код rview\_item.xml

Созданы обработчик кнопки, адаптер для RecycleView, класс данных Contact

package com.example.mydialer  
  
import android.content.Context  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.Button  
import android.widget.EditText  
import android.widget.TextView  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import com.google.gson.Gson  
import com.google.gson.JsonArray  
import timber.log.Timber  
import java.io.IOException  
import java.io.InputStream  
  
class MainActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 var json : String? = null  
 val button: Button = findViewById(R.id.*btn\_search*)  
 val txt: EditText = findViewById(R.id.*et\_search*)  
 val recyclerView: RecyclerView = findViewById(R.id.*rView*)  
  
 try {  
 val inp: InputStream = *assets*.open("phones.json")  
 json = inp.*bufferedReader*().*use***{it**.*readText*()**}**.*replace*("\r\n", "")  
  
 val phones = Gson().fromJson(json, Array<Contact>::class.*java*).*toList*()  
  
 recyclerView.*layoutManager* = LinearLayoutManager(this)  
 recyclerView.*adapter* = Adapter(this, phones as ArrayList<Contact>)  
  
 button.setOnClickListener**{** if(txt.*text*.toString() != ""){  
 val newArr = ArrayList<Contact>()  
 for(i in phones.*indices*){  
 if(txt.*text*.toString() in phones[i].name || txt.*text*.toString() in phones[i].phone || txt.*text*.toString() in phones[i].type){  
 newArr.add(phones[i])  
 }  
 }  
  
 recyclerView.*adapter* = Adapter(this, newArr)  
 }  
 else{  
 recyclerView.*adapter* = Adapter(this, phones)  
 }  
 **}** }  
 catch (e: IOException){  
  
 }  
 }  
}  
  
  
  
data class Contact(  
 var name: String,  
 var phone: String,  
 var type: String  
)  
  
class Adapter(  
 private val context: Context,  
 private val list: ArrayList<Contact>  
) : RecyclerView.Adapter<Adapter.ViewHolder>(){  
  
 class ViewHolder(view: View): RecyclerView.ViewHolder(view) {  
 val name: TextView = view.findViewById(R.id.*textName*)  
 val phone: TextView = view.findViewById(R.id.*textPhone*)  
 val type: TextView = view.findViewById(R.id.*textType*)  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
 val view = LayoutInflater.from(context).inflate(R.layout.*rview\_item*, parent, false)  
 return ViewHolder(view)  
 }  
  
 override fun getItemCount(): Int {  
 return list.*count*()  
 }  
  
 override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
 val contactList = list[position]  
 holder.name.*text* = contactList.name  
 holder.phone.*text* = contactList.phone  
 holder.type.*text* = contactList.type  
  
  
 }  
  
}

Код MainActivity.kt



Итоговое приложение

# Заключение

* Я ознакомился с особенностями создания динамического наполнения данными из интернета в RecyclerView.
* Также познакомился с созданием динамически обновляемого RecyclerView

**Литература**

1. JSON документ для работы - <https://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=ff49fcd4d4a08aa6aafb6ea3de826464&tags=cat&format=json&nojsoncallback=1>
2. GSON - <https://github.com/google/gson>
3. GridLayoutManager для RecyclerView - <https://www.codexpedia.com/android/android-recyclerview-with-gridlayoutmanager/>
4. JSON документ для работы - <https://drive.google.com/u/0/uc?id=1-KO-9GA3NzSgIc1dkAsNm8Dqw0fuPxcR&export=download>
5. Обновление данных в RecyclerView - <https://stackoverflow.com/questions/31367599/how-to-update-recyclerview-adapter-data>