НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №6

з дисципліни

“Програмування мобільних систем”

Виконав:

студент групи ІВ-82

ЗК ІВ-8224

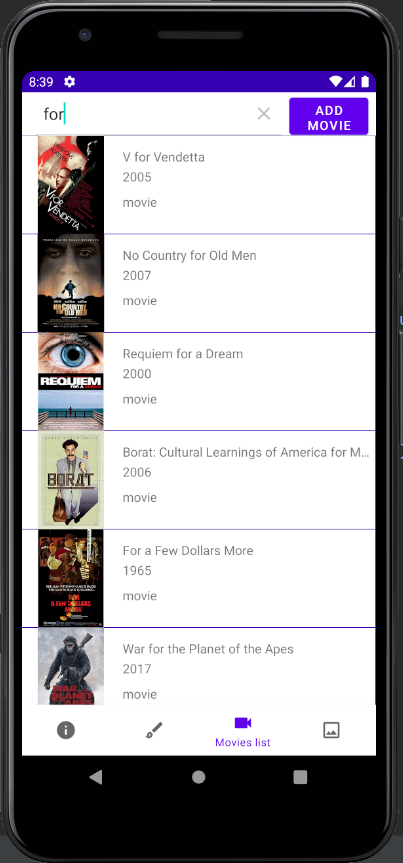
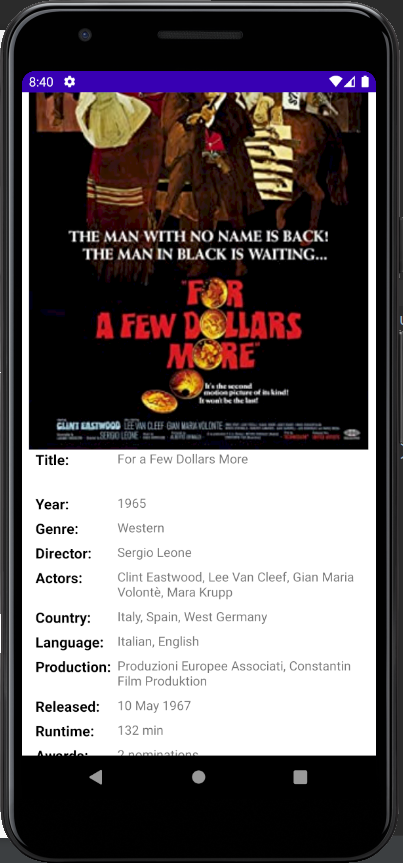
Старшинов Кирило

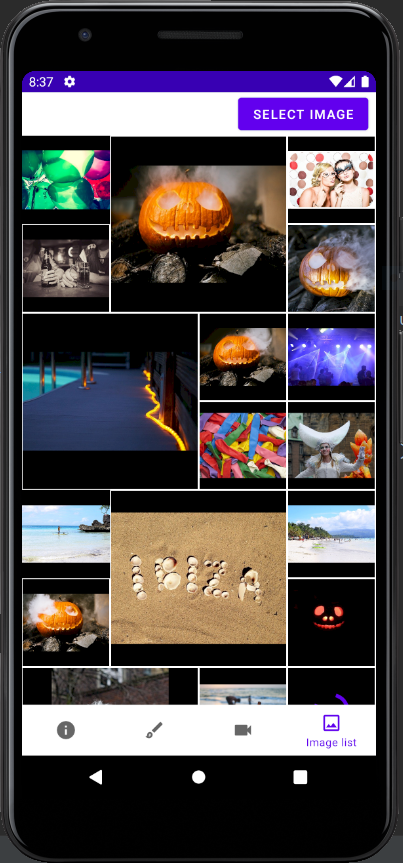
Київ 2021

Скріншоти роботи додатка

(**Варіант 8224 % 6 + 1 = 1**)

(**Варіант 8224 % 2 + 1 = 5**)



Лістинг коду

**MainActivity.java**

**package** ua.kpi.comsys.iv8224.pms;  
  
**import** android.os.Bundle;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.navigation.NavController;  
**import** androidx.navigation.Navigation;  
**import** androidx.navigation.ui.AppBarConfiguration;  
**import** androidx.navigation.ui.NavigationUI;  
  
**import** com.google.android.material.bottomnavigation.BottomNavigationView;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 BottomNavigationView navView = findViewById(R.id.***nav\_view***);  
 *// Passing each menu ID as a set of Ids because each  
 // menu should be considered as top level destinations.* AppBarConfiguration appBarConfiguration = **new** AppBarConfiguration.Builder(  
 R.id.***navigation\_info***, R.id.***navigation\_dashboard***, R.id.***navigation\_videocam***).build();  
 NavController navController = Navigation.*findNavController*(**this**, R.id.***nav\_host\_fragment***);  
  
 NavigationUI.*setupActionBarWithNavController*(**this**, navController, appBarConfiguration);  
 NavigationUI.*setupWithNavController*(navView, navController);  
  
 **if** (getSupportActionBar() != **null**) {  
 getSupportActionBar().hide();  
 }  
 }  
}

**Lab5.java**

**package** ua.kpi.comsys.iv8224.pms.lab5;  
  
**import** android.app.Activity;  
**import** android.content.Intent;  
**import** android.graphics.Color;  
**import** android.net.Uri;  
**import** android.os.AsyncTask;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.Button;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.ProgressBar;  
**import** android.widget.ScrollView;  
**import** android.widget.Toast;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.annotation.RequiresApi;  
**import** androidx.constraintlayout.widget.ConstraintLayout;  
**import** androidx.constraintlayout.widget.ConstraintSet;  
**import** androidx.constraintlayout.widget.Guideline;  
**import** androidx.core.content.ContextCompat;  
**import** androidx.fragment.app.Fragment;  
  
**import** org.json.simple.JSONArray;  
**import** org.json.simple.JSONObject;  
**import** org.json.simple.parser.JSONParser;  
**import** org.json.simple.parser.ParseException;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.net.MalformedURLException;  
**import** java.net.URL;  
**import** java.net.URLConnection;  
**import** java.util.ArrayList;  
  
**import** ua.kpi.comsys.iv8224.pms.R;  
**import** ua.kpi.comsys.iv8224.pms.lab3.Lab3;  
  
**public class** Lab5 **extends** Fragment {  
 **private static final int *RESULT\_LOAD\_IMAGE*** = 2;  
  
 **private static** View *root*;  
 **private static** ScrollView *scrollView*;  
 **private static** LinearLayout *scrollMain*;  
 **private static** ArrayList<ImageView> *allImages*;  
 **private static** ArrayList<ArrayList<Object>> *placeholderList*;  
  
 **public** View onCreateView(@NonNull LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
 *root* = inflater.inflate(R.layout.***lab5***, container, **false**);  
  
 *scrollView* = *root*.findViewById(R.id.***scrollview\_gallery***);  
 *scrollMain* = *root*.findViewById(R.id.***linear\_main***);  
  
 *allImages* = **new** ArrayList<>();  
 *placeholderList* = **new** ArrayList<>();  
  
 Button btnAddImage = *root*.findViewById(R.id.***button\_add\_img***);  
 btnAddImage.setOnClickListener(v -> {  
 Intent gallery = **new** Intent(Intent.***ACTION\_GET\_CONTENT***);  
 gallery.setType(**"image/\*"**);  
 startActivityForResult(gallery, ***RESULT\_LOAD\_IMAGE***);  
 });  
  
 AsyncLoadGallery aTask = **new** AsyncLoadGallery();  
 aTask.executeOnExecutor(AsyncTask.***THREAD\_POOL\_EXECUTOR***,  
 **"19193969-87191e5db266905fe8936d565"**,  
 **"\"fun+party\""**,  
 **"30"**);  
 **return** *root*;  
 }  
  
 **protected static void** loadImages(ArrayList<String> images){  
 **if** (images != **null**) {  
 **for** (String img : images) {  
 *addImage*(**false**, **null**, img);  
 }  
 }  
 **else** {  
 Toast.*makeText*(*root*.getContext(), **"Cannot load data!"**, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
  
 @Override  
 **public void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
  
 **if**(requestCode == ***RESULT\_LOAD\_IMAGE*** && resultCode == Activity.***RESULT\_OK***){  
 Uri imageUri = data.getData();  
 *addImage*(**true**, imageUri, **""**);  
 }  
 }  
 **private static void** addImage(**boolean** isLocal, Uri imageUri, String imageUrl) {  
  
 ProgressBar loadingImageBar = **new** ProgressBar(*root*.getContext());  
 loadingImageBar.setLayoutParams(  
 **new** ConstraintLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***,  
 ViewGroup.LayoutParams.***WRAP\_CONTENT***));  
 loadingImageBar.getIndeterminateDrawable().setColorFilter(  
 ContextCompat.*getColor*(*root*.getContext(), R.color.***purple\_500***),  
 android.graphics.PorterDuff.Mode.***MULTIPLY***);  
 loadingImageBar.setVisibility(View.***GONE***);  
 loadingImageBar.setId(loadingImageBar.hashCode());  
  
 ImageView newImage = **new** ImageView(*root*.getContext());  
 **if** (isLocal)  
 newImage.setImageURI(imageUri);  
 **else** {  
 loadingImageBar.setVisibility(View.***VISIBLE***);  
 **new** Lab3.DownloadImageTask(newImage, loadingImageBar, *root*.getContext()).execute(imageUrl);  
 }  
 newImage.setBackgroundColor(Color.***BLACK***);  
 ConstraintLayout.LayoutParams imageParams =  
 **new** ConstraintLayout.LayoutParams(ConstraintLayout.LayoutParams.***MATCH\_CONSTRAINT***,  
 ConstraintLayout.LayoutParams.***MATCH\_CONSTRAINT***);  
 imageParams.**dimensionRatio** = **"1"**;  
 newImage.setLayoutParams(imageParams);  
 newImage.setId(newImage.hashCode());  
 *setImagePlace*(newImage, loadingImageBar);  
  
 *allImages*.add(newImage);  
 }  
 **private static void** setImagePlace(ImageView newImage, ProgressBar loadBar){  
 ConstraintLayout tmpLayout = **null**;  
 ConstraintSet tmpSet = **null**;  
 **if** (*allImages*.size() > 0) {  
 tmpLayout = (ConstraintLayout) *getConstraintArrayList*(0, *placeholderList*);  
 **if** (*allImages*.size() % 10 != 0) {  
 tmpLayout.addView(newImage);  
 tmpLayout.addView(loadBar);  
 }  
 tmpSet = (ConstraintSet) *getConstraintArrayList*(1, *placeholderList*);  
  
 tmpSet.clone(tmpLayout);  
  
 tmpSet.setMargin(newImage.getId(), ConstraintSet.***START***, 3);  
 tmpSet.setMargin(newImage.getId(), ConstraintSet.***TOP***, 3);  
 tmpSet.setMargin(newImage.getId(), ConstraintSet.***END***, 3);  
 tmpSet.setMargin(newImage.getId(), ConstraintSet.***BOTTOM***, 3);  
  
 tmpSet.connect(loadBar.getId(), ConstraintSet.***START***, newImage.getId(), ConstraintSet.***START***);  
 tmpSet.connect(loadBar.getId(), ConstraintSet.***TOP***, newImage.getId(), ConstraintSet.***TOP***);  
 tmpSet.connect(loadBar.getId(), ConstraintSet.***END***, newImage.getId(), ConstraintSet.***END***);  
 tmpSet.connect(loadBar.getId(), ConstraintSet.***BOTTOM***, newImage.getId(), ConstraintSet.***BOTTOM***);  
 }  
  
  
 **switch** (*allImages*.size() % 10){  
 **case** 0:{  
 *placeholderList*.add(**new** ArrayList<>());  
  
 ConstraintLayout newConstraint = **new** ConstraintLayout(*root*.getContext());  
 *placeholderList*.get(*placeholderList*.size()-1).add(newConstraint);  
 newConstraint.setLayoutParams(  
 **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***MATCH\_PARENT***,  
 ViewGroup.LayoutParams.***WRAP\_CONTENT***));  
 *scrollMain*.addView(newConstraint);  
  
 Guideline vertical\_25 = *makeGuideline*(ConstraintLayout.LayoutParams.***VERTICAL***,  
 0.25f);  
 Guideline vertical\_50 = *makeGuideline*(ConstraintLayout.LayoutParams.***VERTICAL***,  
 0.50f);  
 Guideline vertical\_75 = *makeGuideline*(ConstraintLayout.LayoutParams.***VERTICAL***,  
 0.75f);  
  
 Guideline horizontal\_25 = *makeGuideline*(ConstraintLayout.LayoutParams.***HORIZONTAL***,  
 0.25f);  
 Guideline horizontal\_50 = *makeGuideline*(ConstraintLayout.LayoutParams.***HORIZONTAL***,  
 0.5f);  
 Guideline horizontal\_75 = *makeGuideline*(ConstraintLayout.LayoutParams.***HORIZONTAL***,  
 0.75f);  
  
 newConstraint.addView(vertical\_25, 0);  
 newConstraint.addView(vertical\_50, 1);  
 newConstraint.addView(vertical\_75, 2);  
 newConstraint.addView(horizontal\_25, 3);  
 newConstraint.addView(horizontal\_50, 4);  
 newConstraint.addView(horizontal\_75, 5);  
  
 newConstraint.addView(newImage);  
  
 ConstraintSet newConstraintSet = **new** ConstraintSet();  
 *placeholderList*.get(*placeholderList*.size()-1).add(newConstraintSet);  
 newConstraintSet.clone(newConstraint);  
  
 newConstraintSet.connect(newImage.getId(), ConstraintSet.***START***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***START***);  
 newConstraintSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***TOP***);  
 newConstraintSet.connect(newImage.getId(), ConstraintSet.***END***,  
 vertical\_25.getId(), ConstraintSet.***START***);  
 newConstraintSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 horizontal\_25.getId(), ConstraintSet.***TOP***);  
  
 newConstraintSet.applyTo(newConstraint);  
 **break**;  
 }  
  
 **case** 1: {  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(0).getId(), 0.25f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(2).getId(), 0.75f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(4).getId(), 0.5f);  
  
 tmpSet.connect(newImage.getId(), ConstraintSet.***START***,  
 tmpLayout.getChildAt(0).getId(), ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***TOP***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***END***,  
 tmpLayout.getChildAt(2).getId(), ConstraintSet.***START***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 tmpLayout.getChildAt(4).getId(), ConstraintSet.***TOP***);  
  
 tmpSet.applyTo(tmpLayout);  
 **break**;  
 }  
  
 **case** 2: {  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(2).getId(), 0.75f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(3).getId(), 0.25f);  
  
 tmpSet.connect(newImage.getId(), ConstraintSet.***START***,  
 tmpLayout.getChildAt(2).getId(), ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***TOP***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***END***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 tmpLayout.getChildAt(3).getId(), ConstraintSet.***TOP***);  
  
 tmpSet.applyTo(tmpLayout);  
 **break**;  
 }  
  
 **case** 3: {  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(0).getId(), 0.25f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(3).getId(), 0.25f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(4).getId(), 0.5f);  
  
 tmpSet.connect(newImage.getId(), ConstraintSet.***START***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***START***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 tmpLayout.getChildAt(3).getId(), ConstraintSet.***BOTTOM***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***END***,  
 tmpLayout.getChildAt(0).getId(), ConstraintSet.***START***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 tmpLayout.getChildAt(4).getId(), ConstraintSet.***TOP***);  
  
 tmpSet.applyTo(tmpLayout);  
 **break**;  
 }  
  
 **case** 4: {  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(2).getId(), 0.75f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(3).getId(), 0.25f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(4).getId(), 0.5f);  
  
 tmpSet.connect(newImage.getId(), ConstraintSet.***START***,  
 tmpLayout.getChildAt(2).getId(), ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 tmpLayout.getChildAt(3).getId(), ConstraintSet.***BOTTOM***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***END***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 tmpLayout.getChildAt(4).getId(), ConstraintSet.***TOP***);  
  
 tmpSet.applyTo(tmpLayout);  
 **break**;  
 }  
  
 **case** 5: {  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(1).getId(), 0.5f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(4).getId(), 0.5f);  
  
 tmpSet.connect(newImage.getId(), ConstraintSet.***START***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***START***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 tmpLayout.getChildAt(4).getId(), ConstraintSet.***BOTTOM***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***END***,  
 tmpLayout.getChildAt(1).getId(), ConstraintSet.***START***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***BOTTOM***);  
 tmpSet.applyTo(tmpLayout);  
 **break**;  
 }  
  
  
 **case** 6: {  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(1).getId(), 0.5f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(2).getId(), 0.75f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(4).getId(), 0.5f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(5).getId(), 0.75f);  
  
 tmpSet.connect(newImage.getId(), ConstraintSet.***START***,  
 tmpLayout.getChildAt(1).getId(), ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 tmpLayout.getChildAt(4).getId(), ConstraintSet.***BOTTOM***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***END***,  
 tmpLayout.getChildAt(2).getId(), ConstraintSet.***START***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 tmpLayout.getChildAt(5).getId(), ConstraintSet.***TOP***);  
 tmpSet.applyTo(tmpLayout);  
 **break**;  
 }  
  
 **case** 7: {  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(2).getId(), 0.75f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(4).getId(), 0.5f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(5).getId(), 0.75f);  
  
 tmpSet.connect(newImage.getId(), ConstraintSet.***START***,  
 tmpLayout.getChildAt(2).getId(), ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 tmpLayout.getChildAt(4).getId(), ConstraintSet.***BOTTOM***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***END***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 tmpLayout.getChildAt(5).getId(), ConstraintSet.***TOP***);  
 tmpSet.applyTo(tmpLayout);  
 **break**;  
 }  
  
 **case** 8: {  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(1).getId(), 0.5f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(2).getId(), 0.75f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(5).getId(), 0.75f);  
  
 tmpSet.connect(newImage.getId(), ConstraintSet.***START***,  
 tmpLayout.getChildAt(1).getId(), ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 tmpLayout.getChildAt(5).getId(), ConstraintSet.***BOTTOM***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***END***,  
 tmpLayout.getChildAt(2).getId(), ConstraintSet.***START***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***BOTTOM***);  
 tmpSet.applyTo(tmpLayout);  
 **break**;  
 }  
  
 **case** 9: {  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(2).getId(), 0.75f);  
 tmpSet.setGuidelinePercent(tmpLayout.getChildAt(5).getId(), 0.75f);  
  
 tmpSet.connect(newImage.getId(), ConstraintSet.***START***,  
 tmpLayout.getChildAt(2).getId(), ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***TOP***,  
 tmpLayout.getChildAt(5).getId(), ConstraintSet.***BOTTOM***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***END***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***END***);  
 tmpSet.connect(newImage.getId(), ConstraintSet.***BOTTOM***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***BOTTOM***);  
 tmpSet.applyTo(tmpLayout);  
 **break**;  
 }  
 }  
  
 }  
  
 **private static** Guideline makeGuideline(**int** orientation, **float** percent){  
 Guideline guideline = **new** Guideline(*root*.getContext());  
 guideline.setId(guideline.hashCode());  
  
 ConstraintLayout.LayoutParams guideline\_Params =  
 **new** ConstraintLayout.LayoutParams(ConstraintLayout.LayoutParams.***WRAP\_CONTENT***,  
 ConstraintLayout.LayoutParams.***WRAP\_CONTENT***);  
 guideline\_Params.**orientation** = orientation;  
  
 guideline.setLayoutParams(guideline\_Params);  
  
 guideline.setGuidelinePercent(percent);  
  
 **return** guideline;  
 }  
  
 **private static** Object getConstraintArrayList(**int** index, ArrayList<ArrayList<Object>> list){  
 **return** list.get(list.size()-1).get(index);  
 }  
  
 **private static class** AsyncLoadGallery **extends** AsyncTask<String, Void, ArrayList<String>> {  
 **private** String getRequest(String url){  
 StringBuilder result = **new** StringBuilder();  
 **try** {  
 URL getReq = **new** URL(url);  
 URLConnection bookConnection = getReq.openConnection();  
 BufferedReader in = **new** BufferedReader(**new** InputStreamReader(bookConnection.getInputStream()));  
 String inputLine;  
  
 **while** ((inputLine = in.readLine()) != **null**)  
 result.append(inputLine).append(**"\n"**);  
  
 in.close();  
  
 } **catch** (MalformedURLException e) {  
 System.***err***.println(String.*format*(**"Incorrect URL <%s>!"**, url));  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
 **return** result.toString();  
 }  
  
 **private** ArrayList<String> parseImages(String jsonText) **throws** ParseException {  
 ArrayList<String> result = **new** ArrayList<>();  
  
 JSONObject jsonObject = (JSONObject) **new** JSONParser().parse(jsonText);  
  
 JSONArray images = (JSONArray) jsonObject.get(**"hits"**);  
 **for** (Object img : images) {  
 JSONObject tmp = (JSONObject) img;  
 result.add((String) tmp.get(**"webformatURL"**));  
 }  
  
 **return** result;  
 }  
  
 **private** ArrayList<String> search(String api, String req, String count){  
 String jsonResponse = String.*format*(**"https://pixabay.com/api/?key=%s&q=%s&image\_type=photo&per\_page=%s"**,  
 api, req, count);  
 **try** {  
 **return** parseImages(getRequest(jsonResponse));  
 } **catch** (ParseException e) {  
 System.***err***.println(**"Incorrect content of JSON file!"**);  
 e.printStackTrace();  
 }  
 **return null**;  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.***M***)  
 @Override  
 **protected** ArrayList<String> doInBackground(String... strings) {  
 **return** search(strings[0], strings[1], strings[2]);  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.***M***)  
 @Override  
 **protected void** onPostExecute(ArrayList<String> images) {  
 **super**.onPostExecute(images);  
 Lab5.*loadImages*(images);  
 }  
 }  
}

**Lab3.java**

**package** ua.kpi.comsys.iv8224.pms.lab3;  
  
**import** android.annotation.SuppressLint;  
**import** android.app.Activity;  
**import** android.content.Context;  
**import** android.content.res.Configuration;  
**import** android.graphics.Bitmap;  
**import** android.graphics.BitmapFactory;  
**import** android.os.AsyncTask;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.util.DisplayMetrics;  
**import** android.util.Log;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.PopupWindow;  
**import** android.widget.ProgressBar;  
**import** android.widget.SearchView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.annotation.RequiresApi;  
**import** androidx.constraintlayout.widget.ConstraintLayout;  
**import** androidx.fragment.app.Fragment;  
  
**import** com.daimajia.swipe.SwipeLayout;  
  
**import** org.jetbrains.annotations.NotNull;  
**import** org.json.simple.JSONArray;  
**import** org.json.simple.JSONObject;  
**import** org.json.simple.parser.JSONParser;  
**import** org.json.simple.parser.ParseException;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.InputStreamReader;  
**import** java.net.MalformedURLException;  
**import** java.net.URL;  
**import** java.net.URLConnection;  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
**import** java.util.HashSet;  
**import** java.util.Set;  
  
**import** ua.kpi.comsys.iv8224.pms.R;  
  
**public class** Lab3 **extends** Fragment {  
 **private static** View *root*;  
 **private static** LinearLayout *moviesLayout*;  
 **private static** HashMap<SwipeLayout, Movie> *moviesHash*;  
 **private static** TextView *noItems*;  
 **private static** ProgressBar *loadingBar*;  
 **private static** Set<SwipeLayout> *removeSet*;  
  
 @RequiresApi(api = Build.VERSION\_CODES.***M***)  
 **public** View onCreateView(@NonNull LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
 *root* = inflater.inflate(R.layout.***lab3***, container, **false**);  
 *moviesLayout* = *root*.findViewById(R.id.***scroll\_lay***);  
 *moviesHash* = **new** HashMap<>();  
  
 *noItems* = *root*.findViewById(R.id.***no\_movies\_view***);  
 *loadingBar* = *root*.findViewById(R.id.***no\_items\_progressbar***);  
  
 *removeSet* = **new** HashSet<>();  
  
 SearchView searchView = *root*.findViewById(R.id.***search\_view***);  
  
 searchView.setOnQueryTextListener(**new** SearchView.OnQueryTextListener() {  
 @Override  
 **public boolean** onQueryTextSubmit(String query) {  
 *removeSet*.addAll(*moviesHash*.keySet());  
 **if** (query.length() >= 3) {  
 AsyncLoadMovies aTask = **new** AsyncLoadMovies();  
 *loadingBar*.setVisibility(View.***VISIBLE***);  
 *noItems*.setVisibility(View.***GONE***);  
 aTask.executeOnExecutor(AsyncTask.***THREAD\_POOL\_EXECUTOR***, query);  
 }  
 **else** {  
 **for** (SwipeLayout swipeLayout : *removeSet*) {  
 *binClicked*(swipeLayout);  
 }  
 *removeSet*.clear();  
 }  
 **return false**;  
 }  
  
 @Override  
 **public boolean** onQueryTextChange(String query) {  
 *removeSet*.addAll(*moviesHash*.keySet());  
 **if** (query.length() >= 3) {  
 AsyncLoadMovies aTask = **new** AsyncLoadMovies();  
 *loadingBar*.setVisibility(View.***VISIBLE***);  
 *noItems*.setVisibility(View.***GONE***);  
 aTask.executeOnExecutor(AsyncTask.***THREAD\_POOL\_EXECUTOR***, query);  
 }  
 **else** {  
 **for** (SwipeLayout swipeLayout : *removeSet*) {  
 *binClicked*(swipeLayout);  
 }  
 *removeSet*.clear();  
 }  
 **return false**;  
 }  
 });  
  
 Button btnAddMovie = *root*.findViewById(R.id.***button\_add\_movie***);  
 btnAddMovie.setOnClickListener(v -> {  
 MovieAdd popUpClass = **new** MovieAdd();  
 Object[] popups = popUpClass.showPopupWindow(v);  
  
 View popupView = (View) popups[0];  
 PopupWindow popupWindow = (PopupWindow) popups[1];  
  
 EditText inputTitle = popupView.findViewById(R.id.***input\_title***);  
 EditText inputYear = popupView.findViewById(R.id.***input\_year***);  
 EditText inputType = popupView.findViewById(R.id.***input\_type***);  
  
 Button buttonAdd = popupView.findViewById(R.id.***button\_add\_add***);  
 buttonAdd.setOnClickListener(v1 -> {  
 **if** (inputTitle.getText().toString().length() != 0 &&  
 inputYear.getText().toString().length() != 0 &&  
 inputType.getText().toString().length() != 0) {  
 Object[] tmp = **new** MovieList(*root*.getContext(), *moviesLayout*,  
 **new** Movie(inputTitle.getText().toString(),  
 inputYear.getText().toString(),  
 inputType.getText().toString())).**moviePack**;  
  
 *moviesHash*.put((SwipeLayout) tmp[0], (Movie)tmp[1]);  
 changeLaySizes();  
  
 popupWindow.dismiss();  
 }  
 **else**{  
 Toast.*makeText*(getActivity(), **"Incorrect data!"**,  
 Toast.***LENGTH\_LONG***).show();  
 }  
 });  
 });  
  
  
 changeLaySizes();  
  
 **return** *root*;  
 }  
  
  
 **protected static void** loadMovies(ArrayList<Movie> movies){  
 **if** (movies != **null**) {  
 **for** (SwipeLayout swipeLayout : *removeSet*) {  
 *binClicked*(swipeLayout);  
 }  
 *removeSet*.clear();  
 **if** (movies.size() > 0) {  
 *noItems*.setVisibility(View.***GONE***);  
 **for** (Movie movie :  
 movies) {  
 Object[] tmp = **new** MovieList(*root*.getContext(), *moviesLayout*, movie).**moviePack**;  
  
 *moviesHash*.put((SwipeLayout) tmp[0], (Movie)tmp[1]);  
 }  
 } **else** {  
 *noItems*.setVisibility(View.***VISIBLE***);  
 }  
 }  
 **else** {  
 *noItems*.setVisibility(View.***VISIBLE***);  
 Toast.*makeText*(*root*.getContext(), **"Cannot load data!"**, Toast.***LENGTH\_LONG***).show();  
 }  
 *loadingBar*.setVisibility(View.***GONE***);  
 }  
  
 **public static void** binClicked(SwipeLayout swipeLayout){  
 *moviesHash*.remove(swipeLayout);  
 *moviesLayout*.removeView(swipeLayout);  
 **if** (*moviesHash*.keySet().isEmpty()){  
 *noItems*.setVisibility(View.***VISIBLE***);  
 }  
 }  
  
 @Override  
 **public void** onConfigurationChanged(@NotNull Configuration newConfig) {  
 **super**.onConfigurationChanged(newConfig);  
  
 changeLaySizes();  
 }  
  
 **private void** changeLaySizes(){  
 DisplayMetrics displayMetrics = **new** DisplayMetrics();  
 ((Activity) *root*.getContext()).getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);  
  
 **for** (SwipeLayout moviesList : *moviesHash*.keySet()) {  
 ((ConstraintLayout)moviesList.getChildAt(1)).getChildAt(0).setLayoutParams(  
 **new** ConstraintLayout.LayoutParams(300, 300));  
 }  
 }  
  
 **private static class** AsyncLoadMovies **extends** AsyncTask<String, Void, ArrayList<Movie>> {  
 **private** String getRequest(String url) {  
 StringBuilder result = **new** StringBuilder();  
 **try** {  
 URL getReq = **new** URL(url);  
 URLConnection movieConnection = getReq.openConnection();  
 BufferedReader in = **new** BufferedReader(**new** InputStreamReader(movieConnection.getInputStream()));  
 String inputLine;  
  
 **while** ((inputLine = in.readLine()) != **null**)  
 result.append(inputLine).append(**"\n"**);  
  
 in.close();  
  
 } **catch** (MalformedURLException e) {  
 System.***err***.println(String.*format*(**"Incorrect URL <%s>!"**, url));  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
 **return** result.toString();  
 }  
  
 **private** ArrayList<Movie> parseMovies(String jsonText) **throws** ParseException {  
 ArrayList<Movie> result = **new** ArrayList<>();  
  
 org.json.simple.JSONObject jsonObject = (org.json.simple.JSONObject) **new** JSONParser().parse(jsonText);  
  
 org.json.simple.JSONArray movies = (JSONArray) jsonObject.get(**"Search"**);  
 **for** (Object movie : movies) {  
 org.json.simple.JSONObject tmp = (JSONObject) movie;  
 result.add(**new** Movie(  
 (String) tmp.get(**"Title"**),  
 (String) tmp.get(**"Year"**),  
 (String) tmp.get(**"imdbID"**),  
 (String) tmp.get(**"Type"**),  
 (String) tmp.get(**"Poster"**)  
 ));  
 }  
  
 **return** result;  
 }  
 **private** ArrayList<Movie> search(String newText){  
 String API\_KEY = **"7e9fe69e"**;  
 String jsonResponse = String.*format*(**"http://www.omdbapi.com/?apikey=%s&s=\"%s\"&page=1"**, API\_KEY, newText);  
 **try** {  
 ArrayList<Movie> movies = parseMovies(getRequest(jsonResponse));  
 **return** movies;  
 } **catch** (ParseException e) {  
 System.***err***.println(**"Incorrect content of JSON file!"**);  
 e.printStackTrace();  
 }  
 **return null**;  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.***M***)  
 @Override  
 **protected** ArrayList<Movie> doInBackground(String... strings) {  
 **return** search(strings[0]);  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.***M***)  
 @Override  
 **protected void** onPostExecute(ArrayList<Movie> movies) {  
 **super**.onPostExecute(movies);  
 Lab3.*loadMovies*(movies);  
 }  
 }  
 **public static class** DownloadImageTask **extends** AsyncTask<String, Void, Bitmap> {  
 @SuppressLint(**"StaticFieldLeak"**)  
 ImageView **bmImage**;  
 @SuppressLint(**"StaticFieldLeak"**)  
 ProgressBar **loadingBar**;  
 @SuppressLint(**"StaticFieldLeak"**)  
 Context **context**;  
  
 **public** DownloadImageTask(ImageView bmImage, ProgressBar loadingBar, Context context) {  
 **this**.**bmImage** = bmImage;  
 **this**.**loadingBar** = loadingBar;  
 **this**.**context** = context;  
 }  
  
 **protected** Bitmap doInBackground(String... urls) {  
 String urldisplay = urls[0];  
 Bitmap mIcon11 = **null**;  
 **try** {  
 InputStream in = **new** java.net.URL(urldisplay).openStream();  
 mIcon11 = BitmapFactory.*decodeStream*(in);  
 } **catch** (Exception e) {  
 Log.*e*(**"Error"**, e.getMessage());  
 e.printStackTrace();  
 }  
 **return** mIcon11;  
 }  
  
 **protected void** onPostExecute(Bitmap result) {  
 **if** (result != **null**)  
 **bmImage**.setImageBitmap(result);  
 **else** {  
 **bmImage**.setBackgroundResource(R.drawable.***not\_found***);  
 Toast.*makeText*(**context**, **"Cannot load data!"**, Toast.***LENGTH\_LONG***).show();  
 }  
 **loadingBar**.setVisibility(View.***GONE***);  
 **bmImage**.setVisibility(View.***VISIBLE***);  
 }  
 }  
  
}

**Movie.java**

**package** ua.kpi.comsys.iv8224.pms.lab3;  
  
**public class** Movie {  
 **private final** String **title**;  
 **private final** String **year**;  
 **private** String **rated**;  
 **private** String **released**;  
 **private** String **runtime**;  
 **private** String **genre**;  
 **private** String **director**;  
 **private** String **writer**;  
 **private** String **actors**;  
 **private** String **plot**;  
 **private** String **language**;  
 **private** String **country**;  
 **private** String **awards**;  
 **private** String **imdbRating**;  
 **private** String **imdbVotes**;  
 **private** String **production**;  
 **private final** String **imdbID**;  
 **private final** String **type**;  
 **private final** String **posterPath**;  
  
  
 **public** Movie(String title, String year, String type) {  
 **this**.**title** = title;  
 **this**.**year** = year;  
 **this**.**imdbID** = **"noid"**;  
 **this**.**type** = type;  
 **this**.**posterPath** = **""**;  
  
 }  
  
 **public** Movie(String title, String year, String imdbID, String type, String posterPath){  
 **this**.**title** = title;  
 **this**.**year** = year;  
 **this**.**imdbID** = imdbID;  
 **this**.**type** = type;  
 **this**.**posterPath** = posterPath;  
 }  
  
 **public void** setRated(String rated) {**this**.**rated** = rated;}  
 **public void** setReleased(String released) {**this**.**released** = released;}  
 **public void** setRuntime(String runtime) {**this**.**runtime** = runtime;}  
 **public void** setGenre(String genre) {**this**.**genre** = genre;}  
 **public void** setDirector(String director) {**this**.**director** = director;}  
 **public void** setWriter(String writer) {**this**.**writer** = writer;}  
 **public void** setActors(String actors) {**this**.**actors** = actors;}  
 **public void** setPlot(String plot) {**this**.**plot** = plot;}  
 **public void** setLanguage(String language) {**this**.**language** = language;}  
 **public void** setCountry(String country) {**this**.**country** = country;}  
 **public void** setAwards(String awards) {**this**.**awards** = awards;}  
 **public void** setImdbRating(String imdbRating) {**this**.**imdbRating** = imdbRating;}  
 **public void** setImdbVotes(String imdbVotes) {**this**.**imdbVotes** = imdbVotes;}  
 **public void** setProduction(String production) {**this**.**production** = production;}  
  
 **public** String getRated() {**return rated**;}  
 **public** String getReleased() {**return released**;}  
 **public** String getRuntime() {**return runtime**;}  
 **public** String getGenre() {**return genre**;}  
 **public** String getDirector() {**return director**;}  
 **public** String getWriter() {**return writer**;}  
 **public** String getActors() {**return actors**;}  
 **public** String getPlot() {**return plot**;}  
 **public** String getLanguage() {**return language**;}  
 **public** String getCountry() {**return country**;}  
 **public** String getAwards() {**return awards**;}  
 **public** String getImdbRating() {**return imdbRating**;}  
 **public** String getImdbVotes() {**return imdbVotes**;}  
 **public** String getProduction() {**return production**;}  
  
 **public** String getTitle() {**return title**;}  
 **public** String getYear() {**return year**;}  
 **public** String getImdbID() {**return imdbID**;}  
 **public** String getType() {**return type**;}  
 **public** String getPosterPath() {**return posterPath**;}  
}

**MovieAdd.java**

**package** ua.kpi.comsys.iv8224.pms.lab3;  
  
**import** android.annotation.SuppressLint;  
**import** android.content.Context;  
**import** android.os.Build;  
**import** android.view.Gravity;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.widget.LinearLayout;  
**import** android.widget.PopupWindow;  
  
**import** androidx.annotation.RequiresApi;  
  
**import** ua.kpi.comsys.iv8224.pms.R;  
  
**public class** MovieAdd {  
 @RequiresApi(api = Build.VERSION\_CODES.***M***)  
 @SuppressLint(**"ClickableViewAccessibility"**)  
 **public** Object[] showPopupWindow(**final** View view) {  
 view.getContext();  
 LayoutInflater inflater = (LayoutInflater) view.getContext().getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 @SuppressLint(**"InflateParams"**)  
 View popupView = inflater.inflate(R.layout.***movie\_add***, **null**);  
  
 **int** width = LinearLayout.LayoutParams.***MATCH\_PARENT***;  
 **int** height = LinearLayout.LayoutParams.***MATCH\_PARENT***;  
  
 **boolean** focusable = **true**;  
  
 **final** PopupWindow popupWindow = **new** PopupWindow(popupView, width, height, focusable);  
 popupWindow.showAtLocation(view, Gravity.***CENTER***, 0, 0);  
  
 popupView.setOnTouchListener((v, event) -> {  
 **return true**;  
 });  
  
 **return new** Object[] {popupView, popupWindow};  
 }  
}

**MovieInfo.java**

**package** ua.kpi.comsys.iv8224.pms.lab3;  
  
**import** android.annotation.SuppressLint;  
**import** android.content.Context;  
**import** android.os.AsyncTask;  
**import** android.view.Gravity;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.PopupWindow;  
**import** android.widget.ProgressBar;  
**import** android.widget.TextView;  
  
**import** org.json.simple.JSONObject;  
**import** org.json.simple.parser.JSONParser;  
**import** org.json.simple.parser.ParseException;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.net.MalformedURLException;  
**import** java.net.URL;  
**import** java.net.URLConnection;  
  
**import** ua.kpi.comsys.iv8224.pms.R;  
  
**public class** MovieInfo {  
 **private static** View *popupView*;  
 **private static** ProgressBar *loadingImage*;  
 **private static** ImageView *movieImage*;  
 **private static** Movie *movie*;  
  
 @SuppressLint(**"ClickableViewAccessibility"**)  
 **public void** showPopupWindow(**final** View view, Movie movie) {  
 view.getContext();  
 LayoutInflater inflater = (LayoutInflater) view.getContext().getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 *popupView* = inflater.inflate(R.layout.***movie\_info***, **null**);  
 MovieInfo.*movie* = movie;  
  
  
 **int** width = LinearLayout.LayoutParams.***MATCH\_PARENT***;  
 **int** height = LinearLayout.LayoutParams.***MATCH\_PARENT***;  
  
 **boolean** focusable = **true**;  
  
 **final** PopupWindow popupWindow = **new** PopupWindow(*popupView*, width, height, focusable);  
 popupWindow.showAtLocation(view, Gravity.***CENTER***, 0, 0);  
  
 *loadingImage* = *popupView*.findViewById(R.id.***loadingImageInfo***);  
 *movieImage* = *popupView*.findViewById(R.id.***movie\_info\_image***);  
 AsyncLoadMovieInfo aTask = **new** AsyncLoadMovieInfo();  
 aTask.executeOnExecutor(AsyncTask.***THREAD\_POOL\_EXECUTOR***, movie.getImdbID());  
  
 }  
 **protected static void** setInfoData() {  
 *movieImage*.setVisibility(View.***INVISIBLE***);  
 *loadingImage*.setVisibility(View.***VISIBLE***);  
 **new** Lab3.DownloadImageTask(*movieImage*, *loadingImage*, *popupView*.getContext()).execute(*movie*.getPosterPath());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_title***)).setText(*movie*.getTitle());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_year***)).setText(*movie*.getYear());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_genre***)).setText(*movie*.getGenre());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_director***)).setText(*movie*.getDirector());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_actors***)).setText(*movie*.getActors());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_country***)).setText(*movie*.getCountry());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_language***)).setText(*movie*.getLanguage());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_production***)).setText(*movie*.getProduction());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_released***)).setText(*movie*.getReleased());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_runtime***)).setText(*movie*.getRuntime());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_awards***)).setText(*movie*.getAwards());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_rating***)).setText(*movie*.getImdbRating());  
 ((TextView) *popupView*.findViewById(R.id.***movie\_info\_plot***)).setText(*movie*.getPlot());  
 }  
 **private static class** AsyncLoadMovieInfo **extends** AsyncTask<String, Void, Void> {  
 **private** String getRequest(String url){  
 StringBuilder result = **new** StringBuilder();  
 **try** {  
 URL getReq = **new** URL(url);  
 URLConnection movieConnection = getReq.openConnection();  
 BufferedReader in = **new** BufferedReader(**new** InputStreamReader(movieConnection.getInputStream()));  
 String inputLine;  
  
 **while** ((inputLine = in.readLine()) != **null**)  
 result.append(inputLine).append(**"\n"**);  
  
 in.close();  
  
 } **catch** (MalformedURLException e) {  
 System.***err***.println(String.*format*(**"Incorrect URL <%s>!"**, url));  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **return** result.toString();  
 }  
  
 **private void** parseMovieInfo(String jsonText) **throws** ParseException {  
 JSONObject jsonObject = (JSONObject) **new** JSONParser().parse(jsonText);  
  
 *movie*.setGenre((String) jsonObject.get(**"Genre"**));  
 *movie*.setDirector((String) jsonObject.get(**"Director"**));  
 *movie*.setActors((String) jsonObject.get(**"Actors"**));  
 *movie*.setCountry((String) jsonObject.get(**"Country"**));  
 *movie*.setLanguage((String) jsonObject.get(**"Language"**));  
 *movie*.setProduction((String) jsonObject.get(**"Production"**));  
 *movie*.setReleased((String) jsonObject.get(**"Released"**));  
 *movie*.setRuntime((String) jsonObject.get(**"Runtime"**));  
 *movie*.setAwards((String) jsonObject.get(**"Awards"**));  
 *movie*.setImdbRating(jsonObject.get(**"imdbRating"**) + **"/10"**);  
 *movie*.setPlot((String) jsonObject.get(**"Plot"**));  
  
 }  
 **private void** search(String imdbID) {  
 String API\_KEY=**"493d9265"**;  
 String jsonResponse = String.*format*(**"http://www.omdbapi.com/?apikey=%s&i=%s"**, API\_KEY, imdbID);  
 **try** {  
 parseMovieInfo(getRequest(jsonResponse));  
 } **catch** (ParseException e) {  
 System.***err***.println(**"Incorrect content of JSON file!"**);  
 e.printStackTrace();  
 }  
 }  
 @Override  
 **protected** Void doInBackground(String... strings) {  
 search(strings[0]);  
 **return null**;  
 }  
  
 @Override  
 **protected void** onPostExecute(Void aVoid) {  
 **super**.onPostExecute(aVoid);  
 MovieInfo.*setInfoData*();  
 }  
 }  
}

**MovieList.java**

**package** ua.kpi.comsys.iv8224.pms.lab3;  
  
**import** android.content.Context;  
**import** android.graphics.Color;  
**import** android.text.TextUtils;  
**import** android.view.Gravity;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.ImageButton;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.ProgressBar;  
**import** android.widget.TextView;  
  
**import** androidx.constraintlayout.widget.ConstraintLayout;  
**import** androidx.constraintlayout.widget.ConstraintSet;  
**import** androidx.core.content.ContextCompat;  
  
**import** com.daimajia.swipe.SwipeLayout;  
  
**import** ua.kpi.comsys.iv8224.pms.R;  
  
**public class** MovieList {  
  
 **public** Object[] **moviePack**;  
  
 **public** MovieList(Context context, LinearLayout movieList, Movie movie){  
 **moviePack** = newMovieList(context, movieList, movie);  
 }  
  
 **private** Object[] newMovieList(Context context, LinearLayout movieList, Movie movie){  
 SwipeLayout swipeLayout = **new** SwipeLayout(context);  
 swipeLayout.setShowMode(SwipeLayout.ShowMode.***PullOut***);  
 swipeLayout.setLayoutParams(  
 **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***MATCH\_PARENT***,  
 ViewGroup.LayoutParams.***WRAP\_CONTENT***));  
 movieList.addView(swipeLayout);  
  
 ImageButton deleteButton = **new** ImageButton(context);  
 deleteButton.setImageResource(R.drawable.***ic\_delete\_sweep\_black\_24dp***);  
 deleteButton.setBackgroundColor(Color.***RED***);  
 deleteButton.setPadding(50, 0, 50, 0);  
 LinearLayout.LayoutParams btnBinParams =  
 **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***,  
 ViewGroup.LayoutParams.***MATCH\_PARENT***);  
 btnBinParams.**gravity** = Gravity.***RIGHT***;  
 swipeLayout.setShowMode(SwipeLayout.ShowMode.***PullOut***);  
 swipeLayout.addView(deleteButton, 0, btnBinParams);  
  
 ConstraintLayout movieLayTmp = **new** ConstraintLayout(context);  
 movieLayTmp.setBackgroundResource(R.drawable.***movielist***);  
 movieLayTmp.setLayoutParams(  
 **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***MATCH\_PARENT***,  
 ViewGroup.LayoutParams.***WRAP\_CONTENT***));  
 swipeLayout.addView(movieLayTmp, 1);  
  
 deleteButton.setOnClickListener(v -> Lab3.*binClicked*(swipeLayout));  
 movieLayTmp.setOnClickListener(v -> {  
 **if** (movie.getImdbID().length() != 0 && !movie.getImdbID().equals(**"noid"**)) {  
 MovieInfo popUpClass = **new** MovieInfo();  
 popUpClass.showPopupWindow(v, movie);  
 }  
 });  
  
 ProgressBar loadingImageBar = **new** ProgressBar(context);  
 loadingImageBar.getIndeterminateDrawable().setColorFilter(  
 ContextCompat.*getColor*(context, R.color.***purple\_500***),  
 android.graphics.PorterDuff.Mode.***MULTIPLY***);  
 loadingImageBar.setVisibility(View.***GONE***);  
 loadingImageBar.setId(loadingImageBar.hashCode());  
 movieLayTmp.addView(loadingImageBar);  
  
 ImageView imageTmp = **new** ImageView(context);  
 imageTmp.setId(imageTmp.hashCode());  
 **if** (movie.getPosterPath().length() != 0){  
 imageTmp.setVisibility(View.***INVISIBLE***);  
 loadingImageBar.setVisibility(View.***VISIBLE***);  
 **new** Lab3.DownloadImageTask(imageTmp, loadingImageBar, context).execute(movie.getPosterPath());  
 }  
 ConstraintLayout.LayoutParams imgParams =  
 **new** ConstraintLayout.LayoutParams(300, 300);  
 movieLayTmp.addView(imageTmp, imgParams);  
  
 TextView textTitle = **new** TextView(context);  
 textTitle.setText(movie.getTitle());  
 textTitle.setEllipsize(TextUtils.TruncateAt.***END***);  
 textTitle.setMaxLines(1);  
 textTitle.setId(textTitle.hashCode());  
 movieLayTmp.addView(textTitle, **new** ConstraintLayout.LayoutParams(  
 ConstraintLayout.LayoutParams.***MATCH\_CONSTRAINT***,  
 ConstraintLayout.LayoutParams.***WRAP\_CONTENT***));  
  
 TextView textYear = **new** TextView(context);  
 textYear.setText(movie.getYear());  
 textYear.setEllipsize(TextUtils.TruncateAt.***END***);  
 textYear.setMaxLines(4);  
 textYear.setId(textYear.hashCode());  
 movieLayTmp.addView(textYear, **new** ConstraintLayout.LayoutParams(  
 ConstraintLayout.LayoutParams.***MATCH\_CONSTRAINT***,  
 ConstraintLayout.LayoutParams.***WRAP\_CONTENT***));  
  
 TextView textType = **new** TextView(context);  
 textType.setText(movie.getType());  
 textType.setId(textType.hashCode());  
 movieLayTmp.addView(textType, **new** ConstraintLayout.LayoutParams(  
 ConstraintLayout.LayoutParams.***MATCH\_CONSTRAINT***,  
 ConstraintLayout.LayoutParams.***WRAP\_CONTENT***));  
  
 ConstraintSet textConstraintSet = **new** ConstraintSet();  
 textConstraintSet.clone(movieLayTmp);  
  
 textConstraintSet.connect(imageTmp.getId(), ConstraintSet.***START***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***START***);  
 textConstraintSet.connect(imageTmp.getId(), ConstraintSet.***TOP***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***TOP***);  
 textConstraintSet.connect(imageTmp.getId(), ConstraintSet.***BOTTOM***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***BOTTOM***);  
  
 textConstraintSet.connect(textTitle.getId(), ConstraintSet.***START***,  
 imageTmp.getId(), ConstraintSet.***END***);  
 textConstraintSet.connect(textTitle.getId(), ConstraintSet.***TOP***,  
 imageTmp.getId(), ConstraintSet.***TOP***);  
 textConstraintSet.connect(textTitle.getId(), ConstraintSet.***END***,  
 ConstraintSet.***PARENT\_ID***, ConstraintSet.***END***);  
 textConstraintSet.connect(textTitle.getId(), ConstraintSet.***BOTTOM***,  
 imageTmp.getId(), ConstraintSet.***BOTTOM***);  
  
 textConstraintSet.connect(textYear.getId(), ConstraintSet.***START***,  
 textTitle.getId(), ConstraintSet.***START***);  
 textConstraintSet.connect(textYear.getId(), ConstraintSet.***TOP***,  
 textTitle.getId(), ConstraintSet.***BOTTOM***);  
 textConstraintSet.connect(textYear.getId(), ConstraintSet.***END***,  
 textTitle.getId(), ConstraintSet.***END***);  
  
 textConstraintSet.connect(textType.getId(), ConstraintSet.***START***,  
 textYear.getId(), ConstraintSet.***START***);  
 textConstraintSet.connect(textType.getId(), ConstraintSet.***TOP***,  
 textYear.getId(), ConstraintSet.***BOTTOM***);  
 textConstraintSet.connect(textType.getId(), ConstraintSet.***END***,  
 textYear.getId(), ConstraintSet.***END***);  
  
 textConstraintSet.setMargin(textTitle.getId(), ConstraintSet.***START***, 8);  
 textConstraintSet.setMargin(textTitle.getId(), ConstraintSet.***END***, 8);  
 textConstraintSet.setVerticalBias(textTitle.getId(), 0.15f);  
  
 textConstraintSet.setMargin(textYear.getId(), ConstraintSet.***TOP***, 8);  
  
 textConstraintSet.setMargin(textType.getId(), ConstraintSet.***TOP***, 24);  
  
 textConstraintSet.applyTo(movieLayTmp);  
  
 **return new** Object[] {swipeLayout, movie};  
 }  
}

**Висновок**

В даній лабораторній роботі було удосконалено мобільний додаток, та добавлено нові функції:

* пошук фільмів через мережу
* підтягування онлайн галереї з фотографіями за варіантом

Програма працює коректно. Кінцева мета досягнута.