НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №3

з дисципліни

“Програмування мобільних систем”

Виконав:

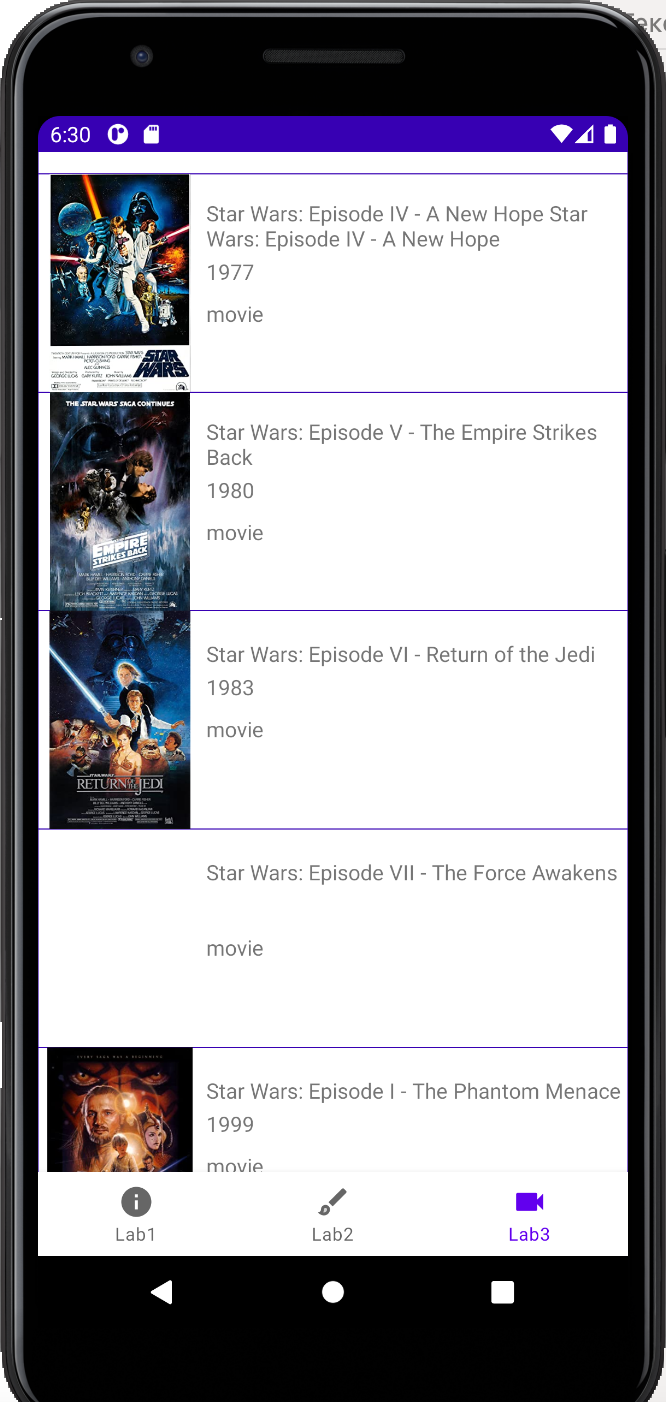
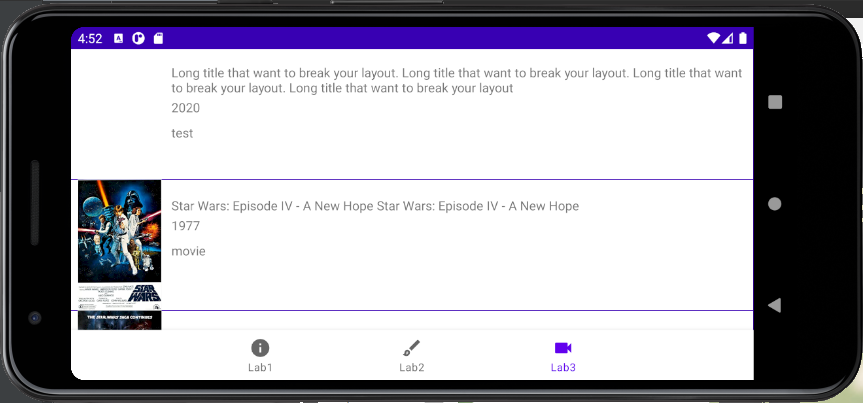
студент групи ІВ-82

ЗК ІВ-8224

Старшинов Кирило

Київ 2021

Скріншоти роботи додатка (**Варіант 8224 % 2 + 1 = 1**)

Лістинг коду

**MainActivity.java**

**package** ua.kpi.comsys.iv8224.third;  
  
**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***);  
 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();  
 }  
  
 }  
  
}

**Movie.java**

**package** ua.kpi.comsys.iv8224.third.lab3;  
  
**public class** Movie {  
 **private final** String **title**;  
 **private final** String **year**;  
 **private final** String **imdbID**;  
 **private final** String **type**;  
 **private final** String **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** String getTitle() {  
 **return title**;  
 }  
  
 **public** String getYear() {  
 **return year**;  
 }  
  
 **public** String getImdbID() {  
 **return imdbID**;  
 }  
  
 **public** String getType() {  
 **return type**;  
 }  
  
 **public** String getPosterPath() {  
 **return posterPath**;  
 }  
}

**MoviesFragment.java**

**package** ua.kpi.comsys.iv8224.third.lab3;  
  
**import** android.content.Context;  
**import** android.text.TextUtils;  
**import** android.view.ViewGroup;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
  
**import** androidx.constraintlayout.widget.ConstraintLayout;  
**import** androidx.constraintlayout.widget.ConstraintSet;  
  
**import** java.lang.reflect.Field;  
  
**import** ua.kpi.comsys.iv8224.third.R;  
  
**public class** MovieList {  
  
 **public** ConstraintLayout **moviePack**;  
  
 **public** MovieList(Context context, LinearLayout movieList, Movie movie){  
 **moviePack** = newMovieList(context, movieList, movie);  
 }  
  
 **private** ConstraintLayout newMovieList(Context context, LinearLayout movieList, Movie movie){  
 ConstraintLayout movieLayTmp = **new** ConstraintLayout(context);  
 movieLayTmp.setBackgroundResource(R.drawable.***movielist***);  
 movieLayTmp.setLayoutParams(  
 **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***MATCH\_PARENT***,  
 ViewGroup.LayoutParams.***WRAP\_CONTENT***));  
 movieList.addView(movieLayTmp);  
  
 ImageView imageTmp = **new** ImageView(context);  
 imageTmp.setId(imageTmp.hashCode());  
 **if** (movie.getPosterPath().length() != 0)  
 imageTmp.setImageResource(  
 *getResId*(movie.getPosterPath().toLowerCase().split(**"\\."**)[0], R.drawable.**class**));  
 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(4);  
 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(1);  
 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** movieLayTmp;  
 }  
  
 **public static int** getResId(String resName, Class<?> c) {  
 **try** {  
 Field idField = c.getDeclaredField(resName);  
 **return** idField.getInt(idField);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 **return** -1;  
 }  
 }  
}

**Lab3.java**

**package** ua.kpi.comsys.iv8224.third.lab3;  
  
**import** android.app.Activity;  
**import** android.content.Context;  
**import** android.content.res.Configuration;  
**import** android.os.Bundle;  
**import** android.util.DisplayMetrics;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.LinearLayout;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.annotation.RawRes;  
**import** androidx.constraintlayout.widget.ConstraintLayout;  
**import** androidx.fragment.app.Fragment;  
  
**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.ByteArrayOutputStream;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.util.ArrayList;  
  
**import** ua.kpi.comsys.iv8224.third.R;  
  
**public class** Lab3 **extends** Fragment {  
 **private** View **root**;  
 **private** ArrayList<ConstraintLayout> **moviesLayout**;  
  
 **public** View onCreateView(@NonNull LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
 **root** = inflater.inflate(R.layout.***lab3***, container, **false**);  
 LinearLayout movieLinear = **root**.findViewById(R.id.***scroll\_lay***);  
 **moviesLayout** = **new** ArrayList<>();  
  
 **try** {  
 ArrayList<Movie> movies = parseMovies(*readTextFile*(**root**.getContext(), R.raw.***movies***));  
 **for** (Movie movie : movies) {  
 **moviesLayout**.add(**new** MovieList(**root**.getContext(), movieLinear, movie).**moviePack**);  
 }  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
  
 changeLaySizes();  
  
 **return root**;  
 }  
  
 @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);  
 **int** width = displayMetrics.**widthPixels**;  
 **for** (ConstraintLayout moviesList :  
 **moviesLayout**) {  
 moviesList.getChildAt(0).setLayoutParams(  
 **new** ConstraintLayout.LayoutParams(300, 400));  
 }  
 }  
 **public static** String readTextFile(Context context, @RawRes **int** id){  
 InputStream inputStream = context.getResources().openRawResource(id);  
 ByteArrayOutputStream outputStream = **new** ByteArrayOutputStream();  
 **byte**[] buffer = **new byte**[1024];  
 **int** size;  
 **try** {  
 **while** ((size = inputStream.read(buffer)) != -1) {  
 outputStream.write(buffer, 0, size);  
 }  
 outputStream.close();  
 inputStream.close();  
 } **catch** (IOException e) {  
 System.***err***.println(**"File can not be read!"**);  
 e.printStackTrace();  
 }  
 **return** outputStream.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;  
 }  
}

**Висновок**

В даній лабораторній роботі було розроблено мобільний додаток, який показує список фільмів.

Програма працює коректно. Кінцева мета досягнута.