НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

з дисципліни

“Програмування мобільних систем”

Виконав:

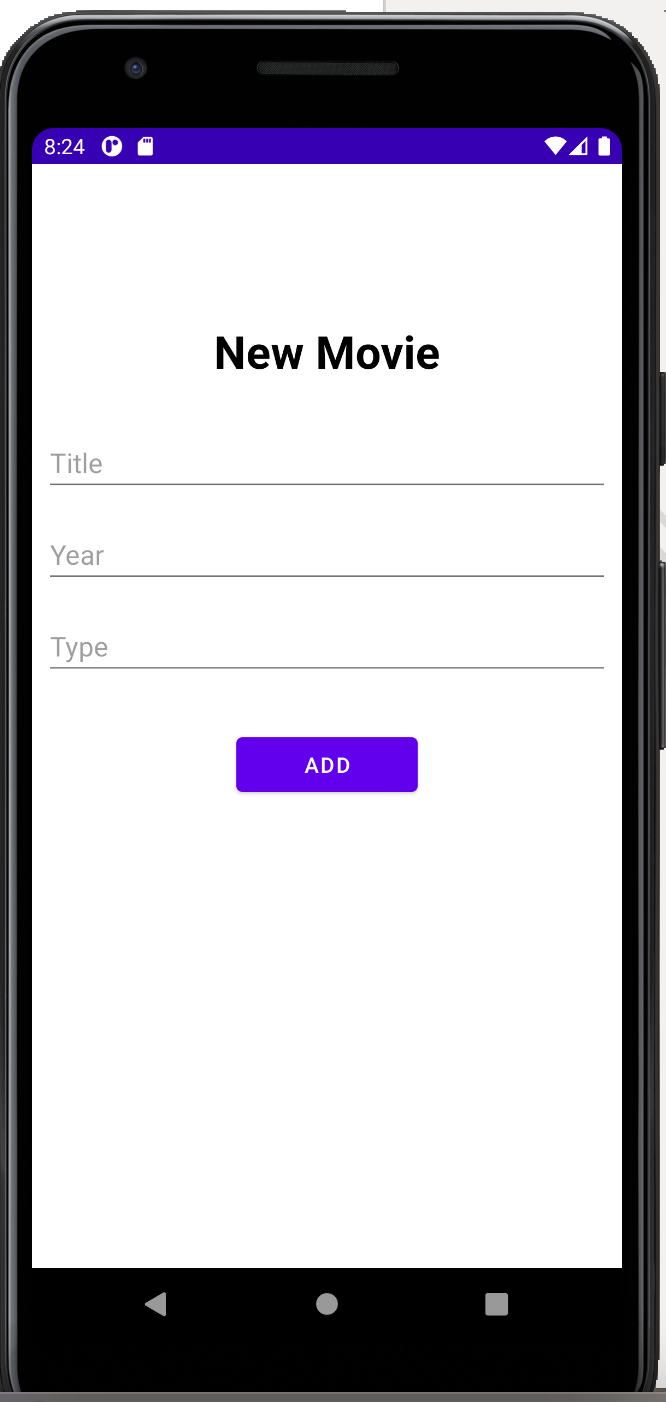
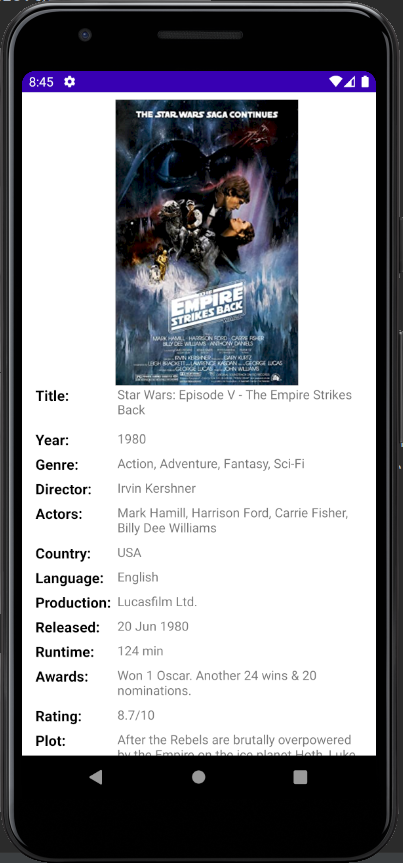
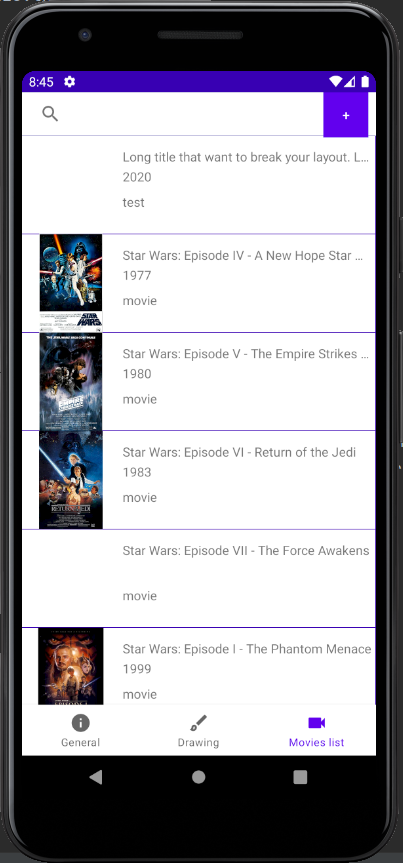
студент групи ІВ-82

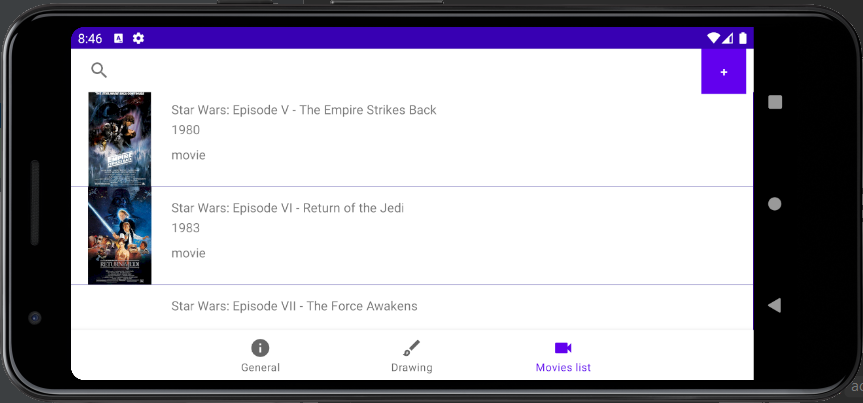
ЗК ІВ-8224

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***);  
 *// 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();  
 }  
 }  
}

**Movie.java**

**package** ua.kpi.comsys.iv8224.third.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.third.lab3;  
  
**import** ua.kpi.comsys.iv8224.third.R;  
  
**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;  
  
**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.third.lab3;  
  
**import** android.annotation.SuppressLint;  
**import** android.content.Context;  
**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.TextView;  
  
**import** androidx.annotation.RawRes;  
  
**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.lang.reflect.Field;  
  
**import** ua.kpi.comsys.iv8224.third.R;  
  
**public class** MovieInfo {  
 @SuppressLint(**"ClickableViewAccessibility"**)  
 **public void** showPopupWindow(**final** View view, Movie movie) {  
 view.getContext();  
 LayoutInflater inflater = (LayoutInflater) view.getContext().getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 @SuppressLint(**"InflateParams"**)  
 View popupView = inflater.inflate(R.layout.***movie\_info***, **null**);  
  
 **try** {  
 parseMovieInfo(*readTextFile*(popupView.getContext(),  
 *getResId*(movie.getImdbID(), R.raw.**class**)), movie);  
 } **catch** (ParseException e) {  
 System.***err***.println(**"Incorrect content of JSON file!"**);  
 e.printStackTrace();  
 }  
  
 **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);  
  
 **if** (movie.getPosterPath().length() != 0){  
 ((ImageView) popupView.findViewById(R.id.***movie\_info\_image***)).setImageResource(  
 *getResId*(movie.getPosterPath().toLowerCase()  
 .split(**"\\."**)[0], R.drawable.**class**));  
 }  
 ((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 void** parseMovieInfo(String jsonText, Movie movie) **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 static int** getResId(String resName, Class<?> c) {  
 **try** {  
 Field idField = c.getDeclaredField(resName);  
 **return** idField.getInt(idField);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 **return** -1;  
 }  
 }  
  
 **private 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 cannot be reading!"**);  
 e.printStackTrace();  
 }  
 **return** outputStream.toString();  
 }  
}

**MovieList.java**

**package** ua.kpi.comsys.iv8224.third.lab3;  
  
**import** android.content.Context;  
**import** android.graphics.Color;  
**import** android.text.TextUtils;  
**import** android.view.Gravity;  
**import** android.view.ViewGroup;  
**import** android.widget.ImageButton;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
  
**import** androidx.constraintlayout.widget.ConstraintLayout;  
**import** androidx.constraintlayout.widget.ConstraintSet;  
  
**import** com.daimajia.swipe.SwipeLayout;  
  
**import** java.lang.reflect.Field;  
  
**import** ua.kpi.comsys.iv8224.third.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);  
 }  
 });  
  
 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(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};  
 }  
  
 **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;  
 }  
 }  
}

**Висновок**

В даній лабораторній роботі було удосконалено мобільний додаток, а саме:

* пошук зі списку сутностей
* додавання своїх фільмів у бібліотеку
* новий екран, на якому відображається повна інформація про сутність

Програма працює коректно. Кінцева мета досягнута.