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 ІО-81

ЗК ІО-8114

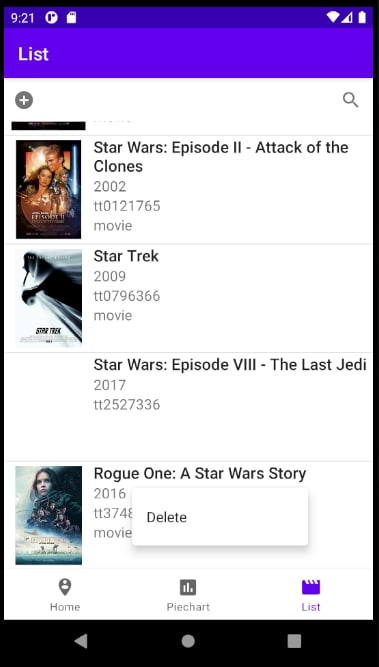
Кобук Назар

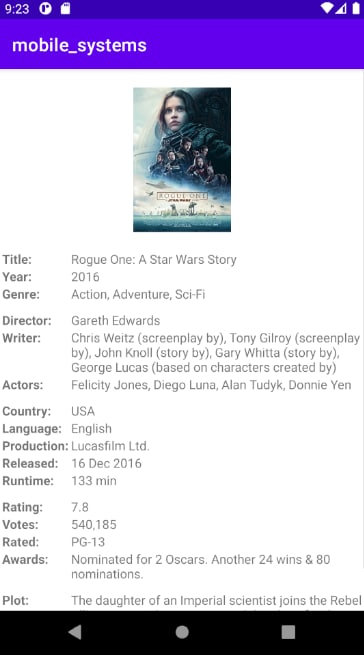
Київ 2021

**Посилання на репозиторій -** [**https://github.com/NazarKobuk/Mobile\_Systems\_1.1/tree/master/4**](https://github.com/NazarKobuk/Mobile_Systems_1.1/tree/master/4)

**Варіант № 1**

**Скріншот роботи додатка**





**Лістинг коду**

## **ListFragment.java**

package ua.kpi.comsys.io8114.ui.list;

import android.content.Intent;

import android.os.Bundle;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.Fragment;

import androidx.appcompat.widget.SearchView;

import android.view.ContextMenu;

import android.view.Gravity;

import android.view.LayoutInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ImageButton;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import com.google.gson.Gson;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.util.ArrayList;

import ua.kpi.comsys.io8114.R;

import ua.kpi.comsys.io8114.ui.list.Movie;

import ua.kpi.comsys.io8114.ui.list.MovieList;

public class ListFragment extends Fragment {

MovieList moviesList;

ListView moviesListView;

AdapterMoviesList adapterMoviesList;

SearchView searchView;

ArrayList<Movie> movies = new ArrayList<>();

TextView noResults;

ImageButton addMovieButton;

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

return inflater.inflate(R.layout.fragment\_list, container, false);

}

@Override

public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {

noResults = view.findViewById(R.id.textView\_noResults);

searchView = view.findViewById(R.id.search\_view);

moviesListView = (ListView) view.findViewById(R.id.moviesListView);

adapterMoviesList = new AdapterMoviesList(this.getContext(), movies);

addMovieButton = (ImageButton) view.findViewById(R.id.button\_addMovieButton);

moviesListView.setAdapter(adapterMoviesList);

// searchView.setImeOptions(searchView.getImeOptions() | EditorInfo.IME\_FLAG\_NO\_EXTRACT\_UI);

registerForContextMenu(moviesListView);

if (adapterMoviesList.getCount() == 0) {

parseFromJson("movies\_list");

movies.addAll(moviesList.getMovies());

}

addMovieButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(view.getContext(), MovieAdditionActivity.class);

startActivityForResult(intent, 1);

}

});

moviesListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) {

Movie item = moviesList.getMovieByPosition(position);

parseFromTxt(item.getImdbID(), item);

Intent intent = new Intent(view.getContext(), MovieDetailsActivity.class);

intent.putExtra("poster", item.getPoster());

intent.putExtra("title", item.getTitle());

intent.putExtra("year", item.getYear());

intent.putExtra("genre", item.getGenre());

intent.putExtra("director", item.getDirector());

intent.putExtra("writer", item.getWriter());

intent.putExtra("actors", item.getActors());

intent.putExtra("country", item.getCountry());

intent.putExtra("language", item.getLanguage());

intent.putExtra("production", item.getProduction());

intent.putExtra("released", item.getReleased());

intent.putExtra("runtime", item.getRuntime());

intent.putExtra("awards", item.getAwards());

intent.putExtra("rating", item.getImdbRating());

intent.putExtra("votes", item.getImdbVotes());

intent.putExtra("rated", item.getRated());

intent.putExtra("plot", item.getPlot());

startActivity(intent);

}

}

);

searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {

@Override

public boolean onQueryTextSubmit(String s) {

return false;

}

@Override

public boolean onQueryTextChange(String s) {

ArrayList<Movie> results = new ArrayList<>();

for (Movie movie: movies) {

if (movie.getTitle().toLowerCase().contains(s.toLowerCase()))

results.add(movie);

}

adapterMoviesList.update(results);

if (results.size() == 0) {

moviesListView.setVisibility(View.GONE);

noResults.setVisibility(View.VISIBLE);

} else {

noResults.setVisibility(View.GONE);

moviesListView.setVisibility(View.VISIBLE);

}

return true;

}

});

}

@Override

public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

if (data != null) {

String title = data.getStringExtra("title");

String year = String.valueOf(data.getIntExtra("year", 666));

String imdbId = data.getStringExtra("imdbId");

String type = data.getStringExtra("type");

Movie movie = new Movie(title, year, imdbId, type);

movie.setPoster("");

movies.add(movie);

moviesList.getMovies().add(movie);

adapterMoviesList.update(movies);

}

}

@Override

public void onCreateContextMenu(@NonNull ContextMenu menu, @NonNull View view,

@Nullable ContextMenu.ContextMenuInfo menuInfo) {

super.onCreateContextMenu(menu, view, menuInfo);

getActivity().getMenuInflater().inflate(R.menu.context\_menu\_movie, menu);

}

@Override

public boolean onContextItemSelected(@NonNull MenuItem item) {

AdapterView.AdapterContextMenuInfo menuInfo =

(AdapterView.AdapterContextMenuInfo) item.getMenuInfo();

int selectedItemIndex = (int) adapterMoviesList.getItemId(menuInfo.position);

switch (item.getItemId()) {

case R.id.option\_delete:

String title = moviesList.getMovies().get(selectedItemIndex).getTitle();

movies.remove(selectedItemIndex);

moviesList.getMovies().remove(selectedItemIndex);

adapterMoviesList.update(movies);

Toast toast = Toast.makeText(getContext(), title + "\nhas been deleted",

Toast.LENGTH\_SHORT);

toast.show();;

return true;

default:

return super.onContextItemSelected(item);

}

}

//Parses JSON file with primary film characteristics.

private void parseFromJson(String fileName) {

Gson gson = new Gson();

BufferedReader br = null;

try {

br = new BufferedReader(new InputStreamReader(getFileLocation(fileName)));

moviesList = gson.fromJson(br, MovieList.class);

} catch (Exception e) {

e.printStackTrace();

} finally {

if (br != null) {

try {

br.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

// Parses TXT file with additional film characteristics.

private void parseFromTxt(String fileName, Movie movie) {

Gson gson = new Gson();

BufferedReader br = null;

try {

br = new BufferedReader(new InputStreamReader(getFileLocation(fileName)));

Movie parsedMovie = gson.fromJson(br, Movie.class);

movie.setTitle(parsedMovie.getTitle());

movie.setYear(parsedMovie.getYear());

movie.setGenre(parsedMovie.getGenre());

movie.setDirector(parsedMovie.getDirector());

movie.setWriter(parsedMovie.getWriter());

movie.setActors(parsedMovie.getActors());

movie.setCountry(parsedMovie.getCountry());

movie.setLanguage(parsedMovie.getLanguage());

movie.setProduction(parsedMovie.getProduction());

movie.setReleased(parsedMovie.getReleased());

movie.setRuntime(parsedMovie.getRuntime());

movie.setAwards(parsedMovie.getAwards());

movie.setImdbRating(parsedMovie.getImdbRating());

movie.setImdbVotes(parsedMovie.getImdbVotes());

movie.setRated(parsedMovie.getRated());

movie.setPlot(parsedMovie.getPlot());

} catch (Exception e) {

e.printStackTrace();

} finally {

if (br != null) {

try {

br.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

private InputStream getFileLocation(String fileName) {

return getResources().openRawResource(getResources().getIdentifier(fileName,

"raw", this.getContext().getPackageName()));

}

}

**Висновок**

## В результаті виконання лаборатоної було додано новий функціонал у список фільмів у Android додатку. Було доповнено клас **ListFragment.java** **в якому виконано завдання за варіантом, а також налаштовано роботу додатка для коректного відображення його вмісту.**