НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота № 5

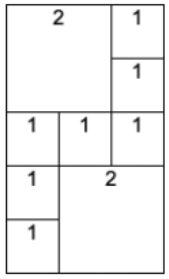
з дисципліни

“Програмування мобільних систем”

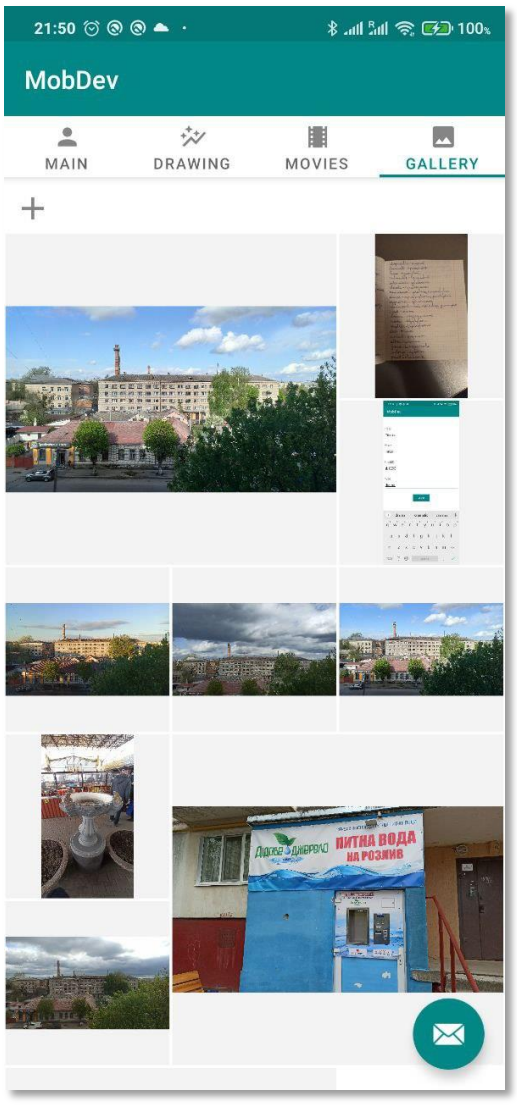
Виконав:
студент групи ІО-82
ЗК ІО-8226
Шевчук Олександр

Київ 2021

Варіант 1
Предметна область – фільми Архів для завантаження

Варіант 1


Скріншот роботи додатку



Лістинг коду

FragmentGallery.java

```
package ua.kpi.compsys.io8226.tabs.tab_gallery;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ScrollView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.constraintlayout.widget.ConstraintSet;
import androidx.constraintlayout.widget.Guideline;
import androidx.fragment.app.Fragment;

import java.util.ArrayList;

import ua.kpi.compsys.io8226.R;


public class FragmentGallery extends Fragment {

    private static final int RESULT_LOAD_IMAGE = 2;

    private ArrayList<ImageView> imageViews;
    private ArrayList<ArrayList<Object>> placeholders;
    private ScrollView scrollView;
    private LinearLayout linearLayout;
    private View view;


    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {

        view = inflater.inflate(R.layout.fragment_gallery, container, false);
        setRetainInstance(true);

        scrollView = view.findViewById(R.id.scrollview_gallery);
        linearLayout = view.findViewById(R.id.linear_main);

        imageViews = new ArrayList<>();
        placeholders = new ArrayList<>();

        ImageButton btnAddImage = view.findViewById(R.id.btn_add_image);
        btnAddImage.setOnClickListener(v -> {
            Intent gallery = new Intent(Intent.ACTION_GET_CONTENT);
```

```
            gallery.setType("image/*");
            startActivityForResult(gallery, RESULT_LOAD_IMAGE);
        });

        for (ImageView image : imageViews) {
            image.setScaleType(ImageView.ScaleType.CENTER_INSIDE);
        }

        return view;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
    }


    @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();
            putImage(linearLayout, imageViews, placeholders, scrollView,
imageUri);
        }
    }


    private Guideline createGuideline(int orientation, float percent){
        Guideline guideline = new Guideline(view.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 void putImage(LinearLayout scrollMain, ArrayList<ImageView> allImages,
                          ArrayList<ArrayList<Object>> placeholderList,
                          ScrollView scrollView, Uri imageUri) {

        ImageView newImage = new ImageView(view.getContext());
        newImage.setImageURI(imageUri);
        newImage.setBackgroundResource(R.color.image_background);

        ConstraintLayout.LayoutParams imageParams =
                new
ConstraintLayout.LayoutParams(ConstraintLayout.LayoutParams.MATCH_CONSTRAINT,
                        ConstraintLayout.LayoutParams.MATCH_CONSTRAINT);
```

```
        imageParams.setMargins(3, 3, 3, 3);
        imageParams.dimensionRatio = "1";
        newImage.setLayoutParams(imageParams);
        newImage.setId(newImage.hashCode());

        ConstraintLayout tmpLayout = null;
        ConstraintSet tmpSet = null;
        if (allImages.size() > 0) {
            tmpLayout = (ConstraintLayout) getConstraint(0, placeholderList);
            tmpSet = (ConstraintSet) getConstraint(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);
        }

        if (allImages.size() % 9 != 0)
            tmpLayout.addView(newImage);

        switch (allImages.size() % 9){
            case 0:{
                placeholderList.add(new ArrayList<>());

                ConstraintLayout newConstraint = new
ConstraintLayout(view.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_33 =
createGuideline(ConstraintLayout.LayoutParams.VERTICAL,
                        0.333333f);
                Guideline vertical_66 =
createGuideline(ConstraintLayout.LayoutParams.VERTICAL,
                        0.666666f);

                Guideline horizontal_20 =
createGuideline(ConstraintLayout.LayoutParams.HORIZONTAL,
                        0.2f);
                Guideline horizontal_40 =
createGuideline(ConstraintLayout.LayoutParams.HORIZONTAL,
                        0.4f);
                Guideline horizontal_60 =
createGuideline(ConstraintLayout.LayoutParams.HORIZONTAL,
                        0.6f);
                Guideline horizontal_80 =
createGuideline(ConstraintLayout.LayoutParams.HORIZONTAL,
                        0.8f);

                newConstraint.addView(vertical_33, 0);
                newConstraint.addView(vertical_66, 1);
                newConstraint.addView(horizontal_20, 2);
                newConstraint.addView(horizontal_40, 3);
```

```
                newConstraint.addView(horizontal_60, 4);
                newConstraint.addView(horizontal_80, 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_66.getId(), ConstraintSet.START);
                newConstraintSet.connect(newImage.getId(), ConstraintSet.BOTTOM,
                        horizontal_40.getId(), ConstraintSet.TOP);

                newConstraintSet.applyTo(newConstraint);
                break;
            }

            case 1: {
                tmpSet.connect(newImage.getId(), ConstraintSet.START,
                        tmpLayout.getChildAt(1).getId(), ConstraintSet.START);
                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(2).getId(), ConstraintSet.TOP);

                tmpSet.applyTo(tmpLayout);
                break;
            }

            case 2: {
                tmpSet.connect(newImage.getId(), ConstraintSet.START,
                        tmpLayout.getChildAt(1).getId(), ConstraintSet.START);
                tmpSet.connect(newImage.getId(), ConstraintSet.TOP,
                        tmpLayout.getChildAt(2).getId(), 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.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.connect(newImage.getId(), ConstraintSet.START,
            tmpLayout.getChildAt(0).getId(), ConstraintSet.START);
    tmpSet.connect(newImage.getId(), ConstraintSet.TOP,
            tmpLayout.getChildAt(3).getId(), ConstraintSet.BOTTOM);
    tmpSet.connect(newImage.getId(), ConstraintSet.END,
            tmpLayout.getChildAt(1).getId(), ConstraintSet.START);
    tmpSet.connect(newImage.getId(), ConstraintSet.BOTTOM,
            tmpLayout.getChildAt(4).getId(), ConstraintSet.TOP);

    tmpSet.applyTo(tmpLayout);
    break;
}

case 5: {
    tmpSet.connect(newImage.getId(), ConstraintSet.START,
            tmpLayout.getChildAt(1).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 6: {
    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(0).getId(), ConstraintSet.START);
    tmpSet.connect(newImage.getId(), ConstraintSet.BOTTOM,
            tmpLayout.getChildAt(5).getId(), ConstraintSet.TOP);

    tmpSet.applyTo(tmpLayout);
    break;
}

case 7: {
    tmpSet.connect(newImage.getId(), ConstraintSet.START,
            tmpLayout.getChildAt(0).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,
            ConstraintSet.PARENT_ID, ConstraintSet.BOTTOM);

    tmpSet.applyTo(tmpLayout);
```

```
                break;
            }

            case 8: {
                tmpSet.connect(newImage.getId(), ConstraintSet.START,
                        ConstraintSet.PARENT_ID, ConstraintSet.START);
                tmpSet.connect(newImage.getId(), ConstraintSet.TOP,
                        tmpLayout.getChildAt(5).getId(), ConstraintSet.BOTTOM);
                tmpSet.connect(newImage.getId(), ConstraintSet.END,
                        tmpLayout.getChildAt(0).getId(), ConstraintSet.START);
                tmpSet.connect(newImage.getId(), ConstraintSet.BOTTOM,
                        ConstraintSet.PARENT_ID, ConstraintSet.BOTTOM);

                tmpSet.applyTo(tmpLayout);
                break;
            }
        }

        allImages.add(newImage);
        scrollView.post(() -> scrollView.fullScroll(View.FOCUS_DOWN));
    }

    private Object getConstraint(int index, ArrayList<ArrayList<Object>> list){
        return list.get(list.size()-1).get(index);
    }
}
```

Висновок

В результаті виконання лабораторної я додав нову вкладку із функціоналом контейнера для зображень, які обирає користувач із файлів на пристрої. Контейнер організовує у стилі, вказаному за варіантом.