

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

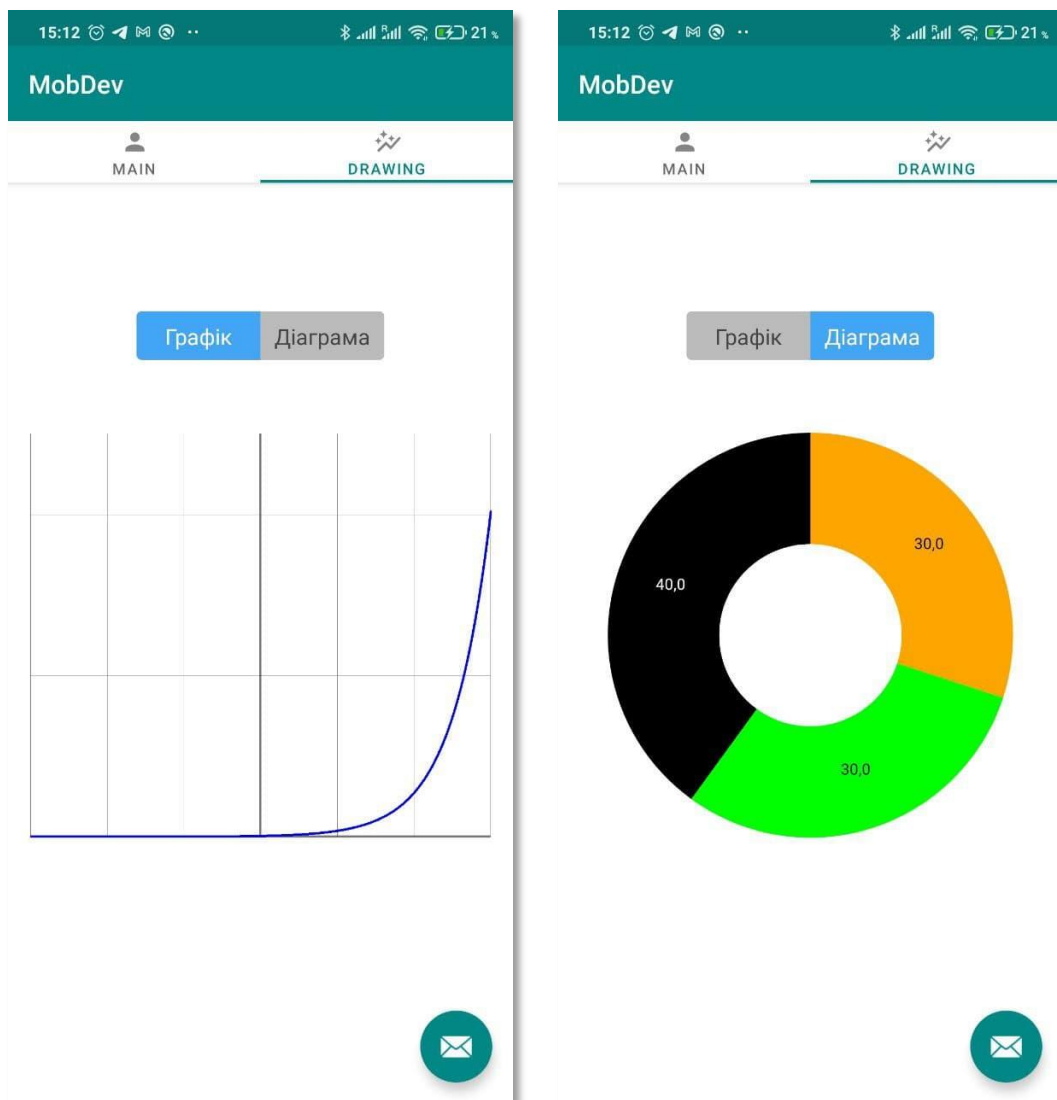
Лабораторна робота № 2
з дисципліни
“Програмування мобільних систем”

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

Київ 2021

Варіант 1
<p>а. Функція $y = e^x$ на проміжку $x \in [-6; 6]$;</p> <p>б. Кільце кругової діаграми із секторами, що займають відповідний відсоток кола та мають відповідний колір: 30% (помаранчевий), 30% (зелений), 40% (чорний).</p>

Скріншоти роботи додатку



Лістинг коду

FragmentSecond.java

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

import android.graphics.Color;
import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.github.mikephil.charting.animation.Easing;
import com.github.mikephil.charting.charts.PieChart;
import com.github.mikephil.charting.data.PieData;
import com.github.mikephil.charting.data.PieDataSet;
import com.github.mikephil.charting.data.PieEntry;
import com.jjoe64.graphview.GraphView;
import com.jjoe64.graphview.series.DataPoint;
import com.jjoe64.graphview.series.LineGraphSeries;
import com.lolollox.androidtoggleswitch.widgets.ToggleSwitch;

import java.util.ArrayList;
import java.util.List;

public class FragmentSecond extends Fragment {
    private GraphView coordPlot;
    private PieChart pieChart;
    private ToggleSwitch toggleSwitch;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_second, container, false);
    }

    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        coordPlot = (GraphView) view.findViewById(R.id.coordPlot);
        pieChart = (PieChart) view.findViewById(R.id.pieChart);
        toggleSwitch = (ToggleSwitch) view.findViewById(R.id.toggleGraphs);
        pieChart.setUsePercentValues(true);

        toggleSwitch.setOnChangeListener(new ToggleSwitch.OnChangeListener() {
            @Override
            public void onToggleSwitchChanged(int position) {
                if (position == 0) {
                    drawPlot();
                } else {
                    drawCircDiagram();
                }
            }
        });
    }
}
```

```

        toggleSwitch.setCheckedPosition(0);
        drawPlot();
    }

    public void drawPlot() {
        LineGraphSeries<DataPoint> series = new LineGraphSeries<>();
        pieChart.setVisibility(View.INVISIBLE);
        coordPlot.setVisibility(View.VISIBLE);
        double x, y;
        x = -6;
        int points = 1200;

        for (int i = 0; i < points; i++) {
            x += 0.01;
            y = Math.pow(Math.E, x);
            series.appendData(new DataPoint(x, y), true, 1200);
        }
        series.setAnimated(true);
        series.setColor(Color.BLUE);
        coordPlot.removeAllSeries();
        coordPlot.addSeries(series);

        coordPlot.getViewport().setXAxisBoundsManual(true);
        coordPlot.getViewport().setMinX(-6);
        coordPlot.getViewport().setMaxX(6);

        coordPlot.getViewport().setYAxisBoundsManual(true);
        coordPlot.getViewport().setMinY(0);
        coordPlot.getViewport().setMaxY(500);

        coordPlot.getViewport().setScalable(true);
        coordPlot.getViewport().setScalableY(true);

        coordPlot.getGridLabelRenderer().setHorizontalLabelsVisible(false);
        coordPlot.getGridLabelRenderer().setVerticalLabelsVisible(false);
    }

    public void drawCircDiagram() {
        coordPlot.setVisibility(View.INVISIBLE);
        pieChart.setVisibility(View.VISIBLE);

        List<PieEntry> values = new ArrayList<>();

        values.add(new PieEntry(30f));
        values.add(new PieEntry(30f));
        values.add(new PieEntry(40f));

        PieDataSet pieDataSet = new PieDataSet(values, "Pie Chart");

        final int[] MY_COLORS = {Color.rgb(255,165,0),
            Color.rgb(0,255,0), Color.rgb(0,0,0)};
        final int[] TEXT_COLORS = {Color.rgb(0,0,0),
            Color.rgb(0,0,0), Color.rgb(255,255,255)};

        ArrayList<Integer> colors = new ArrayList<>();
        ArrayList<Integer> textColors = new ArrayList<>();

        for(int c: MY_COLORS) {

```

```

        colors.add(c);
    }

    for (int c: TEXT_COLORS) {
        textColors.add(c);
    }

    pieDataSet.setValueTextColors(textColors);
    pieDataSet.setColors(colors);

    PieData pieData = new PieData(pieDataSet);

    pieChart.setData(pieData);
    pieChart.getDescription().setEnabled(false);
    pieData.setValueTextSize(12);
    pieChart.getLegend().setEnabled(false);
    pieChart.setHoleRadius(45f);
    pieChart.setTransparentCircleRadius(0f);
    pieChart.animateY(1000, Easing.EaseInOutExpo);
    }
}

```

Висновок

В результаті виконання лабораторної я додав нові елементи у другу вкладку додатку, налаштував їх відповідно до варіанту.