МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**ЗВІТ**

**з лабораторної роботи № 2**

**«СТВОРЕННЯ ОБРОБНИКІВ ДЛЯ ЕЛЕМЕНТІВ УПРАВЛІННЯ У ANDROID-ЗАСТОСУНКАХ. РОЗМІЩЕННЯ ВІДЖЕТІВ У МОБІЛЬНИХ ЗАСТОСУНКАХ НА ГОЛОВНОМУ ЕКРАНІ. РОБОТА ЗІ СТИЛЯМИ»**

**Виконав:**

студент 3-го курсу, групи КП-91,

спеціальності 121 – Інженерія

програмного забезпечення

*Маховой Олександр Вікторович*

**Перевірив:**

к. т. н, старший викладач

*Хайдуров Владислав Володимирович*

Київ – 2021

**Зміст**

[**Вступ**](#_53ul1mtli2ue) **3**

[**Постановка задачі**](#_w61nz6xb5bez) **4**

[**Хід виконання**](#_z6nxf7bftfw9) **8**

[Функціональна частина для простого калькулятора](#_og7oix7gnpbr) 8

[Аналогічний калькулятор](#_lmifex749jpq) 11

[Гра хрестики-нулики](#_j82h5aynk1vt) 15

[Перший застосунок](#_j6qsjn6kdn01) 22

[Другий застосунок](#_bg4fuu94g2zg) 25

[Третій застосунок](#_26c8yf86yxs9) 26

[**Відповіді на контрольні запитання**](#_fab01o3ie38e) **30**

[Які елементи управляння / віджети використані у Вашій роботи? Вкажіть переваги використання одного обробника для кількох елементів управління у створеному Android-застосунку.](#_7520dhlnkn46) 30

[Навести приклад обробника для класичного об’єкта Button.](#_2y9grwcqi4mr) 30

[**Висновки**](#_lhzcig617h5u) **31**

[**Використані джерела**](#_r1odn3quodej) **32**

# Вступ

В мобільному застосунку для інтерактивної взаємодії з користувачем необхідно обробляти події натискання елементів UI. Саме цю задачу виконує механізм підписки обробників на події. Окрім цього, для кастомізації UI Android надає можливість налаштування стилів. В ході даної лабораторної роботи і буде використано ці можливості Android.

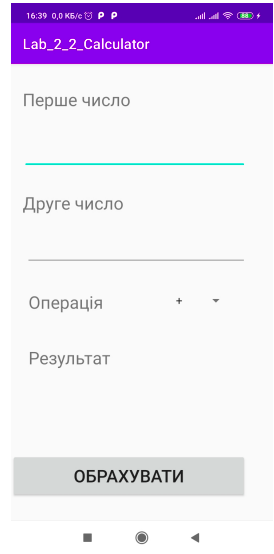
# Постановка задачі

1. Ознайомитись із усіма теоретичними відомостями до лабораторної роботи.

2. Розробити функціональну частину для простого калькулятора, інтерфейс якого був розроблений у попередній лабораторній роботі. Для кожної кнопки створити власний обробник подій.

3. Створити копію проекту, який повністю реалізовано у пункті 2. Замінити обробник подій для кожної кнопки одним обробником для масиву кнопок, що мають текстові надписи 0, 1, 2, ..., 9. Обробник виділити окремим методом (функцією). У звіт додати програмний код та результати роботи Android-програми.

4. Створити аналогічний калькулятор, що має два текстові поля, у які користувач самостійно вводить два числа. Для вибору операцій (+, -, \*, /) а головному вікні (Main\_Activity) створити Spinner. Для виконання обрахунків додати до інтерфейсу кнопку (Button). Інтерфейс має бути подібний до такого:



У звіт додати програмний код та результати роботи Android-програми.

5. За аналогією до пункту 2 створити простий проект гру хрестики-нулики для поля розміром 4х4. Кожна клітинка повинна мати список Spinner. У звіт додати програмний код та результати роботи Android-програми.

6. Розробити три мобільних застосунки, кожен з яких складається з одного activity. У першому застосунку має бути представлений інтерфейс, що складається з восьми кнопок (рис. 37). Відстань між кнопками – 2%. Перша і остання кнопка повинні бути на рівній відстані від країв екрану.

7. Інтерфейс другого програмного додатку на рис. 38. Верстка повинна здійснюватися з використанням RelativeLayout (не використовувати LinearLayout).



Рис. 37. Графічний інтерфейс першого мобільного застосунка



Рис. 38. Графічний інтерфейс другого мобільного застосунка



Рис. 39. Графічний інтерфейс третього мобільного застосунка

8. Третій додаток (рис. 39) повинен у інтерфейсі кнопку, яка повинна бути вирівняна по центру екрана. Колір контуру кнопки #505050. Товщину контуру обрати за бажанням від 5 до 12. Радіус округлений 24dp. Колір фону екрану #FFFFFF. При натисканні на кнопку її колір повинен змінюватися на світло-зелений. Висота кнопки повинна складати 20% від висоти екрана.

9. Усі розмітки додати у звіт.

10.Оформити звіт до даної лабораторної роботи.

# Хід виконання

## Функціональна частина для простого калькулятора

Скріншот роботи:

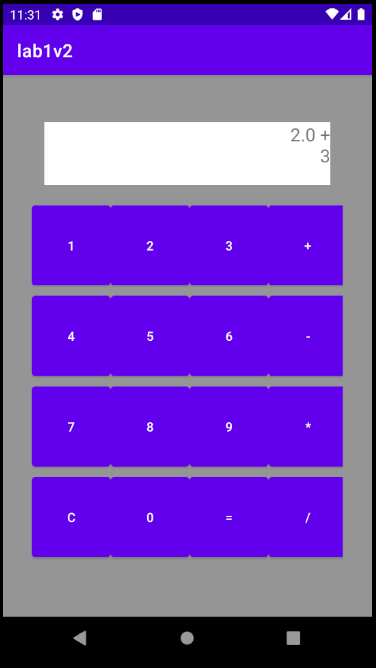


Рис.1 Робота програми на емуляторі.

Приклади коду:

|  |
| --- |
| **MainActivity.java** |
| **package com.example.lab1v2;**  **import androidx.appcompat.app.AppCompatActivity;**  **import android.view.View;**  **import android.widget.Button;**  **import android.os.Bundle;**  **import android.widget.TextView;**  **import java.net.Inet4Address;**  **public class MainActivity extends AppCompatActivity {**  **TextView screen;**  **String prev="", temp="";**  **Character oper=' ';**  **@Override**  **public void onCreate(Bundle savedInstanceState) {**  **System.out.println("---------------start-----------------");**  **super.onCreate(savedInstanceState);**  **setContentView(R.layout.activity\_main);**  **screen = findViewById(R.id.screen);**  **screen.setText(prev + ' ' + oper + "\n ");**  **setListenersOnNumbers();**  **setListenersOnOperators();**  **setListenerOnClear();**  **setListenerOnEquals();**  **}**  **private void setListenerOnEquals() {**  **findViewById(R.id.buttonEq).setOnClickListener(new View.OnClickListener() {**  **@Override**  **public void onClick(View v) {**  **onEqualsClick(v);**  **}**  **});**  **}**  **private void setListenersOnOperators(){**  **View.OnClickListener onOperatorClickHandler = new View.OnClickListener() {**  **@Override**  **public void onClick(View v) {**  **onOperatorClick(v);**  **}**  **};**  **findViewById(R.id.buttonPlus).setOnClickListener(onOperatorClickHandler);**  **findViewById(R.id.buttonMinus).setOnClickListener(onOperatorClickHandler);**  **findViewById(R.id.buttonMultiply).setOnClickListener(onOperatorClickHandler);**  **findViewById(R.id.buttonDiv).setOnClickListener(onOperatorClickHandler);**  **}**  **private void setListenersOnNumbers(){**  **View.OnClickListener onNumberClickHandler = new View.OnClickListener() {**  **@Override**  **public void onClick(View v) {**  **onNumberClick(v);**  **}**  **};**  **findViewById(R.id.button0).setOnClickListener(onNumberClickHandler);**  **findViewById(R.id.button1).setOnClickListener(onNumberClickHandler);**  **findViewById(R.id.button2).setOnClickListener(onNumberClickHandler);**  **findViewById(R.id.button3).setOnClickListener(onNumberClickHandler);**  **findViewById(R.id.button4).setOnClickListener(onNumberClickHandler);**  **findViewById(R.id.button5).setOnClickListener(onNumberClickHandler);**  **findViewById(R.id.button6).setOnClickListener(onNumberClickHandler);**  **findViewById(R.id.button7).setOnClickListener(onNumberClickHandler);**  **findViewById(R.id.button8).setOnClickListener(onNumberClickHandler);**  **findViewById(R.id.button9).setOnClickListener(onNumberClickHandler);**  **}**  **private float countResult() throws Exception {**  **switch (oper){**  **case '+':**  **return Float.parseFloat(prev) + Float.parseFloat(temp);**  **case '-':**  **return Float.parseFloat(prev) - Float.parseFloat(temp);**  **case '\*':**  **return Float.parseFloat(prev) \* Float.parseFloat(temp);**  **case '/':**  **float t = Float.parseFloat(temp);**  **if (t == 0) throw new Exception("Dividing by zero");**  **else return Float.parseFloat(prev) / t;**  **}**  **return Float.parseFloat(temp);**  **}**  **private void setListenerOnClear() {**  **findViewById(R.id.buttonC).setOnClickListener(new View.OnClickListener() {**  **@Override**  **public void onClick(View v) {**  **onClearClick(v);**  **}**  **});**  **}**  **public void onEqualsClick(View view){**  **if (oper==' ' && temp=="") return;**  **if (oper == ' ') {**  **prev = "0";**  **oper = '+';**  **}**  **try {**  **if (temp!="") prev = Float.toString(countResult());**  **}**  **catch (Exception err) {**  **}**  **temp = prev;**  **oper = ' ';**  **prev = "";**  **screen.setText(prev +' ' +oper+ '\n' + temp);**  **}**  **public void onClearClick(View view){**  **if (temp.length()<1) return;**  **temp = new StringBuffer(temp).delete(temp.length()-1, temp.length()).toString();**  **screen.setText(prev +' ' +oper+ '\n' + temp);**  **}**  **public void onOperatorClick(View view) {**  **if (oper==' ' && temp=="") return;**  **Button but = (Button) findViewById(view.getId());**  **if (oper == ' ') {**  **prev = "0";**  **oper = '+';**  **}**  **try {**  **if (temp!="") prev = Float.toString(countResult());**  **}**  **catch (Exception err) {**  **}**  **oper = but.getText().charAt(0);**  **temp = "";**  **screen.setText(prev +' ' +oper+ '\n' + temp);**  **}**  **public void onNumberClick(View view) {**  **if (temp.length()<8) {**  **Button but = (Button) findViewById(view.getId());**  **//if (temp.length()>1)**  **if (temp=="" || temp == "0") temp = but.getText().toString();**  **else temp += but.getText().toString();**  **//else temp = but.getText().toString();**  **screen.setText(prev +' ' +oper+ '\n' + temp);**  **}**  **}**  **}** |

## Аналогічний калькулятор

Скріншот роботи:

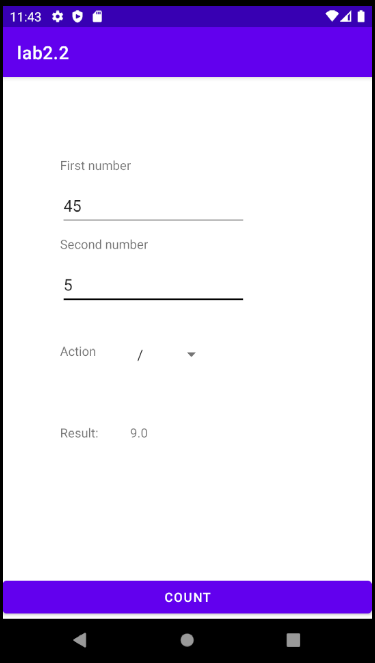


Рис.2 Робота програми на емуляторі.

Приклади коду:

|  |
| --- |
| **MainActivity.java** |
| **package com.example.lab22;**  **import androidx.appcompat.app.AppCompatActivity;**  **import android.os.Bundle;**  **import android.view.View;**  **import android.widget.EditText;**  **//import android.graphics.Color;**  **import android.graphics.PorterDuff;**  **import android.widget.Spinner;**  **import android.widget.TextView;**  **public class MainActivity extends AppCompatActivity {**  **EditText number1;**  **EditText number2;**  **Spinner oper;**  **TextView res;**  **@Override**  **protected void onCreate(Bundle savedInstanceState) {**  **super.onCreate(savedInstanceState);**  **setContentView(R.layout.activity\_main);**  **number1 = (EditText) findViewById(R.id.number1);**  **number2 = (EditText) findViewById(R.id.number2);**  **oper = (Spinner) findViewById(R.id.spinner2);**  **res = (TextView) findViewById(R.id.result);**  **number1.setOnClickListener(**  **new View.OnClickListener() {**  **@Override**  **public void onClick(View view) {**  **number1.getBackground().mutate().setColorFilter(getResources().getColor(android.R.color.black), PorterDuff.Mode.SRC\_ATOP);**  **}**  **}**  **);**  **number2.setOnClickListener(**  **new View.OnClickListener() {**  **@Override**  **public void onClick(View view) {**  **number2.getBackground().mutate().setColorFilter(getResources().getColor(android.R.color.black), PorterDuff.Mode.SRC\_ATOP);**  **}**  **}**  **);**  **findViewById(R.id.buttonStart).setOnClickListener(**  **new View.OnClickListener() {**  **@Override**  **public void onClick(View view) {**  **count();**  **}**  **}**  **);**  **}**  **private void count(){**  **String num1 = number1.getText().toString();**  **String num2 = number2.getText().toString();**  **boolean f=false;**  **if (num1.length()<1) {**  **number1.getBackground().mutate().setColorFilter(getResources().getColor(android.R.color.holo\_red\_light), PorterDuff.Mode.SRC\_ATOP);**  **f = true;**  **}**  **if (num2.length()<1) {**  **number2.getBackground().mutate().setColorFilter(getResources().getColor(android.R.color.holo\_red\_light), PorterDuff.Mode.SRC\_ATOP);**  **f = f || true;**  **}**  **if (f) return;**  **float a = Float.parseFloat(num1);**  **float b = Float.parseFloat(num2);**  **float r;**  **try {**  **switch (oper.getSelectedItem().toString().charAt(0)) {**  **case '+': {**  **r = a + b;**  **break;**  **}**  **case '-': {**  **r = a - b;**  **break;**  **}**  **case '\*': {**  **r = a \* b;**  **break;**  **}**  **case '/': {**  **if (b == 0) throw new Exception("Dividing by zero");**  **r = a / b;**  **break;**  **}**  **default: throw new Exception("Unknown operator");**  **}**  **res.setText(String.valueOf(r));**  **} catch (Exception err) {**  **System.out.println(err.toString());**  **number2.getBackground().mutate().setColorFilter(getResources().getColor(android.R.color.holo\_red\_light), PorterDuff.Mode.SRC\_ATOP);**  **}**  **}**  **}** |

|  |
| --- |
| **strings.xml** |
| <resources>  <string name="app\_name">lab2.2</string>  <string-array name="operators">  <item>+</item>  <item>-</item>  <item>\*</item>  <item>/</item>  </string-array>  </resources> |

|  |
| --- |
| **themes.xml** |
| <resources xmlns:tools="http://schemas.android.com/tools">  <!-- Base application theme. -->  <style name="Theme.Lab22" parent="Theme.MaterialComponents.DayNight.DarkActionBar">  <!-- Primary brand color. -->  <item name="colorPrimary">@color/purple\_500</item>  <item name="colorPrimaryVariant">@color/purple\_700</item>  <item name="colorOnPrimary">@color/white</item>  <!-- Secondary brand color. -->  <item name="colorSecondary">@color/teal\_200</item>  <item name="colorSecondaryVariant">@color/teal\_700</item>  <item name="colorOnSecondary">@color/black</item>  <!-- Status bar color. -->  <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>  <!-- Customize your theme here. -->  </style>  </resources> |

|  |
| --- |
| **activity main.xml** |
| **<?xml version="1.0" encoding="utf-8"?>**  **<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:app="http://schemas.android.com/apk/res-auto"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **tools:context=".MainActivity">**  **<TextView**  **android:id="@+id/textView2"**  **android:layout\_width="202dp"**  **android:layout\_height="27dp"**  **android:layout\_marginStart="64dp"**  **android:layout\_marginTop="176dp"**  **android:text="Second number"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toTopOf="parent" />**  **<EditText**  **android:id="@+id/number2"**  **android:layout\_width="208dp"**  **android:layout\_height="48dp"**  **android:layout\_marginStart="64dp"**  **android:layout\_marginTop="92dp"**  **android:ems="10"**  **android:inputType="number"**  **android:maxLength="10"**  **android:maxLines="1"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toBottomOf="@+id/textView" />**  **<TextView**  **android:id="@+id/textView"**  **android:layout\_width="202dp"**  **android:layout\_height="27dp"**  **android:layout\_marginStart="64dp"**  **android:layout\_marginTop="88dp"**  **android:text="First number"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toTopOf="parent" />**  **<EditText**  **android:id="@+id/number1"**  **android:layout\_width="208dp"**  **android:layout\_height="48dp"**  **android:layout\_marginStart="64dp"**  **android:layout\_marginTop="4dp"**  **android:ems="10"**  **android:inputType="number"**  **android:maxLength="10"**  **android:maxLines="1"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toBottomOf="@+id/textView" />**  **<TextView**  **android:id="@+id/textView3"**  **android:layout\_width="46dp"**  **android:layout\_height="27dp"**  **android:layout\_marginStart="64dp"**  **android:layout\_marginTop="40dp"**  **android:text="Action"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toBottomOf="@+id/number2" />**  **<Spinner**  **android:id="@+id/spinner2"**  **android:layout\_width="92dp"**  **android:layout\_height="27dp"**  **android:layout\_marginStart="32dp"**  **android:layout\_marginTop="40dp"**  **app:layout\_constraintStart\_toEndOf="@+id/textView3"**  **app:layout\_constraintTop\_toBottomOf="@+id/number2"**  **android:entries="@array/operators"/>**  **<TextView**  **android:id="@+id/textResult"**  **android:layout\_width="46dp"**  **android:layout\_height="27dp"**  **android:layout\_marginStart="64dp"**  **android:layout\_marginTop="64dp"**  **android:text="Result:"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toBottomOf="@+id/textView3" />**  **<TextView**  **android:id="@+id/result"**  **android:layout\_width="92dp"**  **android:layout\_height="27dp"**  **android:layout\_marginStart="32dp"**  **android:layout\_marginTop="64dp"**  **android:text="sasaas"**  **app:layout\_constraintStart\_toEndOf="@+id/textResult"**  **app:layout\_constraintTop\_toBottomOf="@+id/spinner2" />**  **<Button**  **android:id="@+id/buttonStart"**  **android:layout\_width="match\_parent"**  **android:layout\_height="0dp"**  **android:text="count"**  **app:layout\_constraintBottom\_toBottomOf="parent"**  **app:layout\_constraintEnd\_toEndOf="parent"**  **app:layout\_constraintStart\_toStartOf="parent" />**  **</androidx.constraintlayout.widget.ConstraintLayout>** |

## 

## Гра хрестики-нулики

Скріншот роботи:

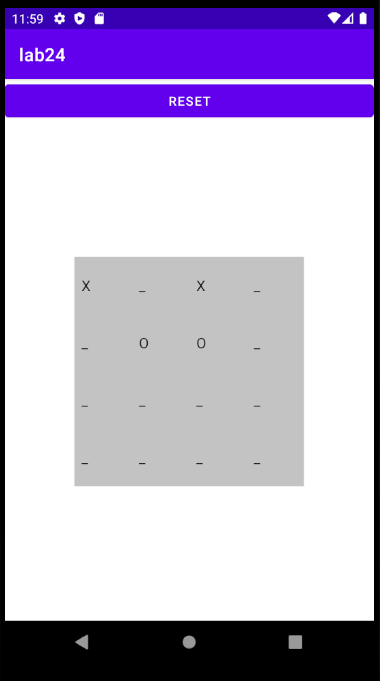


Рис.3 Робота програми на емуляторі.

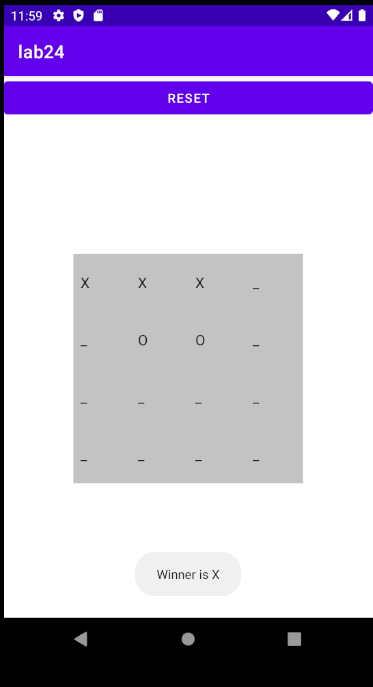


Рис.3 Робота програми на емуляторі (перемога одного з гравців).

Приклади коду:

|  |
| --- |
| **MainActivity.java** |
| **package com.example.lab24;**  **import androidx.appcompat.app.AppCompatActivity;**  **import android.os.Bundle;**  **import android.view.View;**  **import android.widget.AdapterView;**  **import android.widget.ArrayAdapter;**  **import android.widget.Spinner;**  **import android.widget.Toast;**  **public class MainActivity extends AppCompatActivity {**  **final Spinner[][] map = new Spinner[4][4];**  **@Override**  **protected void onCreate(Bundle savedInstanceState) {**  **super.onCreate(savedInstanceState);**  **setContentView(R.layout.activity\_main);**  **initElements();**  **setListenersOnElements();**  **findViewById(R.id.buttonReset).setOnClickListener(new View.OnClickListener() {**  **@Override**  **public void onClick(View view) {**  **reset();**  **}**  **});**  **}**  **private void setListenersOnElements(){**  **for (Spinner[] line: map) {**  **for (Spinner el : line) {**  **el.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {**  **@Override**  **public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {**  **checkWinner(adapterView.getSelectedItem().toString().charAt(0));**  **}**  **@Override**  **public void onNothingSelected(AdapterView<?> adapterView) {**  **}**  **});**  **}**  **}**  **}**  **private void checkWinner(char x){**  **if (x!='X' && x!='O') return;**  **boolean win=false;**  **for (Spinner[] line: map)**  **win = win || (countOfSame(line, x)>=3);**  **for (int i = 0;!win && i < 4; i++) {**  **Spinner[] arr = {map[0][i], map[1][i], map[2][i], map[3][i]};**  **win = win || (countOfSame(arr, x)>=3);**  **}**  **win = win || (countOfSame(new Spinner[]{map[1][0], map[2][1], map[3][2]}, x)>=3)**  **|| (countOfSame(new Spinner[]{map[0][0], map[1][1], map[2][2], map[3][3]}, x)>=3)**  **|| (countOfSame(new Spinner[]{map[0][1], map[1][2], map[2][3]}, x)>=3)**  **|| (countOfSame(new Spinner[]{map[0][2], map[1][1], map[2][0]}, x)>=3)**  **|| (countOfSame(new Spinner[]{map[0][3], map[1][2], map[2][1], map[3][0]}, x)>=3)**  **|| (countOfSame(new Spinner[]{map[1][3], map[2][2], map[3][1]}, x)>=3);**  **if (win){**  **makeWinner(x);**  **}**  **}**  **private void makeWinner(char x){**  **Toast toast = Toast.makeText(getApplicationContext(),**  **"Winner is "+x, Toast.LENGTH\_SHORT);**  **toast.show();**  **}**  **private int countOfSame(Spinner[] arr, char x){**  **int r = 0;**  **int max=0;**  **for(Spinner el: arr){**  **char t =el.getSelectedItem().toString().charAt(0);**  **if (t==x) r++;**  **else {**  **max = Math.max(r, max);**  **r=0;**  **}**  **}**  **return max;**  **}**  **private void initElements(){**  **map[0][0] = (Spinner) findViewById(R.id.el1);**  **map[0][1] = (Spinner) findViewById(R.id.el2);**  **map[0][2] = (Spinner) findViewById(R.id.el3);**  **map[0][3] = (Spinner) findViewById(R.id.el4);**  **map[1][0] = (Spinner) findViewById(R.id.el5);**  **map[1][1] = (Spinner) findViewById(R.id.el6);**  **map[1][2] = (Spinner) findViewById(R.id.el7);**  **map[1][3] = (Spinner) findViewById(R.id.el8);**  **map[2][0] = (Spinner) findViewById(R.id.el9);**  **map[2][1] = (Spinner) findViewById(R.id.el10);**  **map[2][2] = (Spinner) findViewById(R.id.el11);**  **map[2][3] = (Spinner) findViewById(R.id.el12);**  **map[3][0] = (Spinner) findViewById(R.id.el13);**  **map[3][1] = (Spinner) findViewById(R.id.el14);**  **map[3][2] = (Spinner) findViewById(R.id.el15);**  **map[3][3] = (Spinner) findViewById(R.id.el16);**  **}**  **private void reset(){**  **for (Spinner[] line: map) {**  **for (Spinner el : line) {**  **el.setSelection(0, true);**  **}**  **}**  **}**  **}** |

|  |
| --- |
| **strings.xml** |
| <resources>  <string name="app\_name">lab24</string>  <string name="el\_width">64dp</string>  <string name="el\_height">64dp</string>  <string-array name="elements">  <item>\_</item>  <item>X</item>  <item>O</item>  </string-array>  </resources> |

|  |
| --- |
| **themes.xml** |
| <resources xmlns:tools="http://schemas.android.com/tools">  <!-- Base application theme. -->  <style name="Theme.Lab24" parent="Theme.MaterialComponents.DayNight.DarkActionBar">  <!-- Primary brand color. -->  <item name="colorPrimary">@color/purple\_500</item>  <item name="colorPrimaryVariant">@color/purple\_700</item>  <item name="colorOnPrimary">@color/white</item>  <!-- Secondary brand color. -->  <item name="colorSecondary">@color/teal\_200</item>  <item name="colorSecondaryVariant">@color/teal\_700</item>  <item name="colorOnSecondary">@color/black</item>  <!-- Status bar color. -->  <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>  <!-- Customize your theme here. -->  </style>  </resources> |

|  |
| --- |
| **activity\_main.xml** |
| **<?xml version="1.0" encoding="utf-8"?>**  **<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:app="http://schemas.android.com/apk/res-auto"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **tools:context=".MainActivity">**  **<TableLayout**  **android:id="@+id/tableLayout"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **app:layout\_constraintBottom\_toBottomOf="parent"**  **app:layout\_constraintEnd\_toEndOf="parent"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toBottomOf="@+id/buttonReset">**  **<TableRow**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent">**  **<Spinner**  **android:id="@+id/el1"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el2"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el3"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el4"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **</TableRow>**  **<TableRow**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent">**  **<Spinner**  **android:id="@+id/el5"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el6"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el7"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el8"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **</TableRow>**  **<TableRow**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent">**  **<Spinner**  **android:id="@+id/el9"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el10"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el11"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el12"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **</TableRow>**  **<TableRow**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent">**  **<Spinner**  **android:id="@+id/el13"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el14"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el15"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **<Spinner**  **android:id="@+id/el16"**  **android:layout\_width="64dp"**  **android:layout\_height="64dp"**  **android:background="@color/element"**  **android:entries="@array/elements" />**  **</TableRow>**  **</TableLayout>**  **<Button**  **android:id="@+id/buttonReset"**  **android:layout\_width="match\_parent"**  **android:layout\_height="wrap\_content"**  **android:text="RESET"**  **app:layout\_constraintEnd\_toEndOf="parent"**  **app:layout\_constraintHorizontal\_bias="0.0"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toTopOf="parent" />**  **</androidx.constraintlayout.widget.ConstraintLayout>** |

## Перший застосунок

Скріншот роботи:

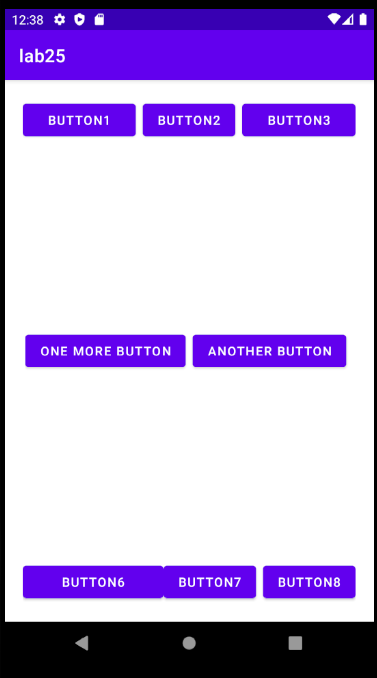


Рис.5 Робота програми на емуляторі.

Приклади коду:

|  |
| --- |
| **activity main.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_margin="0dp"  android:padding="20dp"  tools:context=".MainActivity">  <LinearLayout  android:id="@+id/linearLayout"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="0.25"  android:orientation="horizontal"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.496"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent">  <Button  android:id="@+id/button1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Button1" />  <Button  android:id="@+id/button2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Button2" />  <Button  android:id="@+id/button3"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Button3" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent">  <Button  android:id="@+id/button6"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:text="Button6" />  <Button  android:id="@+id/button7"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Button7" />  <Button  android:id="@+id/button8"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Button8" />  </LinearLayout>  <LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent">  <Button  android:id="@+id/button4"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="One more button" />  <Button  android:id="@+id/button5"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Another button" />  </LinearLayout>  </androidx.constraintlayout.widget.ConstraintLayout> |

## Другий застосунок

Скріншот роботи:

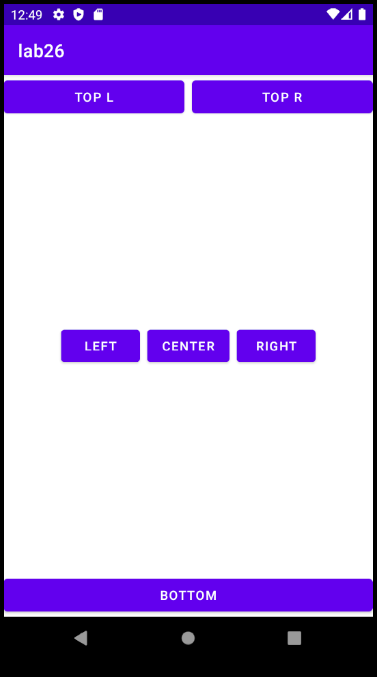


Рис.6 Робота програми на емуляторі.

Приклади коду:

|  |
| --- |
| **activity main.xml** |
| **<?xml version="1.0" encoding="utf-8"?>**  **<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:app="http://schemas.android.com/apk/res-auto"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:id="@+id/parent"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **tools:context=".MainActivity">**  **<Button**  **android:id="@+id/button"**  **android:layout\_width="match\_parent"**  **android:layout\_height="wrap\_content"**  **android:text="Bottom"**  **app:layout\_constraintBottom\_toBottomOf="parent"**  **app:layout\_constraintEnd\_toEndOf="parent"**  **app:layout\_constraintStart\_toStartOf="parent" />**  **<Button**  **android:id="@+id/button2"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Top L"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toTopOf="parent" />**  **<Button**  **android:id="@+id/button3"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Top R"**  **app:layout\_constraintEnd\_toEndOf="parent"**  **app:layout\_constraintTop\_toTopOf="parent" />**  **<Button**  **android:id="@+id/button6"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_marginLeft="10dp"**  **android:text="Right"**  **app:layout\_constraintBottom\_toBottomOf="parent"**  **app:layout\_constraintStart\_toEndOf="@+id/button4"**  **app:layout\_constraintTop\_toTopOf="parent" />**  **<Button**  **android:id="@+id/button5"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_marginRight="10dp"**  **android:text="Left"**  **app:layout\_constraintBottom\_toBottomOf="parent"**  **app:layout\_constraintEnd\_toStartOf="@+id/button4"**  **app:layout\_constraintTop\_toTopOf="parent" />**  **<Button**  **android:id="@+id/button4"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="center"**  **app:layout\_constraintBottom\_toBottomOf="parent"**  **app:layout\_constraintEnd\_toEndOf="parent"**  **app:layout\_constraintStart\_toStartOf="parent"**  **app:layout\_constraintTop\_toTopOf="parent" />**  **</androidx.constraintlayout.widget.ConstraintLayout>** |

## Третій застосунок

Скріншот роботи:

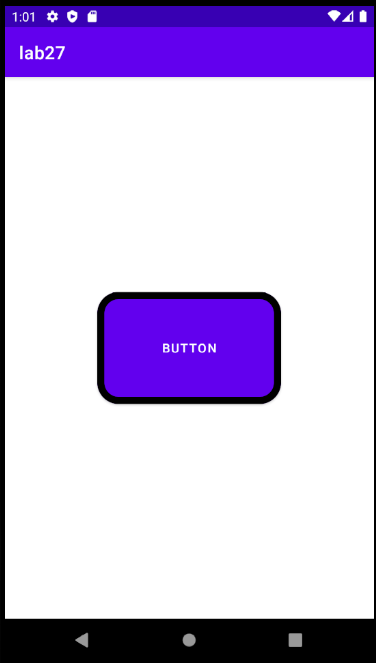


Рис.7 Робота програми на емуляторі.

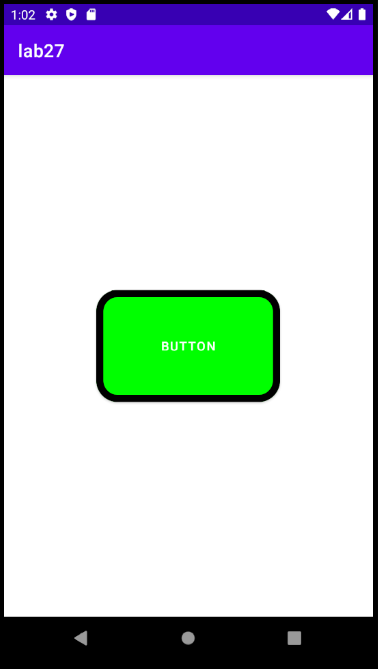


Рис.7 Робота програми на емуляторі (кнопка нажата).

Приклади коду:

|  |
| --- |
| **MainActivity.java** |
| package com.example.lab27;  import androidx.appcompat.app.AppCompatActivity;  import androidx.constraintlayout.widget.ConstraintLayout;  import android.content.res.Resources;  import android.graphics.Point;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  public class MainActivity extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  Point size = new Point();  getWindowManager().getDefaultDisplay().getSize(size);  double perY = size.y / 100.0;  double perX = size.x / 100.0;  Button bt = (Button) findViewById(R.id.button);  ConstraintLayout.LayoutParams params = new ConstraintLayout.LayoutParams((int)(perY\*20\*1.5), (int)(perY\*20));  params.topToTop = ConstraintLayout.LayoutParams.PARENT\_ID;  params.bottomToBottom = ConstraintLayout.LayoutParams.PARENT\_ID;  params.leftToLeft = ConstraintLayout.LayoutParams.PARENT\_ID;  params.rightToRight = ConstraintLayout.LayoutParams.PARENT\_ID;  bt.setLayoutParams(params);  //bt.setBackgroundResource(R.drawable.ugli);  bt.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  //findViewById(R.id.main).setBackgroundColor(getResources().getColor(R.color.light\_green));  findViewById(R.id.button).setBackgroundColor(getResources().getColor(R.color.light\_green));  // bt.setBackgroundResource(R.drawable.ugli);  }  });  }  } |

|  |
| --- |
| **activity main.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:id="@+id/main"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#FFFFFFFF"  tools:context=".MainActivity">  <Button  android:id="@+id/button"  android:layout\_width="216dp"  android:layout\_height="66dp"  android:text="Button"  app:cornerRadius="24dp"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:strokeColor="#000000"  app:strokeWidth="8dp" />  </androidx.constraintlayout.widget.ConstraintLayout> |

# Відповіді на контрольні запитання

## Які елементи управляння / віджети використані у Вашій роботи? Вкажіть переваги використання одного обробника для кількох елементів управління у створеному Android-застосунку.

Основним віджетом, який був використаний майже в усіх застосунках був *Button.* Крім цього в лабораторній роботі були використані такі віджети як *Spinner*, *Toast*, *TextView*, *View* та *EditText*.   
Основною перевагою використання одного обробника для кількох елементів є виконання принципу DRY (don’t repeat yourself). Окрім цього, використання одного обробника замість багатьох призводить до зменшення кількості створених об’єктів, а отже й до економії оперативної пам’яті девайсу.

## 

## Навести приклад обробника для класичного об’єкта Button.

|  |
| --- |
| View.OnClickListener onClickListener= new View.OnClickListener(){  @Override  public void onClick(View view){  Button b = (Button)view;  tview.setText(tview.getText()+ b.getText().toString());  }  };  bt0.setOnClickListener(onClickListener); |

# Висновки

В результаті виконання лабораторної роботи було побудовано 3 інтерактивних додатки: 2 різних калькулятори то гра “Хрестики-Нулики”. Крім цього, на прикладі трьох розробленних інтерфейсів, було опрацьовано можливості, які Android надає для розмітки. Усі програми було протестовані на емуляторі.

# Використані джерела

1. R. Boyer Android 9 development cookbook. 3 edition. Birmingham: Packt Publishing Ltd., 2018.