

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»

Виконав:

студент 3-го курсу, групи КП-83,
спеціальності 121 – Інженерія
програмного забезпечення
Палій Дмитро Володимирович

Перевірив:

к. т. н, старший викладач
Хайдуров Владислав Володимирович

ЗМІСТ

ВСТУП

1. Постановка задачі
2. Короткі теоретичні відомості методу розв'язання задачі
3. Програмна реалізація задачі
4. Результати роботи програми
5. Контрольні питання

ВИСНОВКИ

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

ВСТУП

Android-розробка заключається не лише в створенні дизайну застосунків. Щоб елементи на екрані виконували якісь дії, для них необхідно прописати обробники подій. В даній роботі будуть розглянуті різні способи створення обробників подій.

1. 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 гри хрестики-нулики для поля розміром 4x4. Кожна клітинка повинна мати список Spinner. У звіт додати програмний код та результати роботи Android-програми.

6. Оформити звіт до даної лабораторної роботи.

2. Короткі теоретичні відомості методу розв'язання задачі

Якщо розробник розмістив на екрані кнопку і буде натискати на неї, то нічого не станеться. Необхідно написати код, який буде виконуватися при натисканні. Існує кілька способів обробки натискань на кнопку.

Перший спосіб – атрибут `onClick`. Відносно новий спосіб, спеціально розроблений для Android – використовувати атрибут `onClick` (на панелі властивостей відображається як `OnClick`):
`android:onClick="onMyButtonClick"`

Ім'я для події можна вибрати довільне. Далі потрібно прописати в класі активності придумане вами ім'я методу, який буде обробляти натискання. Метод повинен бути відкритим (`public`) і з одним параметром, що використовують об'єкт `View`. Потрібно вивчити п'ять слів для створення методу, а сам метод помістити в клас:

```
public void onMyButtonClick(View view)
{
    // выводим сообщение
    Toast.makeText(this, "Натиснута кнопка!!!",
        Toast.LENGTH_SHORT).show();
}
```

Коли користувач натискає на кнопку, то викликається метод `onMyButtonClick()`, який в свою чергу генерує спливаюче повідомлення. Слід звернути увагу, що при подібному підході вам не доведеться навіть оголошувати кнопку через конструкцію `(Button) findViewById (R.id.button1)`, так як Android сама зрозуміє, що до чого. Даний метод можна застосовувати не тільки до кнопки, але і до інших елементів і дозволяє скоротити кількість рядків коду.

Даний спосіб не буде працювати в фрагментах. Кнопка повинна бути частиною активності, а не фрагмента.

Другий спосіб – метод `setOnClickListener()`. Більш традиційний спосіб в Java – через метод `setOnClickListener()`, який прослуховує натискання на кнопку. Припустимо, у вас на екрані вже є кнопка `button`. У коді ви оголошуєте її звичайним способом:

```
Button button = (Button) findViewById (R.id.button);
```

Наступний крок – написання методу для натискання. Потрібно надрукувати ім'я елемента і поставите крапку `button`. – середовище розробки покаже список доступних виразів для продовження коду. Розробник може вручну переглянути і вибрати потрібний варіант, а можна продовжувати введення символів запускає прискорити процес. Так як з натисканням кнопок вам часто доведеться працювати, то слід запам'ятати назву його методу (хоча б перші кілька символів) – набравши чотири символи (`seto`), можна побачити один варіант, що залишився, далі можна відразу натиснути клавішу `Enter`, чи не набираючи залишилися символи. У розробника з'явиться рядок такого виду: `button.setOnClickListener ()` Курсор буде знаходитися всередині дужок і з'явиться підказка `OnClickListener`. Починайте набирати `new OnClickListener`. Тут також не обов'язково набирати ім'я повністю. Набравши слово `Oncl`, ви побачите потрібний варіант і знову натискайте `Enter`. В результаті ви отримаєте готову заготовку для обробки натискання кнопки:

```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
    }  
});
```

Тепер у розробника є робоча заготовка і відразу всередині фігурних дужок методу `onClick()` можна писати власний програмний код. Як варіант, можна винести код для `OnClickListener` в окреме місце, це зручно, коли кнопок на екрані кілька і такий підхід дозволить упорядкувати код. Видаліть попередній приклад і треба почати писати код заново. Принцип такий же, трохи змінюється порядок.

У попередньому прикладі відразу прописувалось в методі `setOnClickListener` слухача `new OnClickListener ...` з методом `onClick ()`. Можна спочатку окремо оголосити окрему змінну

```
myButtonClickListener:
OnClickListener myButtonClickListener = new
OnClickListener() {
    @Override
    public void onClick(View v) {
    }
};
```

Під час набору активно використовуйте підказки через `Ctrl + Space`. Набрали кілька символів у першого слова і натискайте цю комбінацію, набрали після слова `new` кілька символів і знову натискайте зазначену комбінацію – заготовка буде створена за декілька секунд, а ви уникнете можливих помилок.

Для новачка опис може здатися сумбурним і незрозумілим, але краще самостійно виконати ці операції і зрозуміти механізм.

Третій спосіб є родинним другим способом і також є традиційним для Java. Кнопка привласнює собі обробника за допомогою методу `setOnClickListener (View.OnClickListener l)`, тобто підійде будь-який об'єкт з інтерфейсом `View.OnClickListener`. Ми можемо вказати, що наш клас `Activity` буде використовувати інтерфейс `View.OnClickListener`.

Знову стираємо код від попереднього прикладу. Далі після слів `extends Activity` дописуємо слова `implements OnClickListener`. При появі підказки не помиліться. Зазвичай першим йде інтерфейс для діалогів, а другим потрібний нам `View.OnClickListener`. Назва вашого класу буде підкреслено хвилястою червоною рисою, клацніть слово `public` і дочекайтеся появи червоної лампочки, виберіть варіант `Implement methods`. З'явиться діалогове вікно з виділеним методом `onClick`. Вибираємо його і в коді з'явиться заготовка для натискання кнопки.

```
@Override
public void onClick(View v) {
}
```

Метод буде реалізований не в окремому об'єкті-обробнику, а в Activity, який і буде виступати оброблювачем. У методі onCreate() призначимо обробник кнопки. Це буде об'єкт this, тобто поточний об'єкт нашої активності.

```
button.setOnClickListener(this);
```


2. Програмна реалізація задачі

XML-код простого калькулятора для другого та третього пунктів завдання такий самий, як і для попередньої лабораторної.

JAVA-код для 2-го пункту завдання:

```
public class MainActivity extends AppCompatActivity {

    enum Operation {
        DIV,
        MUL,
        ADD,
        SUB,
        NONE
    }

    Button Bt0, Bt1, Bt2, Bt3, Bt4, Bt5, Bt6, Bt7, Bt8, Bt9, BtAdd,
    BtSub, BtMul, BtDiv, BtEq, BtC, BtDot;
    TextView tview;
    String input = "0";
    Operation op = Operation.NONE;
    Boolean opChosen = false;
    Boolean hasDot = false;
    Double result = 0.0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Bt0 = (Button) findViewById(R.id.btn0);
        Bt1 = (Button) findViewById(R.id.btn1);
        Bt2 = (Button) findViewById(R.id.btn2);
        Bt3 = (Button) findViewById(R.id.btn3);
        Bt4 = (Button) findViewById(R.id.btn4);
        Bt5 = (Button) findViewById(R.id.btn5);
        Bt6 = (Button) findViewById(R.id.btn6);
        Bt7 = (Button) findViewById(R.id.btn7);
        Bt8 = (Button) findViewById(R.id.btn8);
        Bt9 = (Button) findViewById(R.id.btn9);
        BtAdd = (Button) findViewById(R.id.btnAdd);
        BtSub = (Button) findViewById(R.id.btnSub);
        BtMul = (Button) findViewById(R.id.btnMul);
        BtDiv = (Button) findViewById(R.id.btnDiv);
        BtC = (Button) findViewById(R.id.btnC);
        BtEq = (Button) findViewById(R.id.btnEq);
        BtDot = (Button) findViewById(R.id.btnDot);
        tview = (TextView) findViewById(R.id.textView);

        tview.setText(input);
        Bt0.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                inputNumber("0");
            }
        });
        // код інших кнопок для введення цифр аналогічний
    }
}
```

Звіт до лабораторної роботи No 2. © Палій Д. В.

```
BtDot.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!hasDot) {
            if (input.equals("0")) inputNumber("0.");
            else inputNumber(".");
            hasDot = true;
        }
    }
});

BtC.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        resetInput();
        result = 0.0;
        tview.setText(input);
        op = Operation.NONE;
    }
});

BtAdd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        executeOperation();
        op = Operation.ADD;
        opChosen = true;
    }
});

BtSub.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        executeOperation();
        op = Operation.SUB;
        opChosen = true;
    }
});

BtDiv.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        executeOperation();
        op = Operation.DIV;
        opChosen = true;
    }
});

BtMul.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        executeOperation();
        op = Operation.MUL;
        opChosen = true;
    }
});
```

Звіт до лабораторної роботи No 2. © Палій Д. В.

```
BtEq.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        executeOperation();
    }
});

private void inputNumber(String num) {
    if (opChosen) {
        resetInput();
    }
    if (input.equals("0")) input = "";
    input += num;
    tview.setText(input);
}

private void resetInput() {
    opChosen = false;
    hasDot = false;
    input = "0";
}

private void executeOperation() {
    Double num = Double.parseDouble(this.input);
    switch (op) {
        case ADD: {
            result += num;
            break;
        }
        case DIV: {
            if (num == 0) {
                Toast toast = Toast.makeText(getApplicationContext(),
                    "Can't divide by 0", Toast.LENGTH_SHORT);
                toast.show();
                resetInput();
                input = result.toString();
                tview.setText(input);
                break;
            }
            result /= num;
            break;
        }
        case MUL: {
            result *= num;
            break;
        }
        case SUB: {
            result -= num;
            break;
        }
        case NONE: {
            result = num;
            break;
        }
    }
    op = Operation.NONE;
    opChosen = false;
}
```

Звіт до лабораторної роботи No 2. © Палій Д. В.

```
        input = result.toString();
        tview.setText(input);
    }
}
```

JAVA-код для 3-го пункту завдання:

```
Button NumButtons[], BtAdd, BtSub, BtMul, BtDiv, BtEq, BtC, BtDot;
TextView tview;
String input = "0";
Operation op = Operation.NONE;
Boolean opChosen = false;
Boolean hasDot = false;
Double result = 0.0;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    NumButtons = new Button[10];

    NumButtons[0] = (Button) findViewById(R.id.btn0);
    NumButtons[1] = (Button) findViewById(R.id.btn1);
    NumButtons[2] = (Button) findViewById(R.id.btn2);
    NumButtons[3] = (Button) findViewById(R.id.btn3);
    NumButtons[4] = (Button) findViewById(R.id.btn4);
    NumButtons[5] = (Button) findViewById(R.id.btn5);
    NumButtons[6] = (Button) findViewById(R.id.btn6);
    NumButtons[7] = (Button) findViewById(R.id.btn7);
    NumButtons[8] = (Button) findViewById(R.id.btn8);
    NumButtons[9] = (Button) findViewById(R.id.btn9);

    View.OnClickListener numOnClickListener = new View.OnClickListener()
    {
        @Override
        public void onClick(View view) {
            Button currButton = (Button) view;
            inputNumber((String) currButton.getText());
        }
    };

    for (Button btn: NumButtons) {
        btn.setOnClickListener(numOnClickListener);
    }
    // Решта коду така сама, як і в попередньому пункті
}
```

JAVA-код для 4-го пункту завдання:

```
package com.example.lab1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
```

Звіт до лабораторної роботи No 2. © Палій Д. В.

```
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    Button EvalButton;
    EditText FirstNumberInput, SecondNumberInput;
    TextView ResultView;
    Spinner OpSpinner;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        EvalButton = findViewById(R.id.evalButton);
        FirstNumberInput = findViewById(R.id.firstNumInput);
        SecondNumberInput = findViewById(R.id.secondNumInput);
        ResultView = findViewById(R.id.result);
        OpSpinner = findViewById(R.id.opSpinner);

        EvalButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Double firstNum =
Double.parseDouble(FirstNumberInput.getText().toString());
                Double secondNum =
Double.parseDouble(SecondNumberInput.getText().toString());

                String op = OpSpinner.getSelectedItem().toString();

                Double result = 0.0;
                switch (op) {
                    case "+": {
                        result = firstNum + secondNum;
                        break;
                    }
                    case "/": {
                        if (secondNum == 0) {
                            Toast toast = Toast.makeText(getApplicationContext(),
                                "Can't divide by 0", Toast.LENGTH_SHORT);
                            toast.show();
                            result = Double.POSITIVE_INFINITY;
                            break;
                        }
                        result = firstNum / secondNum;
                        break;
                    }
                    case "*": {
                        result = firstNum * secondNum;
                        break;
                    }
                    case "-": {
                        result = firstNum - secondNum;
                        break;
                    }
                }
            }
        })
    }
}
```

```
        ResultView.setText(result.toString());
    }
});
}
```

XML-код калькулятора для п.4 завдання

```
<?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:background="@color/colorBg"
    android:padding="0dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginBottom="8dp"
        android:text="@string/label_second"
        android:textSize="24sp"
        app:layout_constraintBottom_toTopOf="@+id/secondNumInput"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/firstNumInput" />

    <EditText
        android:id="@+id/secondNumInput"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginBottom="8dp"
        android:autoText="true"
        android:ems="10"
        android:importantForAutofill="no"
        android:inputType="numberSigned"
        android:textSize="24sp"
        app:layout_constraintBottom_toTopOf="@+id/opSpinner"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
```

Звіт до лабораторної роботи No 2. © Палій Д. В.

```
app:layout_constraintTop_toBottomOf="@+id/textView2" />

<EditText
    android:id="@+id/firstNumInput"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    android:ems="10"
    android:importantForAutofill="no"
    android:inputType="numberSigned"
    android:textSize="24sp"
    app:layout_constraintBottom_toTopOf="@+id/textView2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />

<TextView
    android:id="@+id/textView"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    android:text="@string/label_first"
    android:textSize="24sp"
    app:layout_constraintBottom_toTopOf="@+id/firstNumInput"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Spinner
    android:id="@+id/opSpinner"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    android:entries="@array/operations"
    app:layout_constraintBottom_toTopOf="@+id/result"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/textView3"
    app:layout_constraintTop_toBottomOf="@+id/secondNumInput" />

<Button
    android:id="@+id/evalButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
```

Звіт до лабораторної роботи No 2. © Палій Д. В.

```
android:layout_marginStart="8dp"
android:layout_marginLeft="8dp"
android:layout_marginEnd="8dp"
android:layout_marginRight="8dp"
android:padding="0sp"
android:text="@string/eval_btn"
android:textSize="24sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="64dp"
    android:text="@string/label_ops"
    android:textSize="24sp"
    app:layout_constraintBottom_toTopOf="@+id/textView4"
    app:layout_constraintEnd_toStartOf="@+id/opSpinner"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/secondNumInput" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:text="@string/result_label"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id/result"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />

<TextView
    android:id="@+id/result"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/textView4"
    app:layout_constraintTop_toBottomOf="@+id/opSpinner" />
```



```
</androidx.constraintlayout.widget.ConstraintLayout>
```

JAVA-код гри хрестики-нулики:

```
package com.example.lab1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    Button[][] Buttons;
    String currPlayer = "X";
    int tilesActivated = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Buttons = new Button[4][4];

        Buttons[0][0] = findViewById(R.id.btn00);
        Buttons[0][1] = findViewById(R.id.btn01);
        Buttons[0][2] = findViewById(R.id.btn02);
        Buttons[0][3] = findViewById(R.id.btn03);
        Buttons[1][0] = findViewById(R.id.btn10);
        Buttons[1][1] = findViewById(R.id.btn11);
        Buttons[1][2] = findViewById(R.id.btn12);
        Buttons[1][3] = findViewById(R.id.btn13);
        Buttons[2][0] = findViewById(R.id.btn20);
        Buttons[2][1] = findViewById(R.id.btn21);
        Buttons[2][2] = findViewById(R.id.btn22);
        Buttons[2][3] = findViewById(R.id.btn23);
        Buttons[3][0] = findViewById(R.id.btn30);
        Buttons[3][1] = findViewById(R.id.btn31);
        Buttons[3][2] = findViewById(R.id.btn32);
        Buttons[3][3] = findViewById(R.id.btn33);

        View.OnClickListener buttonClickListener = new
        View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Button currButton = (Button) view;
                int x = -1, y = -1;
                for (int i = 0; i < 4; i++)
                    for (int j = 0; j < 4; j++) {
                        if (Buttons[i][j] == currButton) {
                            x = i;
                            y = j;
                        }
                    }
            }
        }
    }
}
```

Звіт до лабораторної роботи No 2. © Палій Д. В.

```
    }
    currButton.setText(currPlayer);
    currButton.setEnabled(false);
    tilesActivated += 1;

    if ((Buttons[x][0].getText() == Buttons[x][1].getText()
    && Buttons[x][1].getText() == Buttons[x][2].getText() &&
    Buttons[x][2].getText() == Buttons[x][3].getText()) ||
        (Buttons[0][y].getText() == Buttons[1][y].getText()
    && Buttons[1][y].getText() == Buttons[2][y].getText() &&
    Buttons[2][y].getText() == Buttons[3][y].getText()) ||
        (x == 3 - y && Buttons[0][3].getText() ==
    Buttons[1][2].getText() && Buttons[1][2].getText() ==
    Buttons[2][1].getText() && Buttons[2][1].getText() ==
    Buttons[3][0].getText()) ||
        (x == y && Buttons[0][0].getText() ==
    Buttons[1][1].getText() && Buttons[1][1].getText() ==
    Buttons[2][2].getText() && Buttons[2][2].getText() ==
    Buttons[3][3].getText())) {
        Toast.makeText(getApplicationContext(), "Player "
+ currPlayer + " won!", Toast.LENGTH_SHORT).show();
        resetField();
    }
    else
    if (tilesActivated == 16) {
        Toast.makeText(getApplicationContext(), "Draw!",
Toast.LENGTH_SHORT).show();
        resetField();
    }

    currPlayer = currPlayer.equals("X") ? "O" : "X";
}
};

for (int i = 0; i < 4; i++)
    for (int j = 0; j < 4; j++)
        Buttons[i][j].setOnClickListener(buttonClickListener);
}

private void resetField() {
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++) {
            Buttons[i][j].setText("");
            Buttons[i][j].setEnabled(true);
        }

    tilesActivated = 0;
}
}
```

XML-код гри хрестики-нулики:

```
<?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"
```

Звіт до лабораторної роботи No 2. © Палій Д. В.

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/colorBg"
tools:context=".MainActivity">

<Button
    android:id="@+id/btn30"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_margin="1dp"
    android:background="@color/colorTextBg"
    android:textColor="@color/colorText"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline3"
    app:layout_constraintHorizontal_bias="0.507"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btn20" />
    ...Решта кнопок такі самі
</androidx.constraintlayout.widget.ConstraintLayout>
```

3. Результати роботи програми

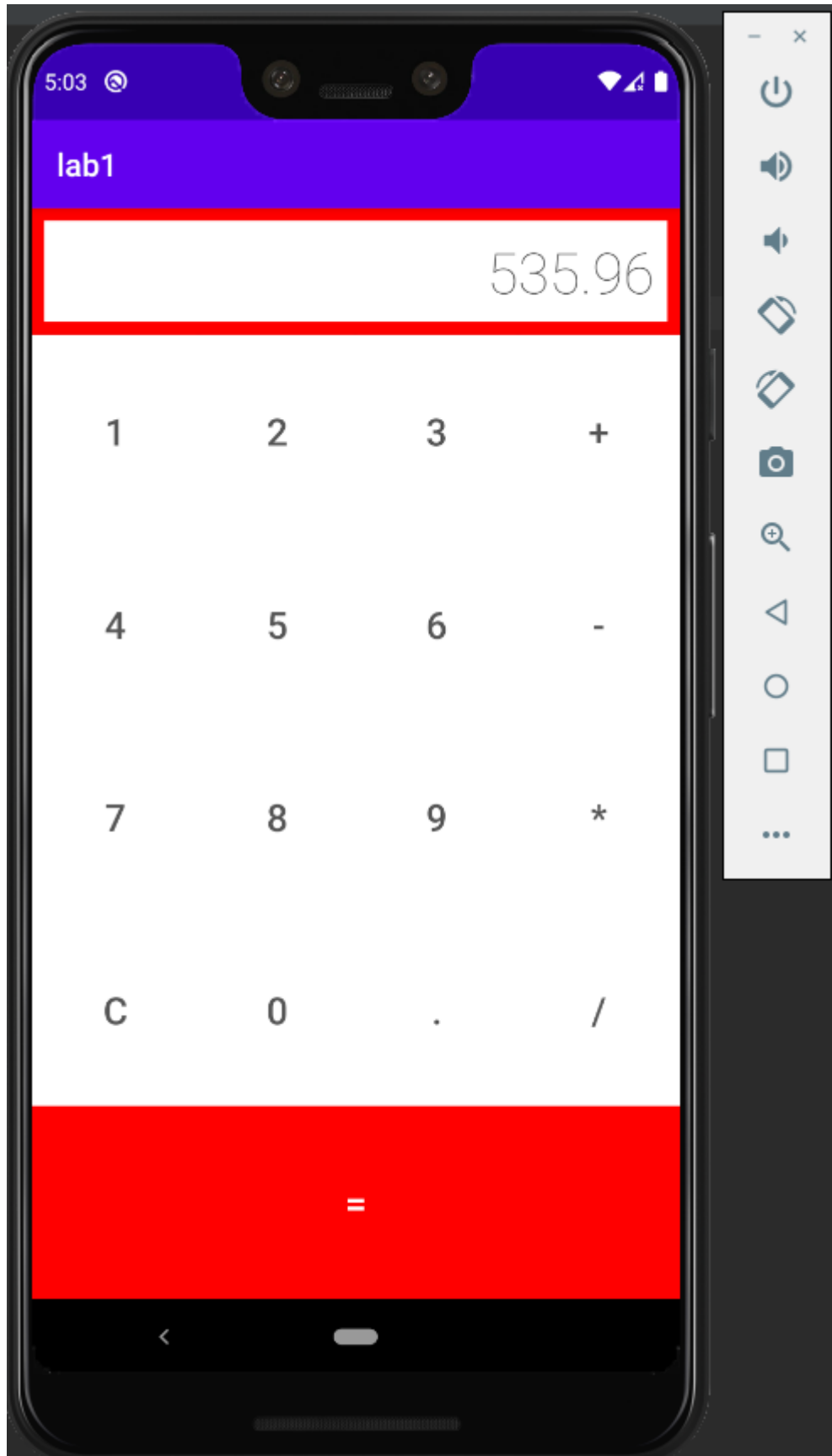


Рис. 1. Інтерфейс калькулятора, п.2-3 завдання

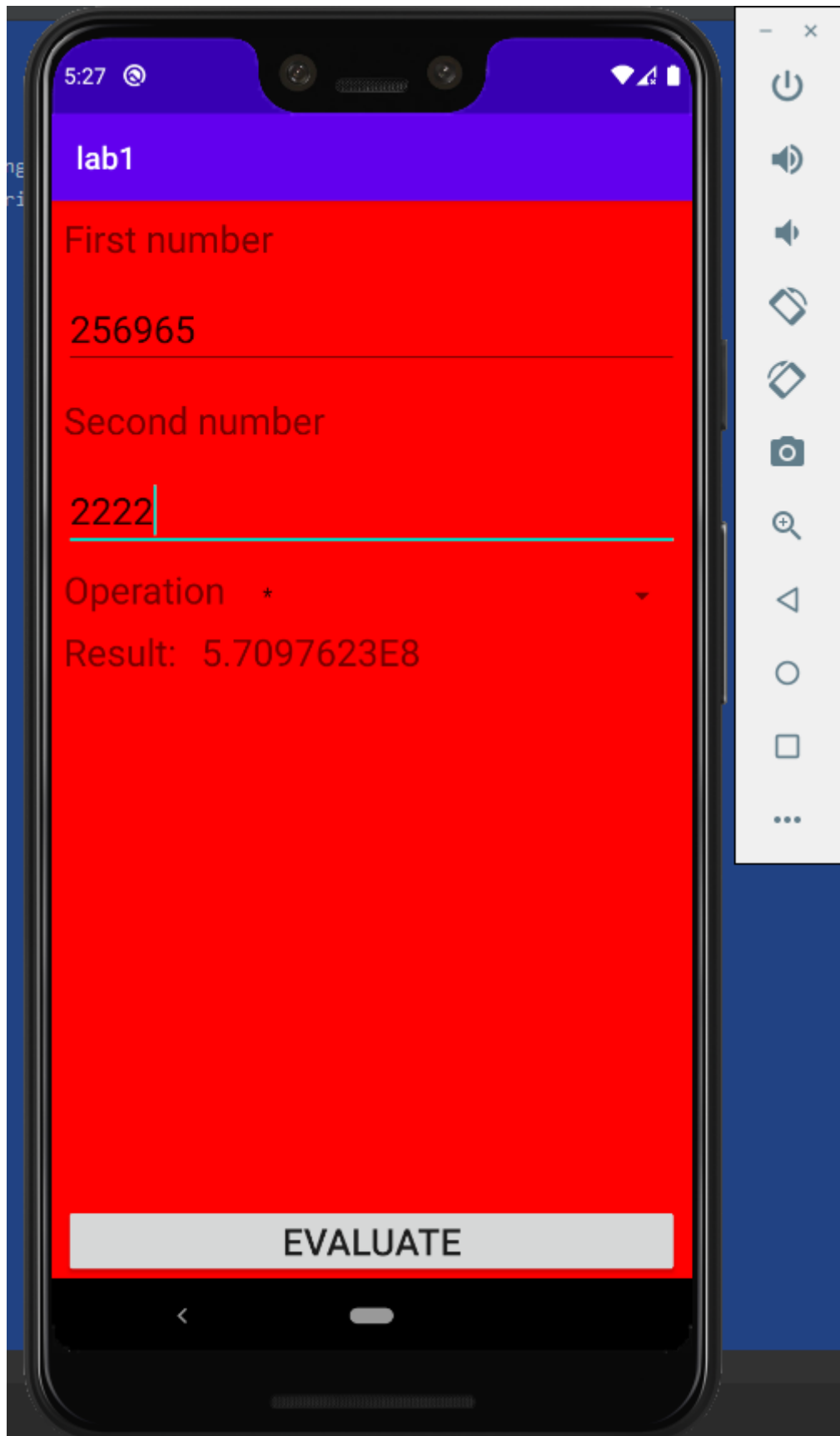


Рис. 2. Інтерфейс калькулятора, п.4 завдання

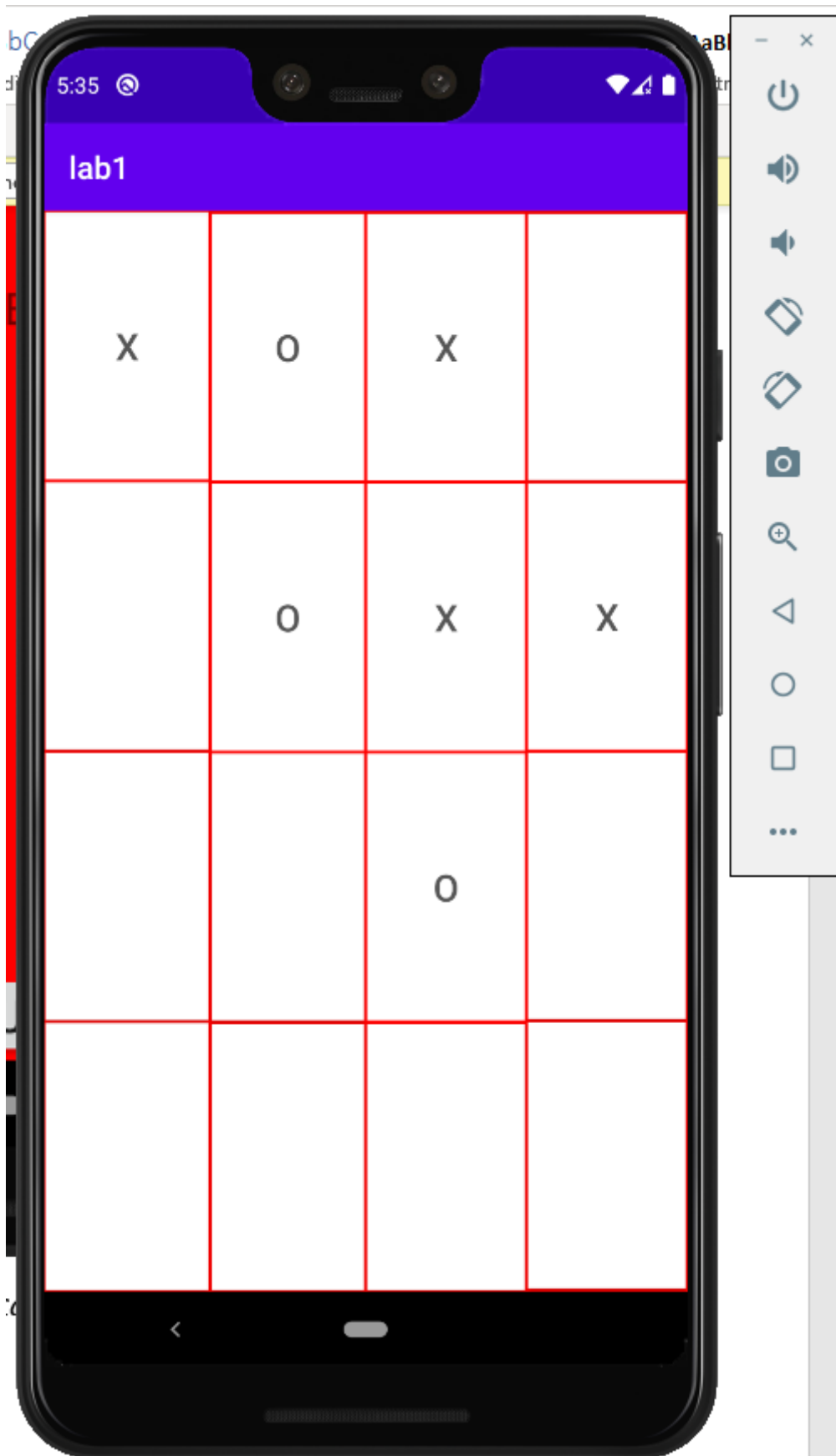


Рис. 3. Гра хрестики-нулики

5. Контрольні питання

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

У моїй роботі були використані такі елементи управління / віджети:

- Button
- TextView
- EditText
- Spinner

При використанні одного обробника для кількох елементів управління виділяється менше пам'яті, а також код стає коротшим та легшим для сприйняття.

3. Навести приклад обробника для класичного об'єкта Button.
onClick()

ВИСНОВОК

В ході розв'язання поставленої задачі були розглянуті різні способи створення обробників подій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. <https://metanit.com/java/android/4.8.php>
2. <https://metanit.com/java/android/4.1.php>
3. <https://metanit.com/java/android/4.9.php>