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

*Мар’яненко Роман Геннадійович*

**Перевірив:**

к. т. н, старший викладач

*Хайдуров Владислав Володимирович*

Київ – 2020

**ЗМІСТ**

**ВСТУП**……………………………………………………………………………….**3**

1. Завдання до лабораторної роботи..………………………………………………4

2. Короткі теоретичні відомості…………………………………………………….5

3. Програмна реалізація задачі……………………………………………………...6

4. Результати роботи програми…………………………………………………….22

**ВИСНОВКИ**………………………………………………………………………..25

**ВСТУП**

Під час виконання лабораторної роботи ми дізнаємося що таке подія та її обробники в Android Studio, навчимося обробляти класичні події які дозволяє створювати середовище розробки для мобільних застосунків, напишемо два варіанти працюючого коду для інтерфейсу калькулятора, створеного у минулій лабораторній роботі та створимо мобільний варіант відомої гри з затвердженими правилами.

**Завдання до лабораторної роботи**

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 (+, -, \*, /) в головному вікні (Main\_Activity) створити Spinner. Для виконання обрахунків додати до інтерфейсу кнопку (Button). Інтерфейс має бути подібний до такого:

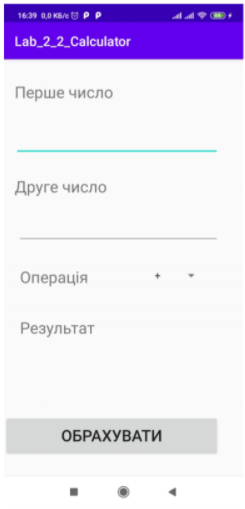


Рис. 1, Приклад інтерфейсу калькулятора

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

1. 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.
2. Оформити звіт до даної лабораторної роботи.

**Короткі теоретичні відомості**

Подія — дія яка розпізнається програмним забезпеченням та обробляється за допомогою певних інструкцій. Комп'ютерні події можуть бути згенеровані або ініціалізовані системно, користувачем або іншими способами.

Обробник подій — це підпрограма, яка опрацьовує матеріали, отримані з програми. Обробник подій певним чином реагує на події, та починає виконувати дії які згенерувала та чи інша подія.

Spinner — віджет з вкладки Containers що нагадує випадаючий список що використовується у ОС Windows.

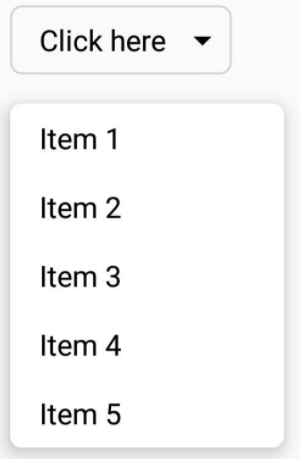


Рис. 2, Зовнішній вигляд віджету Spinner в Android Studio

**Програмна реалізація задачі**

***Пункт 2:***

Файли res/values/colors.xml, layout/activity\_main.xml та res/values/styles.xml не змінилися з минулої ЛР.

|  |
| --- |
| res/values/strings.xml |
| <resources>  <string name="app\_name">task1</string>  <string name="screen\_placeholder">0</string>  <string name="button\_1">1</string>  <string name="button\_2">2</string>  <string name="button\_3">3</string>  <string name="button\_4">4</string>  <string name="button\_5">5</string>  <string name="button\_6">6</string>  <string name="button\_7">7</string>  <string name="button\_8">8</string>  <string name="button\_9">9</string>  <string name="button\_0">0</string>  <string name="button\_plus">+</string>  <string name="button\_minus">-</string>  <string name="button\_divide">/</string>  <string name="button\_multiply">\*</string>  <string name="button\_equal">=</string>  <string name="button\_erase">C</string>  </resources> |

|  |
| --- |
| java/com.example.lab2/MainActivity.java |
| package com.example.lab2;  import androidx.appcompat.app.AppCompatActivity;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.TextView;  public class MainActivity extends AppCompatActivity {  Button button\_1;  Button button\_2;  Button button\_3;  Button button\_4;  Button button\_5;  Button button\_6;  Button button\_7;  Button button\_8;  Button button\_9;  Button button\_0;  Button button\_plus;  Button button\_minus;  Button button\_multiply;  Button button\_divide;  Button button\_erase;  Button button\_equal;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  button\_1 = findViewById(R.id.button\_1);  button\_2 = findViewById(R.id.button\_2);  button\_3 = findViewById(R.id.button\_3);  button\_4 = findViewById(R.id.button\_4);  button\_5 = findViewById(R.id.button\_5);  button\_6 = findViewById(R.id.button\_6);  button\_7 = findViewById(R.id.button\_7);  button\_0 = findViewById(R.id.button\_0);  button\_8 = findViewById(R.id.button\_8);  button\_9 = findViewById(R.id.button\_9);  button\_plus = findViewById(R.id.button\_plus);  button\_minus = findViewById(R.id.button\_minus);  button\_multiply = findViewById(R.id.button\_multiply);  button\_divide = findViewById(R.id.button\_divide);  button\_erase = findViewById(R.id.button\_erase);  button\_equal = findViewById(R.id.button\_equal);  button\_1.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('1');  }  });  button\_2.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('2');  }  });  button\_3.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('3');  }  });  button\_4.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('4');  }  });  button\_5.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('5');  }  });  button\_6.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('6');  }  });  button\_7.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('7');  }  });  button\_8.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('8');  }  });  button\_9.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('9');  }  });  button\_0.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('0');  }  });  button\_plus.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('+');  }  });  button\_minus.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('-');  }  });  button\_multiply.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('\*');  }  });  button\_divide.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('/');  }  });  button\_erase.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('C');  }  });  button\_equal.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  updateScreenStringWithChar('=');  }  });  }  private void updateScreenStringWithChar(char ch) {  TextView tv = findViewById(R.id.screen);  String screenStr = tv.getText().toString();  if (screenStr.contains("=")) screenStr = "0";  screenStr = updateStringWithChar(screenStr, ch);  tv.setText(screenStr);  }  private String updateStringWithChar(String screenStr, char ch) {  if (Character.isDigit(ch)) {  if ((screenStr.length() > 0 && screenStr.charAt(screenStr.length() - 1) == '0' && screenStr.length() == 1) || (screenStr.length() >= 2 && screenStr.charAt(screenStr.length() - 1) == '0' && !Character.isDigit(screenStr.charAt(screenStr.length() - 2))))  screenStr = replaceLastCharInStr(screenStr, ch);  else screenStr += ch;  }  else if (ch == 'C') {  screenStr = screenStr.substring(0, screenStr.length() - 1);  if (screenStr.length() == 0) screenStr += '0';  }  else if (ch == '+' || ch == '-' || ch == '\*' || ch == '/') {  if (screenStr.contains("+") || screenStr.contains("-") || screenStr.contains("\*") || screenStr.contains("/")) {  char lastCh = screenStr.charAt(screenStr.length() - 1);  if (lastCh == '+' || lastCh == '-' || lastCh == '\*' || lastCh == '/')  screenStr = replaceLastCharInStr(screenStr, ch);  }  else screenStr += ch;  }  else if (ch == '=') screenStr = parseAndCalculate(screenStr);  return screenStr;  }  private String replaceLastCharInStr(String str, char ch) {  return str.substring(0, str.length() - 1) + ch;  }  private String parseAndCalculate(String screenStr) {  ExampleParts parts = new ExampleParts(screenStr);  if (parts.getLeftPart() < 0 || parts.getRightPart() < 0 || parts.getOperator() == '\0')  return screenStr;  return screenStr + '=' + parts.calculate();  }  }  class ExampleParts {  private int leftPart;  private int rightPart;  private char operator;  ExampleParts(String str) {  StringBuilder leftPart\_ = new StringBuilder();  StringBuilder rightPart\_ = new StringBuilder();  StringBuilder operator\_ = new StringBuilder();  boolean isOperatorWasHere = false;  for (int i = 0; i < str.length(); i++) {  char ch = str.charAt(i);  if (Character.isDigit(ch)) {  if (!isOperatorWasHere) {  leftPart\_.append(ch);  } else {  rightPart\_.append(ch);  }  }  else if ((ch == '+' || ch == '-' || ch == '\*' || ch == '/') && !isOperatorWasHere) {  operator\_.append(ch);  isOperatorWasHere = true;  }  }  leftPart = (leftPart\_.length() > 0) ? Integer.parseInt(leftPart\_.toString()) : -1;  rightPart = (rightPart\_.length() > 0) ? Integer.parseInt(rightPart\_.toString()) : -1;  operator = (operator\_.length() > 0) ? operator\_.charAt(0) : '\0';  }  public int getLeftPart() {  return leftPart;  }  public int getRightPart() {  return rightPart;  }  public int getOperator() {  return operator;  }  public double calculate() {  if (leftPart < 0 || rightPart < 0 || operator == '\0')  return -1;  switch (operator) {  case '+':  return leftPart + rightPart;  case '-':  return leftPart - rightPart;  case '\*':  return leftPart \* rightPart;  case '/':  return (double) leftPart / (double) rightPart;  default:  return -1;  }  }  } |

***Пункт 3:***

Файли, що не перечислені нище, не було змінено з попереднього пункту

|  |
| --- |
| res/values/styles.xml |
| <resources>  <!-- Base application theme. -->  <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  <!-- Customize your theme here. -->  <item name="colorPrimary">@color/colorPrimary</item>  <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  <item name="colorAccent">@color/colorAccent</item>  </style>  <style name="button">  <item name="android:layout\_height">fill\_parent</item>  <item name="android:layout\_weight">1</item>  <item name="android:textColor">@color/textColor</item>  <item name="android:textSize">30sp</item>  <item name="android:onClick">updateScreenStringWithChar</item>  </style>  </resources> |

|  |
| --- |
| java/com.example.lab2/MainActivity.java |
| package com.example.lab2;  import androidx.appcompat.app.AppCompatActivity;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.TextView;  public class MainActivity extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  }  public void updateScreenStringWithChar(View view) {  Button btn = (Button) view;  char ch = btn.getText().charAt(0);  TextView tv = findViewById(R.id.screen);  String screenStr = tv.getText().toString();  if (screenStr.contains("=")) screenStr = "0";  screenStr = updateStringWithChar(screenStr, ch);  tv.setText(screenStr);  }  private String updateStringWithChar(String screenStr, char ch) {  if (Character.isDigit(ch)) {  if ((screenStr.length() > 0 && screenStr.charAt(screenStr.length() - 1) == '0' && screenStr.length() == 1) || (screenStr.length() >= 2 && screenStr.charAt(screenStr.length() - 1) == '0' && !Character.isDigit(screenStr.charAt(screenStr.length() - 2))))  screenStr = replaceLastCharInStr(screenStr, ch);  else screenStr += ch;  }  else if (ch == 'C') {  screenStr = screenStr.substring(0, screenStr.length() - 1);  if (screenStr.length() == 0) screenStr += '0';  }  else if (ch == '+' || ch == '-' || ch == '\*' || ch == '/') {  if (screenStr.contains("+") || screenStr.contains("-") || screenStr.contains("\*") || screenStr.contains("/")) {  char lastCh = screenStr.charAt(screenStr.length() - 1);  if (lastCh == '+' || lastCh == '-' || lastCh == '\*' || lastCh == '/')  screenStr = replaceLastCharInStr(screenStr, ch);  }  else screenStr += ch;  }  else if (ch == '=') screenStr = parseAndCalculate(screenStr);  return screenStr;  }  private String replaceLastCharInStr(String str, char ch) {  return str.substring(0, str.length() - 1) + ch;  }  private String parseAndCalculate(String screenStr) {  ExampleParts parts = new ExampleParts(screenStr);  if (parts.getLeftPart() < 0 || parts.getRightPart() < 0 || parts.getOperator() == '\0')  return screenStr;  return screenStr + '=' + parts.calculate();  }  }  class ExampleParts {  private int leftPart;  private int rightPart;  private char operator;  ExampleParts(String str) {  StringBuilder leftPart\_ = new StringBuilder();  StringBuilder rightPart\_ = new StringBuilder();  StringBuilder operator\_ = new StringBuilder();  boolean isOperatorWasHere = false;  for (int i = 0; i < str.length(); i++) {  char ch = str.charAt(i);  if (Character.isDigit(ch)) {  if (!isOperatorWasHere) {  leftPart\_.append(ch);  } else {  rightPart\_.append(ch);  }  }  else if ((ch == '+' || ch == '-' || ch == '\*' || ch == '/') && !isOperatorWasHere) {  operator\_.append(ch);  isOperatorWasHere = true;  }  }  leftPart = (leftPart\_.length() > 0) ? Integer.parseInt(leftPart\_.toString()) : -1;  rightPart = (rightPart\_.length() > 0) ? Integer.parseInt(rightPart\_.toString()) : -1;  operator = (operator\_.length() > 0) ? operator\_.charAt(0) : '\0';  }  public int getLeftPart() {  return leftPart;  }  public int getRightPart() {  return rightPart;  }  public int getOperator() {  return operator;  }  public double calculate() {  if (leftPart < 0 || rightPart < 0 || operator == '\0')  return -1;  switch (operator) {  case '+':  return leftPart + rightPart;  case '-':  return leftPart - rightPart;  case '\*':  return leftPart \* rightPart;  case '/':  return (double) leftPart / (double) rightPart;  default:  return -1;  }  }  } |

***Пункт 4:***

|  |
| --- |
| layout/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">  <com.google.android.material.textfield.TextInputLayout  android:id="@+id/first\_num"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView">  <com.google.android.material.textfield.TextInputEditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="@string/input\_field\_placeholder"  android:inputType="number|numberDecimal" />  </com.google.android.material.textfield.TextInputLayout>  <com.google.android.material.textfield.TextInputLayout  android:id="@+id/second\_num"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView4">  <com.google.android.material.textfield.TextInputEditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="@string/input\_field\_placeholder"  android:inputType="numberDecimal" />  </com.google.android.material.textfield.TextInputLayout>  <TextView  android:id="@+id/textView"  style="@style/caption\_style"  android:text="@string/first\_field\_caption"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />  <TextView  android:id="@+id/textView4"  style="@style/caption\_style"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:text="@string/second\_field\_caption"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/first\_num" />  <TextView  android:id="@+id/textView5"  style="@style/caption\_style"  android:layout\_marginTop="30dp"  android:text="@string/result\_caption"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/operators" />  <Spinner  android:id="@+id/operators"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_margin="30sp"  android:entries="@array/operators"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/second\_num" />  <TextView  android:id="@+id/result"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="7sp"  android:layout\_marginTop="10dp"  android:layout\_marginRight="7sp"  android:textSize="20sp"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView5" />  <Button  android:id="@+id/calculate"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="5dp"  android:paddingLeft="40sp"  android:paddingRight="40sp"  android:text="@string/calculate\_button\_caption"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout> |

|  |
| --- |
| java/com.example.lab2/MainActivity.java |
| package com.example.task3;  import androidx.appcompat.app.AppCompatActivity;  import android.annotation.SuppressLint;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.Spinner;  import android.widget.TextView;  import com.google.android.material.textfield.TextInputLayout;  public class MainActivity extends AppCompatActivity {  Button calculate;  TextView result;  Spinner operators;  TextInputLayout firstNum;  TextInputLayout secondNum;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  calculate = findViewById(R.id.calculate);  result = findViewById(R.id.result);  operators = findViewById(R.id.operators);  firstNum = findViewById(R.id.first\_num);  secondNum = findViewById(R.id.second\_num);  calculate.setOnClickListener(new View.OnClickListener() {  @SuppressLint("SetTextI18n")  @Override  public void onClick(View v) {  String firstValueStr = firstNum.getEditText().getText().toString();  String secondValueStr = secondNum.getEditText().getText().toString();  String operator = operators.getSelectedItem().toString();  if (firstValueStr.length() <= 0 || secondValueStr.length() <= 0) {  result.setText(R.string.input\_error);  result.setTextColor(getResources().getColor(R.color.error\_color));  return;  }  double firstValue = Double.parseDouble(firstValueStr);  double secondValue = Double.parseDouble(secondValueStr);  double resultValue;  switch (operator) {  case "+": {  resultValue = firstValue + secondValue;  break;  }  case "-": {  resultValue = firstValue - secondValue;  break;  }  case "\*": {  resultValue = firstValue \* secondValue;  break;  }  case "/": {  resultValue = firstValue / secondValue;  break;  }  default: {  resultValue = 0;  break;  }  }  result.setText(Double.toString(resultValue));  result.setTextColor(getResources().getColor(R.color.result\_color));  }  });  }  } |

***Пункт 5:***

|  |
| --- |
| layout/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="410dp"  android:layout\_height="404dp"  android:paddingLeft="20sp"  android:paddingRight="20sp"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent">  <TableRow  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1">  <Button  android:id="@+id/button4"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button3"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button2"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  </TableRow>  <TableRow  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1">  <Button  android:id="@+id/button8"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button7"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button6"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button5"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  </TableRow>  <TableRow  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1">  <Button  android:id="@+id/button12"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button11"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button10"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button9"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  </TableRow>  <TableRow  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1">  <Button  android:id="@+id/button16"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button15"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button14"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  <Button  android:id="@+id/button13"  style="@style/button\_style"  android:text="@string/button\_placeholder" />  </TableRow>  </TableLayout>  <TextView  android:id="@+id/textView"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="15dp"  android:textColor="#000000"  android:textSize="30sp"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />  <Button  android:id="@+id/restart"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="10dp"  android:text="@string/restart\_button\_caption"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout> |

|  |
| --- |
| java/com.example.lab2/MainActivity.java |
| package com.example.task4;  import androidx.appcompat.app.AppCompatActivity;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.TableLayout;  import android.widget.TableRow;  import android.widget.TextView;  import java.util.ArrayList;  public class MainActivity extends AppCompatActivity {  TextView tv;  TableLayout tl;  boolean isTic = true;  boolean isWin = false;  ArrayList<ArrayList<Button>> buttons = new ArrayList<>();  Button restart;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  tv = findViewById(R.id.textView);  tl = findViewById(R.id.tableLayout);  restart = findViewById(R.id.restart);  restart.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  isWin = false;  tv.setText("");  for (int i = 0; i < buttons.size(); i++) {  for (int j = 0; j < buttons.get(i).size(); j++) {  buttons.get(i).get(j).setText(R.string.button\_placeholder);  }  }  }  });  for (int i = 0; i < tl.getChildCount(); i++) {  TableRow row = (TableRow)tl.getChildAt(i);  buttons.add(new ArrayList<Button>());  for (int j = 0; j < row.getChildCount(); j++) {  Button button = (Button) row.getChildAt(j);  buttons.get(i).add(button);  }  }  }  public void processClick(View view) {  Button button = (Button) view;  if (button.getText() != getResources().getString(R.string.tic) && button.getText() != getResources().getString(R.string.tac) && !isWin) {  setTicOrTac(button);  if (checkHorizontalCells() || checkVerticalCells() || checkDiagonalCells()) {  isWin = true;  tv.setText(isTic ? R.string.tac\_won : R.string.tic\_won);  }  }  }  private void setTicOrTac(Button button) {  if (isTic)  button.setText(R.string.tic);  else  button.setText(R.string.tac);  isTic = !isTic;  }  private boolean checkHorizontalCells() {  for (int i = 0; i < buttons.size(); i++) {  String iterator = buttons.get(i).get(0).getText().toString();  boolean res = true;  for (int j = 1; j < buttons.get(i).size() && res; j++) {  String iteratorNext = buttons.get(i).get(j).getText().toString();  res = iteratorNext.equals(iterator);  }  if (res && (iterator.equals(getResources().getString(R.string.tic)) || iterator.equals(getResources().getString(R.string.tac))))  return true;  }  return false;  }  private boolean checkVerticalCells() {  for (int i = 0; i < buttons.size(); i++) {  String iterator = buttons.get(0).get(i).getText().toString();  boolean res = true;  for (int j = 1; j < buttons.get(i).size() && res; j++) {  String iteratorNext = buttons.get(j).get(i).getText().toString();  res = iteratorNext.equals(iterator);  }  if (res && (iterator.equals(getResources().getString(R.string.tic)) || iterator.equals(getResources().getString(R.string.tac))))  return true;  }  return false;  }  private boolean checkDiagonalCells() {  boolean res = true;  String iteratorLeft = buttons.get(0).get(0).getText().toString();  for (int i = 1; i < buttons.size() && res; i++) {  String iteratorNext = buttons.get(i).get(i).getText().toString();  res = iteratorNext.equals(iteratorLeft);  }  if (res && (iteratorLeft.equals(getResources().getString(R.string.tic)) || iteratorLeft.equals(getResources().getString(R.string.tac))))  return true;  res = true;  String iteratorRight = buttons.get(buttons.size() - 1).get(0).getText().toString();  for (int i = 1; i < buttons.size() && res; i++) {  String iteratorNext = buttons.get(buttons.size() - i - 1).get(i).getText().toString();  res = iteratorNext.equals(iteratorRight);  }  return res && (iteratorRight.equals(getResources().getString(R.string.tic)) || iteratorRight.equals(getResources().getString(R.string.tac)));  }  } |

**Контрольні запитання**

1. У роботі використані класичні кнопки (Button), випадаючий список (Spinner) та поля для вводу тексту (EditTextLayout). Перевагою використання одного обробника є економія пам’яті та компактність коду.
2. Класичним прикладом обробника для Button є обробние onClick що викликається після натиску на елемент Button.

**Результати роботи програми**

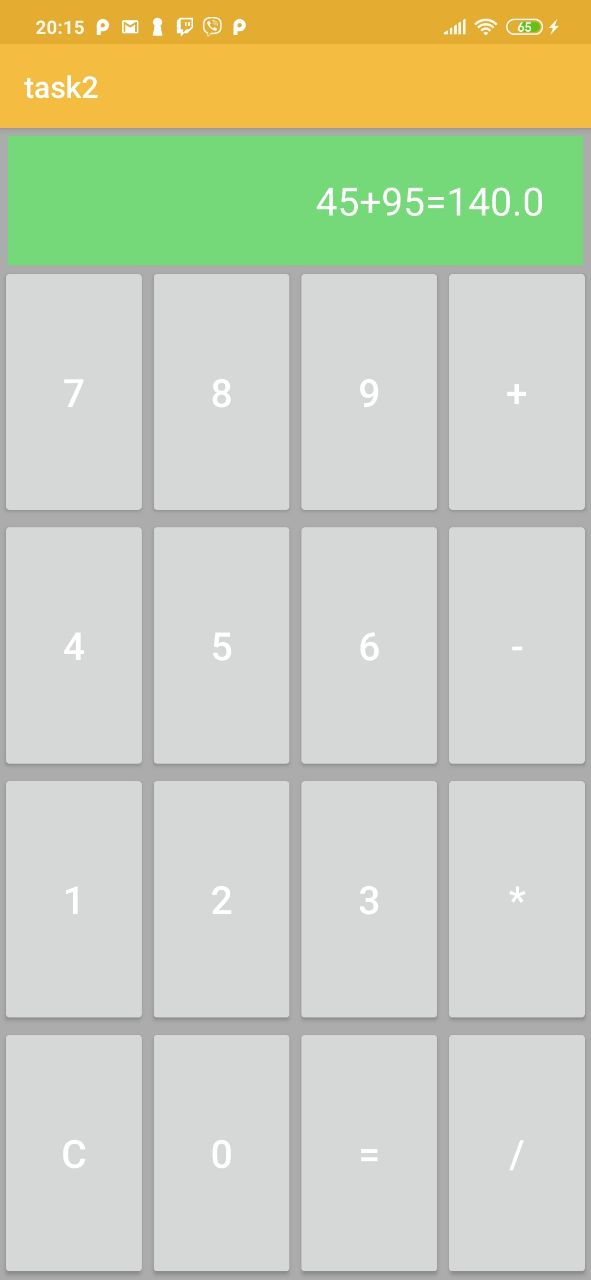


Рис 3.1, результат роботи простого калькулятора



Рис 3.2, результат роботи другої версії калькулятора

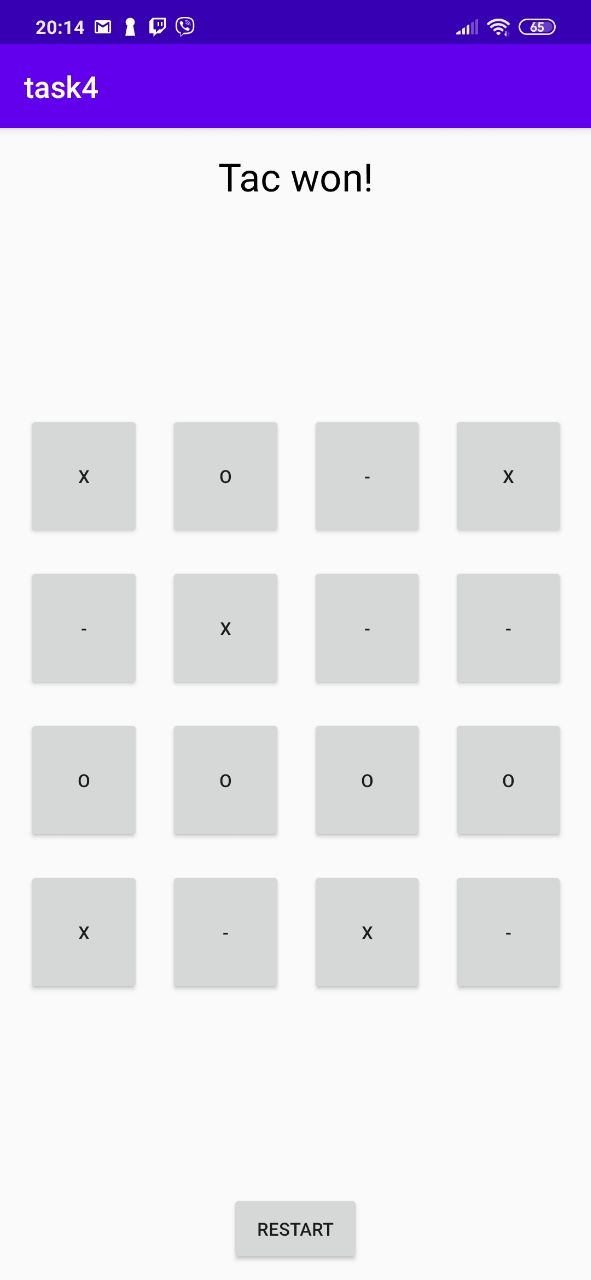


Рис 3.3, демонстрація завершеної гри у хрестики-нулики

**ВИСНОВКИ**

Під час виконання лабораторної роботи ми навчились створювати обробники для подій, що передбачені функціоналом додатку Android Studio.