Відкритий міжнародний університет розвитку людини «Україна»

Інститут комп`ютерних технологій

Кафедра обслуговування комп`ютерних систем та мереж

Курсова робота

з дисципліни «Об`єктовно-орієнтувальне програмування»

на тему: «Калькулятор»

Студента 3 курсу коледжу «Освіта»

групи КІ 22/16

напряму підготовки:

«Комп`ютерна інженерія»

освітньо-кваліфікаційний рівень:

молодший спеціаліст

Іванов А. А.

Керівник: ст. викл. Галісєєв Г. В.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів \_\_\_ оцінка ECTS:\_\_\_

м. Київ – 2018 рік

**Калькулятор на java**

1. Головний клас в якому запускаємо JFrame:

package mainapp;

public class MainApp

{

public static void main(String[] args)

{

new JFrameApp();

}

}

1. Клас JFrameApp наслідуємо від JFrame. Створюємо вікно нашого додатку, вказуємо на його розміри  додаємо в нього типові кнопки (згорнути, закрити, розгорнути на весь экран,  , а також панель JPanelApp на котрій ми будемо розміщувати всі наші кнопки. За допомогою  говоримо, що потрібно показати вікно нашого додатку:

package mainapp;

import javax.swing.JFrame;

public class JFrameApp extends JFrame

{

public JFrameApp()

{

setBounds(100, 100, 265, 400);

setTitle("Calculator");

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

add( new JPanelApp());

setVisible(true);

}

}

1. На панель JPanelApp ми розміщуємо усі наші елементи калькулятора (кнопки, текстове поле) і інтефейс куристувача на кожну кнопку калькулятора:

package mainapp;

import java.awt.PopupMenu;

import java.awt.Font;

import java.awt.TextField;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class JPanelApp extends JPanel

{

JTextField txt1 = null;

int res = 0;

int res1 = 0;

String op = "";

public JPanelApp()

{

try

{

setLayout(null);

// Specifies the position of the element

final TextField txt1 = new TextField();

txt1.setBounds(10, 10, 235, 25);

JButton b0 = new JButton("0");

b0.setBounds(10, 270, 75, 50);

JButton b1 = new JButton("1");

b1.setBounds(10, 190, 50, 50);

JButton b2 = new JButton("2");

b2.setBounds(60, 190, 50, 50);

JButton b3 = new JButton("3");

b3.setBounds(110, 190, 50, 50);

JButton b4 = new JButton("4");

b4.setBounds(10, 110, 50, 50);

JButton b5 = new JButton("5");

b5.setBounds(60, 110, 50, 50);

JButton b6 = new JButton("6");

b6.setBounds(110, 110, 50, 50);

JButton b7 = new JButton("7");

b7.setBounds(10, 40, 50, 50);

JButton b8 = new JButton("8");

b8.setBounds(60, 40, 50, 50);

JButton b9 = new JButton("9");

b9.setBounds(110, 40, 50, 50);

JButton bRes = new JButton("=");

bRes.setBounds(85, 270, 75, 50);

Font bigFont = new Font("serif", Font.BOLD, 22);

bRes.setFont(bigFont);

JButton bPlus = new JButton("+");

bPlus.setBounds(170, 40, 75, 50);

Font bigFontPlus = new Font("serif", Font.BOLD, 22);

bPlus.setFont(bigFontPlus);

JButton bMinus = new JButton("-");

bMinus.setBounds(170, 110, 75, 50);

Font bigFontMinus = new Font("serif", Font.BOLD, 22);

bMinus.setFont(bigFontMinus);

JButton bMulti = new JButton("\*");

bMulti.setBounds(170, 190, 75, 50);

Font bigFontMulti = new Font("serif", Font.BOLD, 22);

bMulti.setFont(bigFontMulti);

JButton bDivision = new JButton("/");

bDivision.setBounds(170, 270, 75, 50);

Font bigFontDivision = new Font("serif", Font.BOLD, 22);

bDivision.setFont(bigFontDivision);

add(txt1);

add(b0);

add(b1);

add(b2);

add(b3);

add(b4);

add(b5);

add(b6);

add(b7);

add(b8);

add(b9);

add(bRes);

add(bPlus);

add(bMinus);

add(bMulti);

add(bDivision);

b1.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 1);

if (res==0) {

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

b2.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 2);

if (res==0) {

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

b3.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 3);

if (res==0) {

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

b4.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 4);

if (res==0)

{

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

b5.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 5);

if (res==0) {

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

b6.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 6);

if (res==0) {

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

b7.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 7);

if (res==0) {

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

b8.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 8);

if (res==0) {

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

b9.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 9);

if (res==0) {

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

b0.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

txt1.setText(txt1.getText() + 0);

if (res==0) {

res = Integer.valueOf(txt1.getText());

} else {

res1 = Integer.valueOf(txt1.getText());

}

}

});

bPlus.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

res = Integer.valueOf(txt1.getText());

txt1.setText("");

op = "+";

}

});

bMinus.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

res = Integer.valueOf(txt1.getText());

txt1.setText("");

op = "-";

}

});

bMulti.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

res = Integer.valueOf(txt1.getText());

txt1.setText("");

op = "\*";

}

});

bDivision.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg1)

{

res = Integer.valueOf(txt1.getText());

txt1.setText("");

op = "/";

}

});

bRes.addActionListener(new ActionListener()

{

@Override

public void actionPerformed(ActionEvent arg0)

{

int num = Integer.valueOf(res);

int num1 = Integer.valueOf(res1);

String strOp = op;

MetodCalc mc = new MetodCalc();

String strRes = String.valueOf(mc.calc(num, strOp, num1 ));

txt1.setText(strRes);

}

});

}

catch (ArithmeticException exception)

{

//System.out.println("Can not divide by zero ");

}

}

}

1. Клас MetodCalc. У ньому я реалізую метод котрий буде виконувати дії над числами в залежності від знака операції:

package mainapp;

public class MetodCalc

{

public int calc ( int n1, String op, int n2)

{

int res = 0;

switch (op)

{

case "+":

res = n1+n2;

break;

case "-":

res = n1-n2;

break;

case "\*":

res = n1\*n2;

break;

case "/":

res = n1/n2;

break;

default:

res = 0;

break;

}

return res;

}

}