Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra Informatică Aplicată

**RAPORT**

Lucrarea de laborator nr.4

Tehnici Avansate de Programare

A efectuat:

st. gr. C-171 D. Melniciuc

A verificat:

dr., conf.univ. M. Oșovschi

Chişinău 2019

**1. Tema lucrării:**

Crearea excepţiilor

**2.**  **Scopul lucrării:**

* Însuşirea modalităţilor de creare şi realizare a excepţiilor în Java;

**3. Etapele de realizare:**

* 1. Realizarea mai mulror tipuri de excepţii;
  2. Realizarea excepţiilor standarte ;
  3. Crearea excepţiilor poprii ;
  4. Crearea interfeţii programului;
  5. Prezentarea lucrării.

***Var*** 5.      Complex 🡨 ComplexArray.

***Codul:***

import java.util.Arrays;

import java.util.regex.\*;

import javax.swing.\*;

import java.awt.GridBagConstraints;

import java.awt.GridBagLayout;

import java.awt.Insets;

import java.awt.event.\*;

import javax.swing.border.\*;

import Swing.SecondFrame;

public class l4 extends JFrame {

private JButton btn;

private JLabel label1, label2, label3, label4, label5, label6;

private JTextField textField1, textField2, textField3, textField4;

private JCheckBox add, substract;

// double realPart1, realPart2, imaginaryPart1, imaginaryPart2;

public static void main(String[] args) {

new l4();

}

public l4() {

this.setLocationRelativeTo(null);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JPanel thePanel = new JPanel();

thePanel.setLayout(new GridBagLayout());

label1 = new JLabel("nr 1: ");

addComp(thePanel, label1, 0, 0, 1, 1, GridBagConstraints.EAST, GridBagConstraints.NONE);

textField1 = new JTextField("", 4);

addComp(thePanel, textField1, 1, 0, 2, 1, GridBagConstraints.EAST, GridBagConstraints.NONE);

label2 = new JLabel("+");

addComp(thePanel, label2, 5, 0, 1, 1, GridBagConstraints.EAST, GridBagConstraints.NONE);

textField2 = new JTextField("", 4);

addComp(thePanel, textField2, 10, 0, 2, 1, GridBagConstraints.EAST, GridBagConstraints.NONE);

label3 = new JLabel("i");

addComp(thePanel, label3, 15, 0, 1, 1, GridBagConstraints.EAST, GridBagConstraints.NONE);

label4 = new JLabel("nr2: ");

addComp(thePanel, label4, 0, 1, 1, 1, GridBagConstraints.WEST, GridBagConstraints.NONE);

textField3 = new JTextField("", 4);

addComp(thePanel, textField3, 1, 1, 3, 1, GridBagConstraints.EAST, GridBagConstraints.NONE);

label5 = new JLabel("+");

addComp(thePanel, label5, 5, 1, 1, 1, GridBagConstraints.EAST, GridBagConstraints.NONE);

textField4 = new JTextField("", 4);

addComp(thePanel, textField4, 10, 1, 3, 1, GridBagConstraints.EAST, GridBagConstraints.NONE);

label6 = new JLabel("i");

addComp(thePanel, label6, 15, 1, 1, 1, GridBagConstraints.WEST, GridBagConstraints.NONE);

JPanel operations = new JPanel();

operations.setLayout(new GridBagLayout());

add = new JCheckBox("+");

substract = new JCheckBox("-");

ButtonGroup operationsGroup = new ButtonGroup();

operationsGroup.add(add);

operationsGroup.add(substract);

Border border = BorderFactory.createTitledBorder("");

operations.setBorder(border);

operations.add(add);

operations.add(substract);

add.setSelected(true);

addComp(thePanel, operations, 0, 3, 20, 1, GridBagConstraints.CENTER, GridBagConstraints.BOTH);

btn = new JButton("submit");

ListenForButton ltForButton = new ListenForButton();

btn.addActionListener(ltForButton);

addComp(thePanel, btn, 0, 4, 18, 1, GridBagConstraints.CENTER, GridBagConstraints.NONE);

this.add(thePanel);

this.pack();

this.setResizable(false);

this.setVisible(true);

}

private class ListenForButton implements ActionListener {

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btn) {

Matcher m1 = Pattern.compile("^[\\-]?(\\d+)\\/(\\d+)$").matcher(textField1.getText());

Matcher m2 = Pattern.compile("^[\\-]?(\\d+)\\/(\\d+)$").matcher(textField2.getText());

Matcher m3 = Pattern.compile("^[\\-]?(\\d+)\\/(\\d+)$").matcher(textField3.getText());

Matcher m4 = Pattern.compile("^[\\-]?(\\d+)\\/(\\d+)$").matcher(textField4.getText());

ComplexFraction complexFraction = new ComplexFraction();

if (!m1.matches() && !m2.matches() && !m3.matches() && !m4.matches()) {

try {

double real1 = Double.parseDouble(textField1.getText());

double imaginary1 = Double.parseDouble(textField2.getText());

double real2 = Double.parseDouble(textField3.getText());

double imaginary2 = Double.parseDouble(textField4.getText());

if (add.isSelected()) {

JOptionPane.showMessageDialog(l4.this, addNumber(real1, imaginary1, real2, imaginary2), "", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(l4.this, substractNumber(real1, imaginary1, real2, imaginary2), "", JOptionPane.INFORMATION\_MESSAGE);

}

} catch(NumberFormatException exept) {

JOptionPane.showMessageDialog(l4.this, "Please enter the Right info", "Error", JOptionPane.ERROR\_MESSAGE);

}

} else {

double realNumerator1 = (m1.matches() && textField1.getText().charAt(0) == '-') ? Double.parseDouble(m1.group(1)) \* -1.0

: ((!m1.matches() && textField1.getText().charAt(0) != '-') ? Double.parseDouble(m1.group(1)) : Double.parseDouble(m1.group(1)));

double realDenominator1 = (m1.matches()) ? Double.parseDouble(m1.group(2)) : 1;

double imaginaryNumerator1 = (m2.matches() && textField2.getText().charAt(0) == '-') ? Double.parseDouble(m2.group(1)) \* -1.0

: ((!m2.matches() && textField2.getText().charAt(0) != '-') ? Double.parseDouble(m2.group(1)) : Double.parseDouble(m2.group(1)));

double imaginaryDenominator1 = (m2.matches()) ? Double.parseDouble(m2.group(2)) : 1;

double realNumerator2 = (m3.matches() && textField3.getText().charAt(0) == '-') ? Double.parseDouble(m3.group(1)) \* -1.0

: ((!m3.matches() && textField3.getText().charAt(0) != '-') ? Double.parseDouble(m3.group(1)) : Double.parseDouble(m3.group(1)));

double realDenominator2 = (m3.matches()) ? Double.parseDouble(m3.group(2)) : 1;

double imaginaryNumerator2 = (m4.matches() && textField4.getText().charAt(0) == '-') ? Double.parseDouble(m4.group(1)) \* -1.0

: ((!m4.matches() && textField4.getText().charAt(0) != '-') ? Double.parseDouble(m4.group(1)) : Double.parseDouble(m4.group(1)));

double imaginaryDenominator2 = (m4.matches()) ? Double.parseDouble(m4.group(2)) : 1;

double[] number1 = {realNumerator1, realDenominator1, imaginaryNumerator1, imaginaryDenominator1};

double[] number2 = {realNumerator2, realDenominator2, imaginaryNumerator2, imaginaryDenominator2};

Arrays.stream(number1).forEach(System.out::print);

System.out.println();

Arrays.stream(number2).forEach(System.out::print);

if (add.isSelected()) {

JOptionPane.showMessageDialog(l4.this, complexFraction.addNumber(number1, number2), "", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(l4.this, complexFraction.substractNumber(number1, number2), "", JOptionPane.INFORMATION\_MESSAGE);

}

}

}

}

}

private void addComp(JPanel thePanel, JComponent comp, int xPos, int yPos, int compWidth, int compHeight, int place, int stretch) {

GridBagConstraints gridConstraints = new GridBagConstraints();

gridConstraints.gridx = xPos;

gridConstraints.gridy = yPos;

gridConstraints.gridwidth = compWidth;

gridConstraints.gridheight = compHeight;

gridConstraints.weightx = 100;

gridConstraints.weighty = 100;

gridConstraints.insets = new Insets(5,5,5,5);

gridConstraints.anchor = place;

gridConstraints.fill = stretch;

thePanel.add(comp, gridConstraints);

}

public double gcd(double a, double b) {

return (b == 0) ? a : gcd(b, a % b);

}

public String addNumber(double r1, double i1, double r2, double i2) {

double realSum = r1 + r2;

double imaginarySum = i1 + i2;

return String.format("%.0f + %.0fi", realSum, imaginarySum);

}

public String substractNumber(double r1, double i1, double r2, double i2) {

double realSubstract = r1 - r2;

double imaginarySubstract = i1 - i2;

return (imaginarySubstract < 0) ? String.format("%.0f - %.0fi", realSubstract, imaginarySubstract \* -1.0) : String.format("%.0f + %.0fi", realSubstract, imaginarySubstract);

}

class ComplexFraction extends l4 {

public String addNumber(double[] n1, double[] n2) {

double realDenominator = ((n1[1] != 1 && n2[1] != 1) && Math.max(n1[1], n2[1]) % Math.min(n1[1], n2[1]) == 0) ? super.gcd(n1[1], n2[1])

: (n1[1] == n2[1] ? n1[1] : n1[1] \* n2[1]);

double realNumerator = n1[0] \* (realDenominator / n1[1]) + n2[0] \* (realDenominator / n2[1]);

double imaginaryDenominator = ((n1[3] != 1 && n2[3] != 1) && Math.max(n1[3], n2[3]) % Math.min(n1[3], n2[3]) == 0) ? super.gcd(n1[3], n2[3])

: (n1[3] == n2[3] ? n1[3] : n1[3] \* n2[3]);

double imaginaryNumerator = n1[2] \* (imaginaryDenominator / n1[3]) + n2[2] \* (imaginaryDenominator / n2[3]);

double realPart = realNumerator / realDenominator;

double imaginaryPart = imaginaryNumerator / imaginaryDenominator;

return (imaginaryNumerator < 0) ? String.format("%.0f/%.0f - %.0f/%.0fi", realNumerator, realDenominator, imaginaryNumerator, imaginaryDenominator)

: String.format("%.0f/%.0f + %.0f/%.0fi", realNumerator, realDenominator, imaginaryNumerator, imaginaryDenominator);

}

public String substractNumber(double[] n1, double[] n2) {

double realDenominator = (Math.max(n1[1], n2[1]) % Math.min(n1[1], n2[1]) == 0) ? super.gcd(n1[1], n2[1])

: (n1[1] == n2[1] ? n1[1] : n1[1] \* n2[1]);

double realNumerator = n1[0] \* (realDenominator / n1[1]) - n2[0] \* (realDenominator / n2[1]);

double imaginaryDenominator = (Math.max(n1[3], n2[3]) % Math.min(n1[3], n2[3]) == 0) ? super.gcd(n1[3], n2[3])

: (n1[3] == n2[3] ? n1[3] : n1[3] \* n2[3]);

double imaginaryNumerator = n1[2] \* (imaginaryDenominator / n1[3]) - n2[2] \* (imaginaryDenominator / n2[3]);

double realPart = realNumerator / realDenominator;

double imaginaryPart = imaginaryNumerator / imaginaryDenominator;

return (imaginaryNumerator < 0) ? String.format("%.0f/%.0f - %.0f/%.0fi", realNumerator, realDenominator, imaginaryNumerator, imaginaryDenominator)

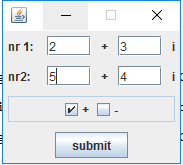
: String.format("%.0f/%.0f + %.0f/%.0fi", realNumerator, realDenominator, imaginaryNumerator, imaginaryDenominator);

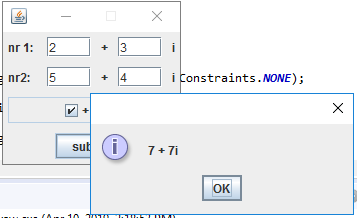
}

}

}

***ScreenShot:***





***Concluzie:***

*Dupa efectuarea laborotorului am dobintid experienta in limbajul de programere java lucrind cu polimorfizm. Am facut programa dupa sarcina data si am creat interfata grafica.*