Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra Informatică Aplicată

**RAPORT**

Lucrarea de laborator nr.7

Tehnici Avansate de Programare

A efectuat:

st. gr. C-171 D. Melniciuc

A verificat:

dr., conf.univ. M. Oșovschi

Chişinău 2019

***1. Tema lucrării:***

Programarea în mediul vizual a proectelor.

***2. Scopul lucrării:***

Însuşirea modalităţilor de creare şi realizare a colecţiilor în Java;

***3. Etapele de realizare:***

- Crearea colecţiilor şi a hărţilor;

- Metode de realizare a colecţiilor şi a hărţilor;

- Metode de complectare şi exstragere a obiectelor din colecţii şi hărţi;

- Crearea interfeţii programului;

- Prezentarea lucrării.

***Var 5.*** Listele I (1. . N) şi U (1. . N),

conţin rezultatele măsurătorilor de tensiune şi curent pentru o rezistenţă necunoscută R.

Găsiţi numărul aproximativ a rezistenţii R.

***Codul:***

}import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

import java.io.\*;

import java.lang.RuntimeException;

import java.util.Timer;

import java.util.TimerTask;

import java.util.\*;

import java.util.stream.Collectors;

class lab7\_2 extends JFrame implements ActionListener {

static JFrame f;

static JButton b;

static JLabel l;

static JLabel l2;

static JLabel l3;

static JLabel l4;

static JTextArea jt;

lab7\_2() {}

public static void main(String[] args) throws InterruptedException {

f = new JFrame("");

l = new JLabel("");

l.setBounds(50,50,100,30);

l2 = new JLabel("");

l2.setBounds(50,50,100,50);

l3 = new JLabel("");

l3.setBounds(50,50,100,50);

l4 = new JLabel("");

l4.setBounds(50,50,100,50);

b = new JButton("submit");

lab7\_2 te = new lab7\_2();

b.addActionListener(te);

JPanel p = new JPanel();

p.add(b);

l.setText("ArrayList & HashMap se vor genera random(0-500)");

p.add(l2);

p.add(l3);

p.add(l4);

p.add(l);

f.add(p);

f.setSize(450, 250);

f.show();

}

public javax.swing.JLabel jLabel1;

public void actionPerformed(ActionEvent e)

{

String s = e.getActionCommand();

if (s.equals("submit")) {

ArrayList<Integer> list = new ArrayList<>();

HashMap<Integer,Integer> map = new HashMap<Integer,Integer>();

Random rand = new Random();

for (int i = 0; i < 6; i++) {

int l = rand.nextInt(500);

int m = rand.nextInt(500);

list.add(l);

map.put(i, m);

}

System.out.println(list);

System.out.println(map);

float sum = 0;

int count = 0;

Iterator<Integer> it = map.keySet().iterator();

while (it.hasNext()) {

int y = it.next();

if (y % 2 == 0) {

sum = (float) (sum + map.get(y));

count++;

}

}

float avg = sum / count;

System.out.println("\nmap avg = " + avg);

float total = 0;

float avg2 = 0;

for(int i = 0; i < list.size(); i++) {

total += list.get(i);

avg2 = total / list.size();

}

System.out.println("list avg:" + avg2);

float zx = (avg + avg2) / 2;

System.out.println("\navg = " + zx + "\n");

l.setText("avg = " + zx);

String listString = list.stream().map(Object::toString).collect(Collectors.joining(", "));

System.out.println("arrlist(String): " + listString);

l2.setText("arrayList: " + listString);

String mapString = map.keySet().stream().map(key -> key + "=" + map.get(key)).collect(Collectors.joining(", ", "{", "}"));

System.out.println("hashMap(String): " + mapString + "\n~~~~~~~~~~~~~~~");

l3.setText("hashMap: " + mapString);

l4.setText("\n ");

try {

File file = new File("in.txt");

FileWriter fw = new FileWriter(file);

PrintWriter pw = new PrintWriter(fw);

pw.println("list: " + listString);

pw.println("map: " + mapString);

pw.println("avg: " + zx);

pw.close();

} catch (IOException exept) {

exept.printStackTrace();

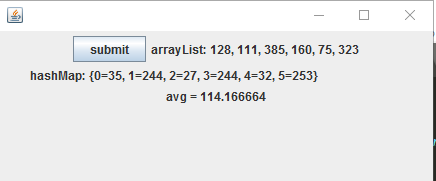
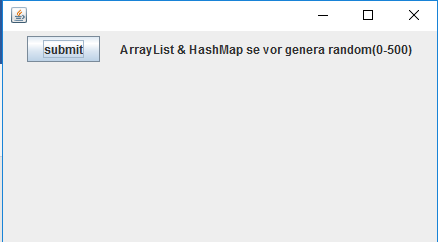
}

}

}

}

***ScreenShot:***



***Concluzie:***

*Dupa efectuarea laborotorului am dobintid experienta in limbajul de programere java lucrind cu colectiile in java. Am utilizat liste, mape si operatii specifice acester colectii.*