**UNIVERSITATEA TEHNICĂ A MOLDOVEI**

**FACULTATEA CALCULATOARE, INFORMATICĂ ȘI MICROELECTRONICĂ**

**CATEDRA CALCULATOARE**

**Raport**

**LUCRARE DE LABORATOR NR.4**

**la Tehnici avansate de programare**

**A realizat: Nicolenco Eugeniu gr. C-162**

**A verificat: Lector univ. Rotaru Lilia**

**Chișinău 2018**

**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.

*Varianta 8*

Se întroduc N parametri de la consolă. Aceşti parametri -sunt elemente ale vectorului detip double, şi in baza vectorului – se creaza clasa cu obiectul DoubleVector. Apoi, programul afisiaza la consola valorile vectorului în formă: Vector: 2.3 5.0 7.3.

***Program Listingul***

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.BorderFactory;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import javax.swing.border.LineBorder;

public class JavaApplication4 extends JFrame implements ActionListener{

public static void SetWindow(){

DoubleVector d = new DoubleVector();

JFrame frame = new JFrame("Fereastra de lucru");

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

JPanel mainPanel = new JPanel();

mainPanel.setLayout(new BorderLayout());

JPanel topPanel = new JPanel();

topPanel.setBorder(BorderFactory.createTitledBorder("Control panel"));

JLabel labelN = new JLabel("N:");

JTextField n= new JTextField();

Dimension standartFieldDimension = new Dimension(30,20);

n.setPreferredSize(standartFieldDimension);

JLabel labelItem = new JLabel("valoarea");

JTextField Item = new JTextField();

Item.setPreferredSize(standartFieldDimension);

Item.setToolTipText("Introduceti Double (Float)");

JButton AddItem = new JButton("Adauga");

JButton RemoveItem = new JButton("Elimina");

JButton Set = new JButton("SetSize");

//----INviz label---//

JLabel Setup = new JLabel();

Setup.setText("false");

Setup.setVisible(false);

//----------------------//

topPanel.add(labelN);

topPanel.add(n);

topPanel.add(Set);

topPanel.add(labelItem);

topPanel.add(Item);

topPanel.add(AddItem);

topPanel.add(RemoveItem);

topPanel.add(Setup);

//-----------------------------

JTextArea Elem = new JTextArea();

Elem.setPreferredSize(new Dimension(150,260));

Elem.setText("true");

JPanel secondcenterPanel = new JPanel();

JButton ShowAll = new JButton("Arata toate");

JButton ShowFirst = new JButton("Arata primul");

JButton ShowLast = new JButton("Arata ultimul");

JButton ShowN = new JButton("Arata N");

JTextField fieldN = new JTextField();

fieldN.setMaximumSize(new Dimension(40,25));

Box VBox = Box.createVerticalBox();

Box HBox = Box.createHorizontalBox();

HBox.add(Box.createRigidArea(new Dimension(90,0)));

HBox.setAlignmentY(CENTER\_ALIGNMENT);

ShowN.setAlignmentX(LEFT\_ALIGNMENT);

fieldN.setAlignmentX(RIGHT\_ALIGNMENT);

HBox.add(ShowN);

HBox.add(Box.createRigidArea(new Dimension(10,0)));

HBox.add(fieldN);

VBox.add(ShowAll);

VBox.add(Box.createRigidArea(new Dimension(0,15)));

VBox.add(ShowFirst);

VBox.add(Box.createRigidArea(new Dimension(0,15)));

VBox.add(ShowLast);

VBox.add(Box.createRigidArea(new Dimension(0,15)));

VBox.add(HBox);

JPanel centerPanel = new JPanel();

centerPanel.setPreferredSize(new Dimension(420,270));

centerPanel.setBorder(BorderFactory.createTitledBorder("Items control"));

centerPanel.add(Elem,BorderLayout.WEST);

centerPanel.add(VBox,BorderLayout.EAST);

JTextArea Exceptions = new JTextArea();

Exceptions.setPreferredSize(new Dimension(420,25));

Exceptions.setOpaque(false);

Exceptions.setFocusable(false);

Exceptions.setFont(new Font("Courier New", Font.ITALIC, 14));

Exceptions.setForeground(Color.RED);

centerPanel.add(Exceptions,BorderLayout.SOUTH);

//------------------------------

JPanel bottomPanel = new JPanel();

bottomPanel.setBorder(BorderFactory.createTitledBorder("Elenents Bar"));

bottomPanel.setMaximumSize(new Dimension(448,60));

JTextArea FinalElem = new JTextArea();

FinalElem.setPreferredSize(new Dimension(420,25));

FinalElem.setFont(new Font("Apple Casual",Font.BOLD,14));

FinalElem.setForeground(Color.MAGENTA);

bottomPanel.add(FinalElem,BorderLayout.NORTH);

mainPanel.add(topPanel,BorderLayout.NORTH);

mainPanel.add(centerPanel,BorderLayout.CENTER);

mainPanel.add(bottomPanel,BorderLayout.SOUTH);

frame.getContentPane().add(mainPanel);

frame.setPreferredSize(new Dimension(450, 520));

frame.pack();

frame.setLocationRelativeTo(null);

frame.setVisible(true);

//-------------Functionals-----------

Set.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

try{

if(n.getText().equals("")){

throw new MyException("Capsula goala");

}else if(n.getText().equals(Integer.toString(0))){

throw new MyException("Zero nu poate fi volumul vectorului");

}else if(n.getText().contains(" ")){

throw new MyException("Capsula contine spatii");

}else if(n.getText().matches("[0-9]+")==false){

throw new MyException("Capsula contine date diferite de numar int");

}

else{

int x = Integer.parseInt(n.getText());

d.init(x);

n.setFocusable(false);

n.setBorder(BorderFactory.createEmptyBorder());

Exceptions.setText(" ");

Set.setVisible(false);

Setup.setText("true");

}

}catch(MyException ex){

n.requestFocus();

n.selectAll();

n.setBorder(new LineBorder(Color.RED,1));

Exceptions.setText(ex.getMessage());

}

}

});

AddItem.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

try{

if(Setup.getText().equals("true")){

try{

if(d.getIndex()>=Integer.parseInt(n.getText())){

throw new Exception("Depasita Limita de valori");

}else if(Item.getText().matches("[0-9.]+")==false){

throw new Exception("Nu aveti element tip Double");

}else{

d.add(Double.parseDouble(Item.getText()));

Elem.setText(d.Show());

Item.requestFocus();

Item.selectAll();

}

}catch(Exception ep){

Exceptions.setText(ep.getMessage());

}

}else{

throw new MyException("Nu este setat N:");

}

}catch(MyException ex){

n.requestFocus();

n.selectAll();

n.setBorder(new LineBorder(Color.RED,1));

Exceptions.setText(ex.getMessage());

}

}

});

RemoveItem.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

try{

if(Setup.getText().equals("true")){

try{

if(d.getIndex()==0){

throw new Exception("Nu Exista valori");

}else if(Item.getText().matches("[0-9.]+")==false){

throw new Exception("Nu aveti element tip Double");

}else{

try{

if(d.exists(Double.parseDouble(Item.getText()))){

d.erase(Double.parseDouble(Item.getText()));

Item.requestFocus();

Item.selectAll();

Item.setBorder(BorderFactory.createLineBorder(Color.white));

Elem.setText(d.Show());

}else{

Elem.requestFocus();

Elem.selectAll();

throw new MyException("Nu exista asa element in tablou");

}

}catch(MyException x){

Item.requestFocus();

Item.setBorder(new LineBorder(Color.RED,1));

Exceptions.setText(x.getMessage());

}

}

}catch(Exception ep){

Exceptions.setText(ep.getMessage());

Item.selectAll();

Item.requestFocus();

}

}else{

throw new MyException("Tabloul este gol");

}

}catch(MyException ex){

n.requestFocus();

n.selectAll();

n.setBorder(new LineBorder(Color.RED,1));

Exceptions.setText(ex.getMessage());

}

}

});

ShowAll.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

try{

if(Setup.getText().equals("true")){

if(d.getIndex()>0){

String line="";

for(int i=0;i<d.getIndex();i++){

line+=d.Show(i)+" ";

}

FinalElem.setText(line);

}else throw new MyException("Tablou fara elemente");

}else throw new MyException("Tablou neinitializat");

}catch(MyException ex){

Exceptions.setText(ex.getMessage());

}

}

});

ShowFirst.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

try{

if(Setup.getText().equals("true")){

if(d.getIndex()>0){

String line="";

line+=d.Show(0)+" ";

FinalElem.setText(line);

}else throw new MyException("Tablou fara elemente");

}else throw new MyException("Tablou neinitializat");

}catch(MyException ex){

Exceptions.setText(ex.getMessage());

}

}

});

ShowLast.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

try{

if(Setup.getText().equals("true")){

if(d.getIndex()>0){

String line="";

line+=d.Show(d.getIndex()-1)+" ";

FinalElem.setText(line);

}else throw new MyException("Tablou fara elemente");

}else throw new MyException("Tablou neinitializat");

}catch(MyException ex){

Exceptions.setText(ex.getMessage());

}

}

});

ShowN.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

try{

if(Setup.getText().equals("true")){

if(d.getIndex()>0){

try{

if(fieldN.getText().equals("")||fieldN.getText().contains(" ")||fieldN.getText().matches("[0-9]+")==false)

throw new Exception("Parametri incorecti");

else if(Integer.parseInt(fieldN.getText())>=d.getIndex()){

throw new Exception("Index in afara tabloului");

}else if(Integer.parseInt(fieldN.getText())<0){

throw new Exception("Valoare negativa nu se primeste");

}else{

FinalElem.setText(d.Show(Integer.parseInt(fieldN.getText())));

fieldN.requestFocus();

fieldN.selectAll();

}

}catch(Exception oe){

Exceptions.setText(oe.getMessage());

}

}else throw new MyException("Tablou fara elemente");

}else throw new MyException("Tablou neinitializat");

}catch(MyException ex){

Exceptions.setText(ex.getMessage());

}

}

});

}

public static void main(String[] args) {

javax.swing.SwingUtilities.invokeLater(new Runnable() {

public void run() {

JFrame.setDefaultLookAndFeelDecorated(true);

SetWindow();

}

});

}

public void actionPerformed(ActionEvent ae) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

}

class DoubleVector{

double[] val;

int index;

int maxAdmisible;

void init(int maxSize){

val = new double[maxSize];

maxAdmisible = maxSize;

}

void add(double value){

val[index++]=value;

System.out.println(val[index-1]);

}

boolean erase(double value){

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

if(val[i]==value){

for(int j=i;j<index-1;j++){

val[j]=val[j+1];

}

double[] aux=new double[maxAdmisible];

index--;

for(int j=0;j<index;j++){

aux[j]=val[j];

}

val=new double[maxAdmisible];

for(int j=0;j<index;j++){

val[j]=aux[j];

}

return true;

}

}

return false;

}

boolean equals(DoubleVector x) throws MyException{

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

if(val[i]!=x.val[i])return false;

}

return true;

}

String Show(){

String line =" ";

for(int j = 0;j<index;j++){

line+=Double.toString(val[j])+"\n ";

}

return line;

}

String Show(int i){

double x = val[i]\*10;

int y=(int)x;

double z = (double)y/10;

String out = Double.toString(z);

return out;

}

double getValue(int i){

return val[i];

}

int getIndex(){

return index;

}

boolean exists(double i){

for(int j =0; j<index;j++){

if(val[j]==i)return true;

}

return false;

}

}

class MyException extends Exception{

public MyException(String message, Throwable cause){

super(message,cause);

}

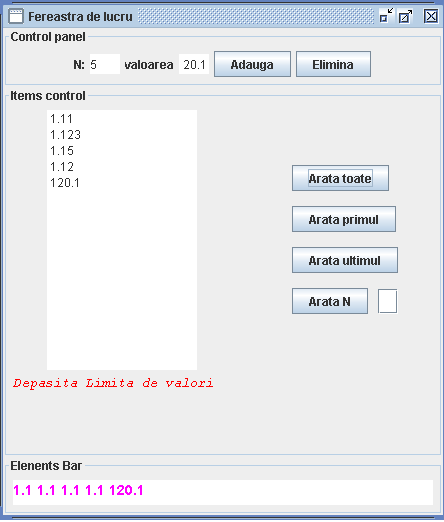
public MyException(String message){

super(message);

}

}

***Rezultatul executarii:***



**Concluzie:**

In urma efectuarii acestei lucrari am insusit modalitatile de creare si realizare a exceptiilor in Java, am creat exceptii proprii care m-au ajutat la indeplinirea conditiilor acestei lucrari.