Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

**Отчёт**

по лабораторной работе № 3

## на тему «Обработка исключительных ситуаций»

по дисциплине «Программирование на языке Java»

Вариант № 4

выполнили: ст-ты гр. 22ВВП1

Китаев Я.Е.

Рябов С.И.

Проверили:

Юрова О.В.

Карамышева Н.С.

2025

**Цель работы:** изучить механизм обработки исключительных ситуаций.

**Задание:** Модифицировать приложение из предыдущей лабораторной работы, реализовав проверку вводимых данных с использованием механизма исключений. Необходимо создать свой класс, унаследованный от класса Exception, и генерировать исключение, если возникает попытка создать экземпляр класса RecIntegral со значениями, не являющимися числами в диапазоне от 0,000001 до 1000000. В качестве обработки исключения необходимо выводить диалог, содержащий предупреждение о некорректности введенных данных.

**Ход работы**

Добавили наследуемый от extends класс для обработки ошибок  
 public class InvalidValueException extends Exception {

public InvalidValueException(String message) {

super(message);

}

}

Создали функцию, которая проверяет вводимые значения  
 private double parseAndValidate(String value) throws InvalidValueException {

try {

double num = Double.parseDouble(value);

if (num < 0.000001 || num > 1000000) {

throw new InvalidValueException("Число должно быть в диапазоне от 0.000001 до 1000000.");

}

return num;

} catch (NumberFormatException e) {

throw new InvalidValueException("Введено некорректное число: " + value);

}

}  
 Добывали обработчики try catch на кнопки

**Листинг программы**

import javax.swing.table.DefaultTableModel;

import javax.swing.JOptionPane;

import java.util.LinkedList;

public class NewJFrame extends javax.swing.JFrame {

DefaultTableModel model;

public NewJFrame() {

initComponents();

model = (DefaultTableModel) jTable1.getModel();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jScrollPane4 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jScrollPane3 = new javax.swing.JScrollPane();

StepTxt = new javax.swing.JTextPane();

Delite = new javax.swing.JButton();

Result = new javax.swing.JButton();

Add = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

DownLimit = new javax.swing.JTextPane();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jScrollPane2 = new javax.swing.JScrollPane();

UpLimit = new javax.swing.JTextPane();

Clear = new javax.swing.JButton();

Rec = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setBackground(new java.awt.Color(153, 0, 102));

setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

jPanel1.setBackground(new java.awt.Color(45, 50, 80));

jPanel1.setPreferredSize(new java.awt.Dimension(577, 400));

jTable1.setBackground(new java.awt.Color(103, 111, 157));

jTable1.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jTable1.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N

jTable1.setForeground(new java.awt.Color(255, 255, 255));

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"нижняя граница", "верхняя граница", "шаг", "результат"

}

) {

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jTable1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

jTable1.setSelectionBackground(new java.awt.Color(249, 209, 122));

jTable1.setShowHorizontalLines(true);

jTable1.setShowVerticalLines(true);

jTable1.getTableHeader().setReorderingAllowed(false);

jScrollPane4.setViewportView(jTable1);

StepTxt.setBackground(new java.awt.Color(235, 235, 235));

jScrollPane3.setViewportView(StepTxt);

Delite.setBackground(new java.awt.Color(249, 209, 122));

Delite.setText("удалить ");

Delite.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

Delite.setMaximumSize(new java.awt.Dimension(56, 22));

Delite.setMinimumSize(new java.awt.Dimension(56, 22));

Delite.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

DeliteActionPerformed(evt);

}

});

Result.setBackground(new java.awt.Color(249, 209, 122));

Result.setText("вычислить");

Result.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

Result.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

ResultActionPerformed(evt);

}

});

Add.setBackground(new java.awt.Color(249, 209, 122));

Add.setText("добавить");

Add.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));

Add.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

AddActionPerformed(evt);

}

});

DownLimit.setBackground(new java.awt.Color(235, 235, 235));

jScrollPane1.setViewportView(DownLimit);

jLabel1.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 255, 255));

jLabel1.setText("Верхняя граница");

jLabel2.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N

jLabel2.setForeground(new java.awt.Color(255, 255, 255));

jLabel2.setText("Нижняя граница");

jLabel3.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N

jLabel3.setForeground(new java.awt.Color(255, 255, 255));

jLabel3.setText("Шаг");

UpLimit.setBackground(new java.awt.Color(235, 235, 235));

jScrollPane2.setViewportView(UpLimit);

Clear.setBackground(new java.awt.Color(249, 209, 122));

Clear.setText("очистить");

Clear.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

Clear.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

ClearActionPerformed(evt);

}

});

Rec.setBackground(new java.awt.Color(249, 209, 122));

Rec.setText("заполнить");

Rec.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

Rec.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

RecActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGap(34, 34, 34)

.addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(21, 21, 21)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(42, 42, 42)

.addComponent(jLabel3)))

.addGap(23, 23, 23))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addComponent(jLabel1)

.addGap(18, 18, 18)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(87, 87, 87)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(Delite, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(Result, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(Add, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 98, Short.MAX\_VALUE))))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(111, 111, 111)

.addComponent(Clear, javax.swing.GroupLayout.PREFERRED\_SIZE, 98, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(104, 104, 104)

.addComponent(Rec, javax.swing.GroupLayout.PREFERRED\_SIZE, 98, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(46, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(25, 25, 25)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel1))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3)))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(Add)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(Delite, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(Result)))

.addGap(25, 25, 25)

.addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED\_SIZE, 171, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(Clear, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(Rec, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(31, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 532, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 6, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

public class InvalidValueException extends Exception {

public InvalidValueException(String message) {

super(message);

}

}

public class RecIntegral {

private LinkedList<String[]> list = new LinkedList<>();

public void RecTable(String DownL, String UpL, String Step, String Result) throws InvalidValueException {

double down = parseAndValidate(DownL);

double up = parseAndValidate(UpL);

double step = parseAndValidate(Step);

if (down >= up) {

throw new InvalidValueException("Нижний предел должен быть меньше верхнего.");

}

if (step <= 0) {

throw new InvalidValueException("Шаг должен быть положительным числом.");

}

list.add(new String[]{DownL, UpL, Step, Result});

}

public String[][] GetTable() {

String[][] records = new String[list.size()][4];

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

records[i] = list.get(i);

}

return records;

}

public void DelElemList(int NumberElem) {

list.remove(NumberElem);

}

public void ChangeValue(int NumberElem, String DownL, String UpL, String Step, String Result) throws InvalidValueException {

parseAndValidate(DownL);

parseAndValidate(UpL);

parseAndValidate(Step);

list.set(NumberElem, new String[]{DownL, UpL, Step, Result});

}

public boolean hasRecords() {

return !list.isEmpty();

}

private double parseAndValidate(String value) throws InvalidValueException {

try {

double num = Double.parseDouble(value);

if (num < 0.000001 || num > 1000000) {

throw new InvalidValueException("Число должно быть в диапазоне от 0.000001 до 1000000.");

}

return num;

} catch (NumberFormatException e) {

throw new InvalidValueException("Введено некорректное число: " + value);

}

}

}

RecIntegral SaveTable = new RecIntegral();

private void ResetField(){

StepTxt.setText("");

DownLimit.setText("");

UpLimit.setText("");

}

private void DeliteActionPerformed(java.awt.event.ActionEvent evt) {

if (jTable1.getSelectedRow() > -1){

int getRow = jTable1.getSelectedRow();

SaveTable.DelElemList(getRow);

model.removeRow(jTable1.getSelectedRow());

} else {

JOptionPane.showMessageDialog(NewJFrame.this, "Выбери строку для удаления");

}

}

private void ResultActionPerformed(java.awt.event.ActionEvent evt) {

if (jTable1.getSelectedRow() > -1){

int getRow = jTable1.getSelectedRow();

double StepD = Double.parseDouble(jTable1.getValueAt(getRow, 2).toString());

double DownLimitD = Double.parseDouble(jTable1.getValueAt(getRow, 0).toString());

double UpLimitD = Double.parseDouble(jTable1.getValueAt(getRow, 1).toString());

double Square = 0;

// if (StepD <= 0 || UpLimitD < 0 || DownLimitD < 0 || UpLimitD < DownLimitD ) {

// JOptionPane.showMessageDialog(NewJFrame.this, "Введены некорректные значения");

// return;

// }

for (double i = DownLimitD; i < UpLimitD; i += StepD) {

Square += i + StepD > UpLimitD ? (UpLimitD - i) \* (Math.tan(i) + Math.tan(UpLimitD)) / 2 : (StepD / 2) \* (Math.tan(i) + Math.tan(i + StepD));

}

String SquareStr = String.format("%.5f", Square);

try {

SaveTable.ChangeValue(getRow, jTable1.getValueAt(getRow, 0).toString(),

jTable1.getValueAt(getRow, 1).toString(),

jTable1.getValueAt(getRow, 2).toString(),

SquareStr);

jTable1.setValueAt(Square, getRow, 3);

} catch (InvalidValueException e) {

JOptionPane.showMessageDialog(NewJFrame.this, e.getMessage(), "Ошибка ввода", JOptionPane.WARNING\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(NewJFrame.this, "Выбери строку для вычисления");

}

}

private void AddActionPerformed(java.awt.event.ActionEvent evt) {

if (DownLimit.getText().isEmpty() || UpLimit.getText().isEmpty() || StepTxt.getText().isEmpty() ) {

JOptionPane.showMessageDialog(NewJFrame.this, "Введены некорректные значения");

return;

}

try {

SaveTable.RecTable(DownLimit.getText(), UpLimit.getText(), StepTxt.getText(), "0");

model.addRow(new Object[]{DownLimit.getText(), UpLimit.getText(), StepTxt.getText(), 0});

ResetField();

} catch (InvalidValueException e) {

JOptionPane.showMessageDialog(NewJFrame.this, e.getMessage(), "Ошибка ввода", JOptionPane.WARNING\_MESSAGE);

}

}

private void ClearActionPerformed(java.awt.event.ActionEvent evt) {

model = (DefaultTableModel) jTable1.getModel();

model.setRowCount(0);

jTable1.setModel(model);

}

private void RecActionPerformed(java.awt.event.ActionEvent evt) {

if (SaveTable.hasRecords()){

String[][] GetTb = SaveTable.GetTable();

for ( int i = 0; i < GetTb.length;i++){

model.addRow(GetTb[i]);

}

} else {

JOptionPane.showMessageDialog(NewJFrame.this, "Список пуст");

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new NewJFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton Add;

private javax.swing.JButton Clear;

private javax.swing.JButton Delite;

private javax.swing.JTextPane DownLimit;

private javax.swing.JButton Rec;

private javax.swing.JButton Result;

private javax.swing.JTextPane StepTxt;

private javax.swing.JTextPane UpLimit;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JScrollPane jScrollPane4;

private javax.swing.JTable jTable1;

// End of variables declaration

}

**Результат работы программы**

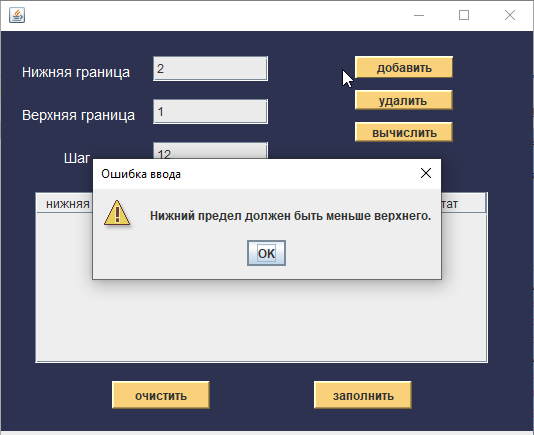


Рисунок 1 – Обработка ошибки

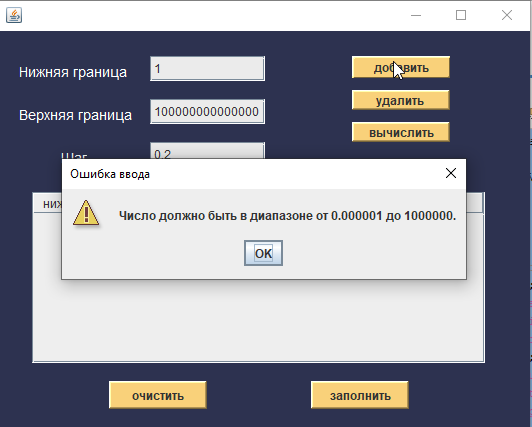


Рисунок 2 – Обработка ошибки

**Вывод:** изучили механизм обработки исключительных ситуаций.