Министерство науки и высшего образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ОТЧЁТ

по лабораторной работе № 3
по курсу «Разработка кроссплатформенных приложений»
Вариант 4

Выполнили
студенты группы 22ВОЭ1
Брюзгин А. С.
Тихонов Д. А.

Приняли
Юрова О. В.

Пенза 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.

Исходный код программы

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package my.contacteditor;

import static java.lang.Math.tan;
import java.util.LinkedList;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author User
 */
```

```
public class ContactEditorUI extends javax.swing.JFrame {

    private LinkedList <RecIntegral> records = new LinkedList();

    private DefaultTableModel tModel;

    int i = 0;
    /**
     * Creates new form ContactEditorUI
     */
    public ContactEditorUI() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jTextField3 = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        jButton4 = new javax.swing.JButton();
        jButton5 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jTextField1.setText("0");

        jTextField2.setText("0");
```

```
jTextField3.setText("0");

jLabel1.setText("upper_limit");

jLabel2.setText("lower_limit");

jLabel3.setText("Step");

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Upper_limit", "Lower_limit", "Step", "Result"
    }
) {
    Class[] types = new Class [] {
        java.lang.Double.class, java.lang.Double.class, java.lang.Double.class, java.lang.Double.class
    };
    boolean[] canEdit = new boolean [] {
        false, false, false, false
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
jTable1.setRowHeight(30);
jScrollPane1.setViewportView(jTable1);
if (jTable1.getColumnModel().getColumnCount() > 0) {
    jTable1.getColumnModel().getColumn(0).setResizable(false);
    jTable1.getColumnModel().getColumn(1).setResizable(false);
    jTable1.getColumnModel().getColumn(2).setResizable(false);
    jTable1.getColumnModel().getColumn(3).setResizable(false);
}

jButton1.setText("ADD");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("COUNT");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    jButton3.setText("DELETE");
    jButton3.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton3ActionPerformed(evt);
        }
    });

    jButton4.setText("Add List");
    jButton4.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton4ActionPerformed(evt);
        }
    });

    jButton5.setText("Delete List");
    jButton5.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton5ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addComponent(jScrollPane1,  javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 477, Short.MAX_VALUE)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
                        .addComponent(jLabel1)
                                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 122, javax.swing.GroupLayout.PREFERRED_SIZE))
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 53, Short.MAX_VALUE)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 122, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel2))
                    .addGap(58, 58, 58)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel3)
                                .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, 122, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                                .addComponent(jButton4, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(jButton1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, 122, Short.MAX_VALUE))
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 122, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(54, 54, 54)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                .addComponent(jButton5, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(jButton3, javax.swing.GroupLayout.DEFAULT_SIZE, 122, Short.MAX_VALUE))))
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(jLabel2)
                    .addComponent(jLabel3))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                                        .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 143,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                      .addComponent(jButton1)
                      .addComponent(jButton2)
                      .addComponent(jButton3))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                      .addComponent(jButton4)
                      .addComponent(jButton5))
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                );

                pack();
            }// </editor-fold>

            private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

                Double ul = Double.valueOf(jTextField1.getText());
                Double ll = Double.valueOf(jTextField2.getText());
                Double stp = Double.valueOf(jTextField3.getText());

               try{
                records.add(new RecIntegral(ul,ll,stp));
                tModel = (DefaultTableModel) jTable1.getModel();
                tModel.addRow(new Object[] {ul, ll, stp});
                 i++;
               }

                catch (ErrRecIntegralVal e) {

                                        JOptionPane.showMessageDialog(null, e.getMessage(), "Ошибка ввода",
JOptionPane.WARNING_MESSAGE);
            }

            }

            private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

                tModel = (DefaultTableModel) jTable1.getModel();
```

```
    int rowNum = jTable1.getSelectedRow();

    if (rowNum == -1){
        JOptionPane.showMessageDialog(null, "Выберите строку для удаления");
    }else{
        tModel.removeRow(rowNum);
        records.remove(rowNum);
        i--;
    }

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    records.clear();

   for (int j = 0; j != i; j++){
    Double ul = (Double) jTable1.getValueAt(j, 0);
    Double ll = (Double) jTable1.getValueAt(j, 1);
    Double stp = (Double) jTable1.getValueAt(j, 2);

    Double s = 0.0, h = 0.0, osn1 = 0.0, osn2 = 0.0, rez = 0.0, n = 0.0;
    Double stp_ost = 0.0, n1 = 0.0, st = 0.0;
    int k = 0;

    n = (ul - ll) / stp;
    n1 = n/1;

    stp_ost = stp * (n - Math.floor(n));


    if (n1 == 0.0){
        for (Double x = ll; x < ul; x += stp){
            rez += (tan(x) + tan(x + stp)) * stp/2;
    }
    }
    else if(n1 != 0.0){
      while (k < n){
      osn1 = tan(ll + st);
      osn2 = tan(ll + st + stp) /*+ tan(stp)*/;
      h = stp;
```

```
                s = ((osn1 + osn2) * h) / 2;
                rez = rez + s;

                st = st + stp;

                k++;
                }
                 osn1 = tan(ll + st);
                 osn2 = tan(ll + stp_ost) /*+ tan(stp_ost)*/;
                 h = stp_ost;

                 s = ((osn1 + osn2) * h) / 2;
                 rez = rez + s;

              }

              jTable1.setValueAt(rez, j, 3);

              records.add(new RecIntegral (ul,ll,stp,rez));

             }

          }

          private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
              tModel.setRowCount(0);

              for (RecIntegral record : records) {
                                          tModel.addRow(new    Object[]{record.getValueUl(),    record.getValueLl(),
record.getValueStp(),record.getValueRez()});
              }

          }

          private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
              tModel.setRowCount(0);
              records.clear();
          }

          /**
           * @param args the command line arguments
           */
          public static void main(String args[]) {
```

```
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
                                                    for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(ContactEditorUI.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(ContactEditorUI.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(ContactEditorUI.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(ContactEditorUI.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ContactEditorUI().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
```

```
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    // End of variables declaration


}
```

Выполнение программы

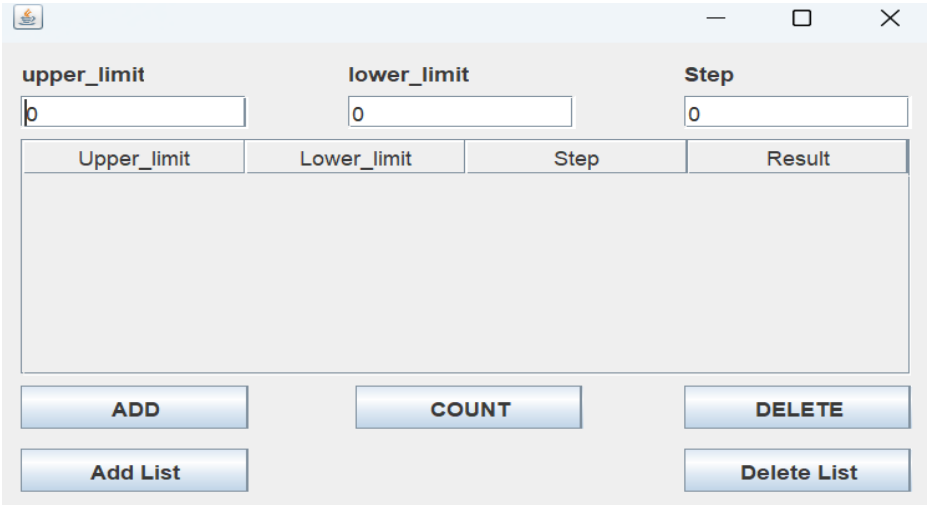


Рисунок 1 — Добавление кнопок Очистить/Заполнить

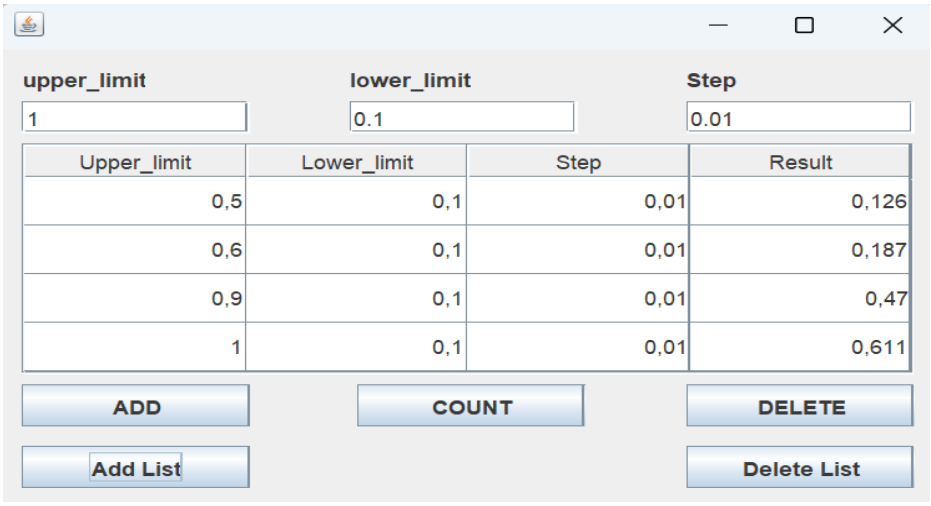
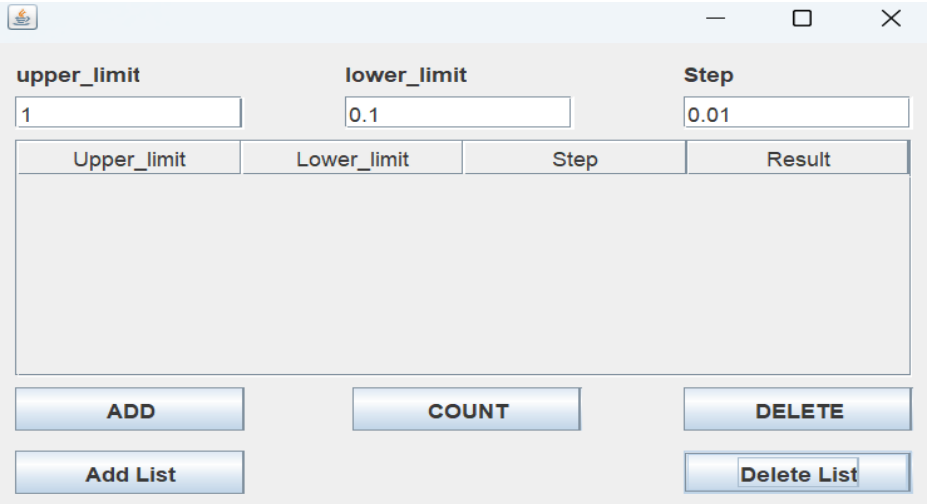


Рисунок 2 — Заполнение

Рисунок 3 — Очищение



Ход работы

```
public class RecIntegral {

    private double ul = 0, ll = 0, stp = 0, rez =0;

    private boolean ValidValue(double value) {
        return value >= 0.000001 && value <= 1000000;
    }

    public RecIntegral (double ul, double ll, double stp) throws ErrRecIntegralVal {
        if (!ValidValue(ul) || !ValidValue(ll) || !ValidValue(stp)) {
            throw new ErrRecIntegralVal("Значения должны быть в диапазоне от 0.000001 до
1000000");
        }
        this.ul = ul;
        this.ll = ll;
        this.stp = stp;
    }

    public RecIntegral( double ul, double ll, double stp, double rez) {
        this.ul = ul;
        this.ll = ll;
        this.stp = stp;
        this.rez = rez;

    }
```

Отлов ошибки

```
try{
    records.add(new RecIntegral(ul,ll,stp));
    tModel = (DefaultTableModel) jTable1.getModel();
    tModel.addRow(new Object[] {ul, ll, stp});
     i++;
    }
        catch (ErrRecIntegralVal e) {

                JOptionPane.showMessageDialog(null, e.getMessage(), "Ошибка ввода", JOptionPane.WARNING_MESSAGE);
        throw new RuntimeException("Error" + e.getMessage(), e);
      }
```

Пояснение к тексту программы(проверка ввода)

Строка 2: Объявление и инициализация полей класса.

Строка 3-5: Проверка введенных значений.

Строка 6-13: Создание конструктора класса, в котором проверяется правильность введенных данных, и если данные некорректны, то выводится ошибка, иначе данные заносятся в поля класса.

Строка 14-19: Перегрузка конструктора класса для записи значений и результатов рассчета.

Пояснение к тексту программы(отлов ошибки)

Строки 1 - 6: Проверка кода, который может вызвать исключение.

Строки 6 – 9: Перехват исключения и вывод сообщения об ошибке в диалоговое окно.

Вывод

Изучен механизм обработки исключительных ситуаций. Написана программа.