

Министерство науки и высшего образования РФ

Пензенский государственный университет

Кафедра «Вычислительная техника»

## **Отчёт**

по лабораторной работе №3

по курсу «Разработка кроссплатформенных приложений»

на тему «Обработка исключительных ситуаций»

Вариант 9

Выполнили студенты группы 23ВВИ2:

Юров Д.М.

Лисов Е.А.

Кочегин В.В.

Приняли:

к.т.н., доцент Юрова О.В.

к.т.н., доцент Карамышева Н.С.

Пенза 2026

**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:

### 1. Модификация программы

Создал класс «`OutOfLimitException`», унаследованный от класса «`Exception`» (рис. 1).

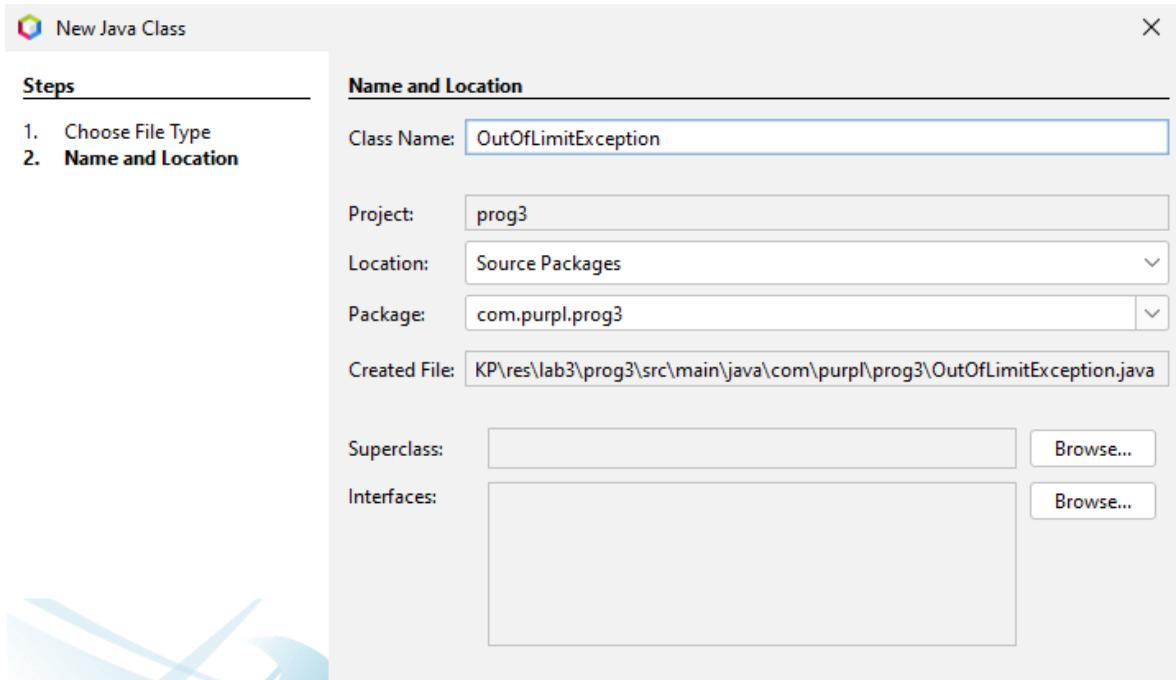


Рисунок 1 - создание файла класса

Добавил конструктор, принимающий сообщение для вывода во время возникновения ошибки – то есть во время создания экземпляра класса «`OutOfLimitException`» (рис. 2).

```
public class OutOfLimitException extends Exception {  
    public OutOfLimitException(String message){  
        super(message);  
    }  
}
```

Рисунок 2 - добавление конструктора

Модифицировал конструктор класса «ResIntegral» - добавил указание об использовании обработки ошибки (класса «OutOfLimitException») и проверку условия по заданию – значение переменной «*a*» должно быть больше «0.000001» и значение переменной «*b*» должно быть меньше «1000000» (рис. 3).

```
public ResIntegral(double a, double b, double h) throws OutOfLimitException {
    if(a < 0.000001 && b > 1000000) {
        throw new OutOfLimitException("Неверные значения а и b - число должно быть в диапазоне: 0,000001-1000000");
    }

    this.a = a;
    this.b = b;
    this.h = h;
}
```

Рисунок 3 - условие вывода ошибки

Так как создание нового экземпляра класса «ResIntegral» и так выполняется внутри конструкции «try{} ... catch{}», нужно добавить еще один блок «catch» для обработки конкретной ошибки (рис. 4).

```
} catch (OutOfLimitException ex) {
    JOptionPane.showMessageDialog(this,
        ex.getMessage(),
        "Ошибка", JOptionPane.WARNING_MESSAGE);
} catch (Exception ex) {
    JOptionPane.showMessageDialog(this,
        "Ошибка: " + ex.getMessage(),
        "Ввод данных", JOptionPane.WARNING_MESSAGE);
}
```

Рисунок 4 - обработка ошибки

## 2. Тестирование и отладка программы

Проверка вывода ошибки при попытке создания экземпляра класса «ResIntegral» со значением переменной «a» менее «0.000001» (рис. 5).

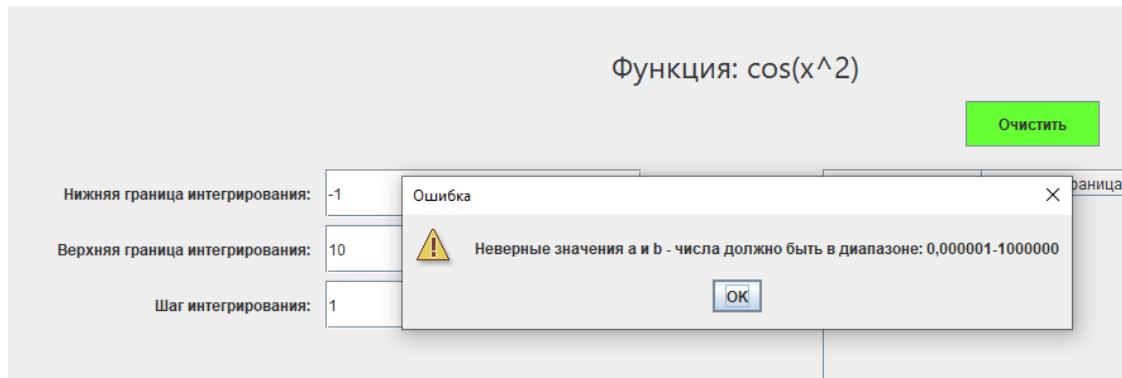


Рисунок 5 - вывод информации об ошибке 1

Тест пройден успешно.

Проверка вывода ошибки при попытке создания экземпляра класса «ResIntegral» со значением переменной «b» более «1000000» (рис. 6).

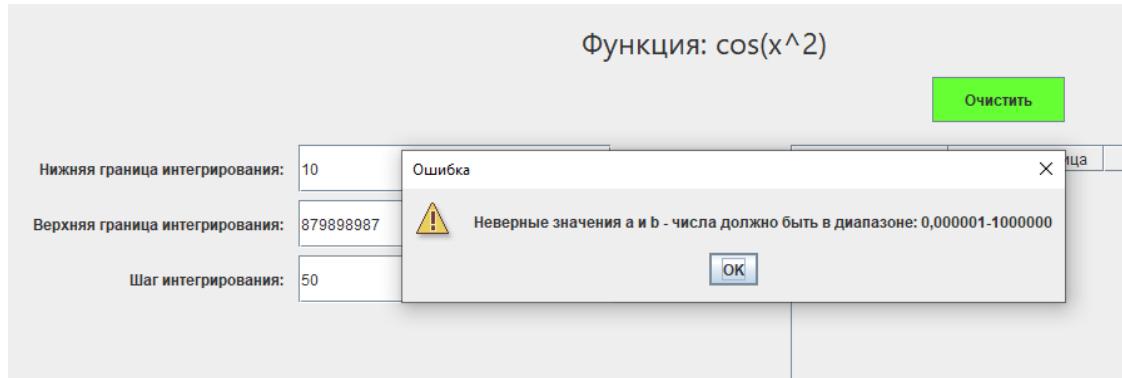


Рисунок 6 - вывод информации об ошибке 2

Тест пройден успешно.

Все тесты пройдены успешно, программа работает корректно.

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

## **Приложение 1 – содержимое файла «Prog3.java»**

```
package com.purpl.prog3;

public class Prog3 {
    public static void main(String[] args) {
        javax.swing.SwingUtilities.invokeLater(() -> new JFrameGUI().setVisible(true));
    }
}
```

## Приложение 2 – содержимое файла «Integration.java»

```
package com.purpl.prog3;

public final class Integration {
    private Integration() {}

    public static class ResIntegral {
        double MIN = 0.000001;
        double MAX = 1000000.0;

        private double a;
        private double b;
        private double h;
        private Double result;

        public ResIntegral(double a, double b, double h) throws OutOfLimitException {
            if(a < MIN || b > MAX) {
                throw new OutOfLimitException("Неверные значения a и b - числа должно быть в
диапазоне: 0,000001-1000000");
            }

            this.a = a;
            this.b = b;
            this.h = h;
        }

        public double getA(){
            return this.a;
        }

        public double getB(){
            return this.b;
        }

        public double getH(){
            return this.h;
        }

        public Double getRes(){
            return this.result;
        }

        public void setA(double num){
            this.a = num;
        }

        public void setB(double num){
            this.b = num;
        }

        public void setH(double num){
            this.h = num;
        }

        public void setRes(double num){
            this.result = num;
        }
    }
}
```

```
public static double integrateSin(double a, double b, double h) {
    if (!Double.isFinite(a) || !Double.isFinite(b) || !Double.isFinite(h)) {
        throw new IllegalArgumentException("Параметры должны быть конечными числами.");
    }
    if (h <= 0) throw new IllegalArgumentException("Шаг должен быть > 0.");
    if (a == b) return 0.0;

    double sign = 1.0;
    if (a > b) {
        double tmp = a; a = b; b = tmp;
        sign = -1.0;
    }

    double sum = 0.0;
    double x = a;

    while (x < b) {
        double x2 = Math.min(x + h, b);
        sum += (Math.cos(x*x) + Math.cos(x2*x2)) * 0.5 * (x2 - x);
        x = x2;
    }
    return sign * sum;
}
}
```

### Приложение 3 – содержимое файла «JFrameGUI.java»

```
package com.purpl.prog3;

import java.util.ArrayList;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * 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
 */

/**
 *
 * @author izada
 */
public class JFrameGUI extends javax.swing.JFrame {

    private static final java.util.logging.Logger logger =
    java.util.logging.Logger.getLogger(JFrameGUI.class.getName());

    private final ArrayList<Integration.ResIntegral> entries = new ArrayList<>();
    private DefaultTableModel tableModel;
    private boolean isCleared = false;

    /**
     * Creates new form JFrameGUI
     */
    public JFrameGUI() {
        initComponents();

        tableModel = new DefaultTableModel(
            new Object[]{"Нижняя граница", "Верхняя граница", "Шаг", "Результат"}, 0) {
            @Override public boolean isCellEditable(int row, int col) {
                if (col == 3) {
                    return false;
                } else {
                    return true;
                }
            }
        );
        jTableVariables.setModel(tableModel);
    }

    /**
     * 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() {

    jPanelMain = new javax.swing.JPanel();
    jTextFieldUpperLimit = new javax.swing.JTextField();
    jTextFieldStep = new javax.swing.JTextField();
    jTextFieldLowerLimit = new javax.swing.JTextField();
    jLabelLowerLimit = new javax.swing.JLabel();
    jLabelUpperLimit = new javax.swing.JLabel();
    jLabelStep = new javax.swing.JLabel();
    jScrollPane2 = new javax.swing.JScrollPane();
    jTableVariables = new javax.swing.JTable();
    jButtonAdd = new javax.swing.JButton();
    jButtonDelete = new javax.swing.JButton();
    jButtonCalc = new javax.swing.JButton();
    jLabelMainHeader = new javax.swing.JLabel();
    jButtonClear = new javax.swing.JButton();
    jButtonFill = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabelLowerLimit.setText("Нижняя граница интегрирования:");

    jLabelUpperLimit.setText("Верхняя граница интегрирования:");

    jLabelStep.setText("Шаг интегрирования:");

    jTableVariables.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
        new String [] {
            "Нижняя граница", "Верхняя граница", "Шаг", "Результат"
        }
    ));
    jScrollPane2.setViewportView(jTableVariables);

    jButtonAdd.setBackground(new java.awt.Color(255, 51, 204));
    jButtonAdd.setText("Добавить");
    jButtonAdd.addActionListener(this::jButtonAddActionPerformed);

    jButtonDelete.setBackground(new java.awt.Color(255, 51, 204));
    jButtonDelete.setText("Удалить");
    jButtonDelete.addActionListener(this::jButtonDeleteActionPerformed);

    jButtonCalc.setBackground(new java.awt.Color(255, 51, 204));
    jButtonCalc.setText("Вычислить");
    jButtonCalc.addActionListener(this::jButtonCalcActionPerformed);

    jLabelMainHeader.setFont(new java.awt.Font("Segoe UI", 0, 24)); // NOI18N
    jLabelMainHeader.setText("Функция: cos(x^2)");

    jButtonClear.setBackground(new java.awt.Color(102, 255, 51));

```



```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanelMainLayout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jButtonClear, javax.swing.GroupLayout.PREFERRED_SIZE, 106,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(61, 61, 61)
        .addComponent(jButtonFill, javax.swing.GroupLayout.PREFERRED_SIZE, 106,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(154, 154, 154))
    );
    jPanelMainLayout.setVerticalGroup(
        jPanelMainLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanelMainLayout.createSequentialGroup()
        .addGap(37, 37, 37)
        .addComponent(jLabelMainHeader)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 12,
Short.MAX_VALUE)
        .addGroup(jPanelMainLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jButtonClear, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButtonFill, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

        .addGroup(jPanelMainLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanelMainLayout.createSequentialGroup()
                .addGroup(jPanelMainLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextFieldLowerLimit,
javax.swing.GroupLayout.PREFERRED_SIZE, 38, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabelLowerLimit))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                .addGroup(jPanelMainLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextFieldUpperLimit,
javax.swing.GroupLayout.PREFERRED_SIZE, 38, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabelUpperLimit))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                .addGroup(jPanelMainLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextFieldStep, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabelStep)))
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 302,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(37, 37, 37)

            .addGroup(jPanelMainLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButtonCalc, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButtonDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButtonAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE))
        )
    )
);

```

```

        .addGap(49, 49, 49))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanelMain, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanelMain, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(0, 0, 0)
        ).addGap(0, 0, 0)
    );
}

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

private void jButtonAddActionPerformed(java.awt.event.ActionEvent evt) {
    if(isCleared == true)
    {
        try
        {
            entries.clear();
        }
        catch (Exception ex)
        {
            JOptionPane.showMessageDialog(this,
                "Ошибка: " + ex.getMessage(),
                "Удаление данных", JOptionPane.WARNING_MESSAGE);
        }
        isCleared = false;
    }

    try {
        double a = parse(jTextFieldLowerLimit.getText(), "Нижняя граница");
        double b = parse(jTextFieldUpperLimit.getText(), "Верхняя граница");
        double h = parse(jTextFieldStep.getText(), "Шаг");

        if (h <= 0) throw new IllegalArgumentException("Шаг должен быть > 0.");

        Integration.ResIntegral e = new Integration.ResIntegral(a, b, h);
        entries.add(e);

        tableModel.addRow(new Object[]{a, b, h, ""});

        jTextFieldLowerLimit.setText("");
        jTextFieldUpperLimit.setText("");
        jTextFieldStep.setText("");

        int last = tableModel.getRowCount() - 1;
        jTableVariables.setRowSelectionInterval(last, last);
    }
}

```

```

        } catch (OutOfRangeException ex) {
            JOptionPane.showMessageDialog(this,
                ex.getMessage(),
                "Ошибка", JOptionPane.WARNING_MESSAGE);
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(this,
                "Ошибка: " + ex.getMessage(),
                "Ввод данных", JOptionPane.WARNING_MESSAGE);
        }
    }

private void jButtonDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    int row = jTableVariables.getSelectedRow();
    if (row < 0) {
        JOptionPane.showMessageDialog(this, "Выбери строку в таблице.",
            "Удаление", JOptionPane.INFORMATION_MESSAGE);
        return;
    }
    entries.remove(row);
    tableModel.removeRow(row);
}

private void jButtonCalcActionPerformed(java.awt.event.ActionEvent evt) {
    int row = jTableVariables.getSelectedRow();
    if (row < 0) {
        JOptionPane.showMessageDialog(this, "Выбери строку в таблице.",
            "Вычисление", JOptionPane.INFORMATION_MESSAGE);
        return;
    }

    try {
        double A = parse(tableModel.getValueAt(row, 0).toString(), "Нижняя граница");
        double B = parse(tableModel.getValueAt(row, 1).toString(), "Верхняя граница");
        double H = parse(tableModel.getValueAt(row, 2).toString(), "Шаг");

        Integration.ResIntegral eE = new Integration.ResIntegral(A, B, H);

        entries.set(row, eE);

        Integration.ResIntegral e = entries.get(row);
        double res = Integration.integrateSin(e.getA(), e.getB(), e.getH());
        e.setRes(res);

        tableModel.setValueAt(res, row, 3);
    } catch (OutOfRangeException ex) {
        JOptionPane.showMessageDialog(this,
            ex.getMessage(),
            "Ошибка", JOptionPane.WARNING_MESSAGE);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(this,
            "Ошибка вычисления: " + ex.getMessage(),
            "Ошибка", JOptionPane.WARNING_MESSAGE);
    }
}

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

```

```

        for (int i = tableViewModel.getRowCount() - 1; i >= 0; i--)
        {
            tableViewModel.removeRow(i);
        }
        jTableVariables.clearSelection();
        isCleared = true;
    }

private void jButtonFillActionPerformed(java.awt.event.ActionEvent evt) {
    tableViewModel.setRowCount(0);

    for (Integration.ResIntegral e : entries) {
        Object res = (e.getRes() == null) ? "" : e.getRes();
        tableViewModel.addRow(new Object[]{ e.getA(), e.getB(), e.getH(), res });
    }
    isCleared = false;
}

private double parse(String s, String fieldName) {
    s = s.trim();
    if (s.isEmpty()) throw new IllegalArgumentException(fieldName + " не задано.");
    s = s.replace(',', '.');
    return Double.parseDouble(s);
}

// Variables declaration - do not modify
private javax.swing.JButton jButtonAdd;
private javax.swing.JButton jButtonCalc;
private javax.swing.JButton jButtonClear;
private javax.swing.JButton jButtonDelete;
private javax.swing.JButton jButtonFill;
private javax.swing.JLabel jLabelLowerLimit;
private javax.swing.JLabel jLabelMainHeader;
private javax.swing.JLabel jLabelStep;
private javax.swing.JLabel jLabelUpperLimit;
private javax.swing.JPanel jPanelMain;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTableVariables;
private javax.swing.JTextField jTextFieldLowerLimit;
private javax.swing.JTextField jTextFieldStep;
private javax.swing.JTextField jTextFieldUpperLimit;
// End of variables declaration
}

```

## **Приложение 4 – содержимое файла «OutOfLimitException.java»**

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.purpl.prog3;

/**
 *
 * @author izada
 */
public class OutOfLimitException extends Exception {
    public OutOfLimitException(String message){
        super(message);
    }
}
```