Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №2

по дисциплине «Программирование на языке Java»

на тему «Работа с коллекциями объектов»

Выполнили:

студенты группы 22ВВП1

Горбатов К.

Коробкин В.

Приняли:

Юрова О.В.

Карамышева Н.С.

Пенза 2025

**Название**

Работа с коллекциями объектов

**Цель работы**

Изучить библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.

**Лабораторное задание**

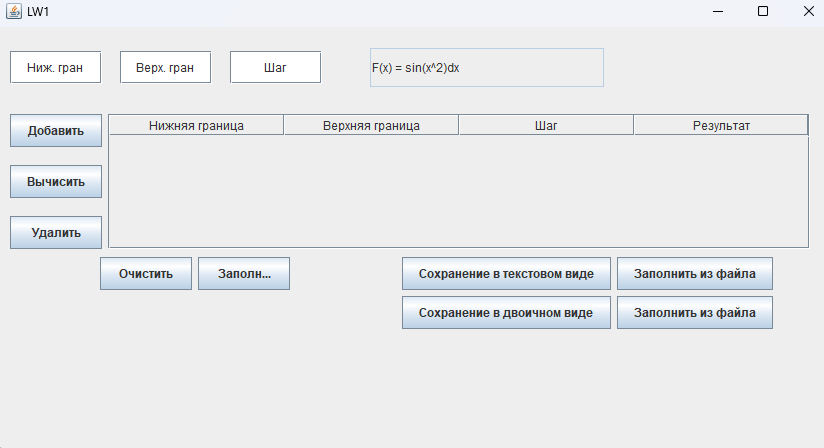
Модифицировать приложение из предыдущей лабораторной работы, реализовав хранение данных таблицы с использованием библиотеки коллекций. Для этого реализовать класс RecIntegral, способный хранить одну запись таблицы. Для нечетных вариантов в качестве класса-коллекции выбрать ArrayList, для четных - LinkedList. Кроме того, добавить пару кнопок: очистить / заполнить, которые будут очищать таблицу и заполнять ее данными из коллекции соответственно. Оформление лабораторной работы должно быть выполнено в соответствии с требованиями, приведенными в Приложении 2.

Вариант 8:



**Ход работы**

Добавили дополнительные кнопки в интерфейс

****

**Создали класс “RecIntegral”**

package lab1;

import java.io.Serializable;

public class RecIntegral implements Serializable {

public double Top, Bottom, Step, Res;

public RecIntegral(double bottom, double top, double step) {

this.Top = top;

this.Bottom = bottom;

this.Step = step;

}

public RecIntegral() {}

public double getBottom() {

return Bottom;

}

public void setBottom(double Bottom) {

this.Bottom = Bottom;

}

public double getTop() {

return Top;

}

public void setTop(double Top) {

this.Top = Top;

}

public double getStep() {

return Step;

}

public void setStep(double Step) {

this.Step = Step;

}

public double getRes() {

return Res;

}

public void setRes(double Res) {

this.Res = Res;

}

@Override

public String toString() {

return "RecIntegral{" +

"bottom=" + Bottom +

", Top=" + Top +

", Step=" + Step +

", Res=" + Res +

'}';

}

}

**Добавили функционал в кнопку “Очистить”**

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// Очистить

DefaultTableModel tbl = (DefaultTableModel) jTable1.getModel();

for (int i = tbl.getRowCount(); i > 0; i--) {

tbl.removeRow(i - 1);

}

}

**Добавили функционал в кнопку “Заполнить”**

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// Заполнить

DefaultTableModel tbl = (DefaultTableModel) jTable1.getModel();

Object rowData[] = new Object[4];

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

rowData[0] = digits.get(i).Bottom;

rowData[1] = digits.get(i).Top;

rowData[2] = digits.get(i).Step;

if(digits.get(i).Res == 0) {

rowData[3] = null;

}

else {

rowData[3] = digits.get(i).Res;

}

tbl.addRow(rowData);

}

}

**Листинг программы**

**LW\_UI.java**

package lab1;

import javax.swing.table.DefaultTableModel;

import java.util.\*;

import javax.swing.\*;

import java.io.\*;

import javax.swing.filechooser.FileNameExtensionFilter;

public class LW\_UI extends javax.swing.JFrame {

/\*\*

\* Creates new form LW\_UI

\*/

public LW\_UI() {

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();

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();

jTextField4 = new javax.swing.JTextField();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton6 = new javax.swing.JButton();

jButton7 = new javax.swing.JButton();

jButton8 = new javax.swing.JButton();

jButton9 = new javax.swing.JButton();

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

setTitle("LW1");

jTextField1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

jTextField1.setText("Ниж. гран");

jTextField2.setHorizontalAlignment(javax.swing.JTextField.CENTER);

jTextField2.setText("Верх. гран");

jTextField2.setToolTipText("");

jTextField3.setHorizontalAlignment(javax.swing.JTextField.CENTER);

jTextField3.setText("Шаг");

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

new Object [][] {

},

new String [] {

"Нижняя граница", "Верхняя граница", "Шаг", "Результат"

}

) {

Class[] types = new Class [] {

java.lang.Object.class, java.lang.Object.class, java.lang.Object.class, java.lang.Double.class

};

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(jTable1);

if (jTable1.getColumnModel().getColumnCount() > 0) {

jTable1.getColumnModel().getColumn(2).setResizable(false);

jTable1.getColumnModel().getColumn(3).setResizable(false);

}

jButton1.setText("Вычисить");

jButton1.addActionListener(new java.awt.event.ActionListener() {

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Удалить");

jButton2.addActionListener(new java.awt.event.ActionListener() {

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

jButton2ActionPerformed(evt);

}

});

jButton3.setText("Добавить");

jButton3.addActionListener(new java.awt.event.ActionListener() {

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

jButton3ActionPerformed(evt);

}

});

jTextField4.setEditable(false);

jTextField4.setText("F(x) = sin(x^2)dx");

jButton4.setText("Заполнить");

jButton4.addActionListener(new java.awt.event.ActionListener() {

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

jButton4ActionPerformed(evt);

}

});

jButton5.setText("Очистить");

jButton5.addActionListener(new java.awt.event.ActionListener() {

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

jButton5ActionPerformed(evt);

}

});

jButton6.setText("Сохранение\n в текстовом \nвиде");

jButton6.addActionListener(new java.awt.event.ActionListener() {

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

jButton6ActionPerformed(evt);

}

});

jButton7.setText("Заполнить из файла");

jButton7.addActionListener(new java.awt.event.ActionListener() {

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

jButton7ActionPerformed(evt);

}

});

jButton8.setText("Сохранение в двоичном виде");

jButton8.addActionListener(new java.awt.event.ActionListener() {

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

jButton8ActionPerformed(evt);

}

});

jButton9.setText("Заполнить из файла");

jButton9.addActionListener(new java.awt.event.ActionListener() {

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

jButton9ActionPerformed(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)

.addGroup(layout.createSequentialGroup()

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(48, 48, 48)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, 234, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addGroup(layout.createSequentialGroup()

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

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton3, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

.addComponent(jScrollPane1)

.addGap(22, 22, 22))))

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

.addGap(102, 102, 102)

.addComponent(jButton5, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jButton4, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 112, Short.MAX\_VALUE)

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

.addGroup(layout.createSequentialGroup()

.addComponent(jButton8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

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

.addComponent(jButton9))

.addGroup(layout.createSequentialGroup()

.addComponent(jButton6)

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

.addComponent(jButton7)))

.addGap(59, 59, 59))

);

layout.setVerticalGroup(

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

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

.addGap(21, 21, 21)

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

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, 39, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 27, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(jButton3, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE))

.addGap(8, 8, 8)

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

.addComponent(jButton5, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton4, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton7, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton6, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

.addComponent(jButton8, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton9, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(120, 120, 120))

);

pack();

}// </editor-fold>

class Chooser {

String FileName;

public String SaveOpen(String NameFilter) {

JFileChooser chooser = new JFileChooser();

FileNameExtensionFilter txt = new FileNameExtensionFilter(

"Text File(txt)", "txt");

FileNameExtensionFilter bin = new FileNameExtensionFilter(

"Binary File(bin)", "bin");

chooser.addChoosableFileFilter(txt);

chooser.addChoosableFileFilter(bin);

int returnVal = chooser.showDialog(null, null);

if (returnVal == JFileChooser.APPROVE\_OPTION) {

FileName = chooser.getSelectedFile().getAbsolutePath();

}

return FileName;

}

}

ArrayList<RecIntegral> digits = new ArrayList<RecIntegral>();

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

// TODO add your handling code here:

//Кнопка добаить

try{

RecIntegral data = InputValidator.validateAndParse(

jTextField1.getText(),

jTextField2.getText(),

jTextField3.getText()

);

((DefaultTableModel) jTable1.getModel()).addRow(

new Object[]{data.getBottom(), data.getTop(), data.getStep()}

);

digits.add(data);

}

catch(ExceptionforRecIntegral ex){

javax.swing.JOptionPane.showMessageDialog(this,

ex.getMessage(),

"Ошибка",

javax.swing.JOptionPane.ERROR\_MESSAGE);

}

}

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

// TODO add your handling code here:

//Кнопка удалить

DefaultTableModel tbl = (DefaultTableModel) jTable1.getModel();

int indexROW = jTable1.getSelectedRow();

tbl.removeRow(indexROW);

digits.remove(indexROW);

}

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

//Кнопка вычислить

if (jTable1.getRowCount() == 0) return;

int selectRow = jTable1.getSelectedRow();

if (selectRow == -1) return;

try {

RecIntegral data = InputValidator.validateAndParse(

jTable1.getValueAt(selectRow, 0).toString(),

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

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

);

CalculateIntegral integral = new CalculateIntegral(data);

double res = integral.calculate();

jTable1.getColumnModel().getColumn(3).setCellRenderer(new DoubleRenderer());

jTable1.setValueAt(res, selectRow, 3);

data.setRes(res);

digits.set(selectRow, data);

} catch(ExceptionforRecIntegral ex){

javax.swing.JOptionPane.showMessageDialog(this,

ex.getMessage(),

"Ошибка",

javax.swing.JOptionPane.ERROR\_MESSAGE);

}

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// Очистить

DefaultTableModel tbl = (DefaultTableModel) jTable1.getModel();

for (int i = tbl.getRowCount(); i > 0; i--) {

tbl.removeRow(i - 1);

}

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// Заполнить

DefaultTableModel tbl = (DefaultTableModel) jTable1.getModel();

Object rowData[] = new Object[4];

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

rowData[0] = digits.get(i).Bottom;

rowData[1] = digits.get(i).Top;

rowData[2] = digits.get(i).Step;

if(digits.get(i).Res == 0) {

rowData[3] = null;

}

else {

rowData[3] = digits.get(i).Res;

}

tbl.addRow(rowData);

}

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

//Сериализация в текстовом виде

ArrayList<String> ser = new ArrayList<String>();

Chooser chsr = new Chooser();

String Filename = chsr.SaveOpen("txt") + ".txt";

try {

FileOutputStream fout = new FileOutputStream(Filename);

BufferedOutputStream bufferFout = new BufferedOutputStream(fout);

ObjectOutputStream ObjFout = new ObjectOutputStream(bufferFout);

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

Object step = null, bottom = null, top = null;

RecIntegral node = digits.get(i);

top = node.Top;

bottom = node.Bottom;

step = node.Step;

ser.add(bottom.toString() + ' ' + top.toString() + ' ' + step.toString());

}

ObjFout.writeObject(ser);

ObjFout.close();

} catch (IOException ex) {

ex.printStackTrace();

}

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

//заполнить из txt

DefaultTableModel tbl = (DefaultTableModel) jTable1.getModel();

Chooser chsr = new Chooser();

String Filename = chsr.SaveOpen("txt");

ArrayList<String> ser = new ArrayList<String>();

while (tbl.getRowCount() != 0) {

tbl.removeRow(tbl.getRowCount() - 1);

}

while (digits.size() != 0) {

digits.remove(digits.size() - 1);

}

try {

FileInputStream fin = new FileInputStream(Filename);

BufferedInputStream bufferFin = new BufferedInputStream(fin);

ObjectInputStream ObjFin = new ObjectInputStream(bufferFin);

ser = (ArrayList<String>) ObjFin.readObject();

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

String str = ser.get(i),

strTop = " ",

strBottom = " ",

strStep = " ";

int size = str.length();

int j = 0;

while(str.charAt(j) != ' '){

strBottom += str.charAt(j);

j++;

}

j++;

while(str.charAt(j) != ' '){

strTop += str.charAt(j);

j++;

}

j++;

while(j != size){

strStep += str.charAt(j);

j++;

}

tbl.addRow(new Object[]{Double.parseDouble(strBottom), Double.parseDouble(strTop), Double.parseDouble(strStep), null});

RecIntegral node = new RecIntegral(Double.parseDouble(strBottom), Double.parseDouble(strTop), Double.parseDouble(strStep));

digits.add(node);

}

} catch (IOException ex) {

ex.printStackTrace();

} catch (ClassNotFoundException ex) {

ex.printStackTrace();

}

}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

//сохранить в bin

ArrayList<String> ser = new ArrayList<String>();

Chooser chsr = new Chooser();

String Filename = chsr.SaveOpen("bin") + ".bin";

try {

FileOutputStream fout = new FileOutputStream(Filename);

BufferedOutputStream bufferFout = new BufferedOutputStream(fout);

ObjectOutputStream ObjFout = new ObjectOutputStream(bufferFout);

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

Object step = null, bottom = null, top = null;

RecIntegral node = digits.get(i);

top = node.Top;

bottom = node.Bottom;

step = node.Step;

ser.add(bottom.toString() + ' ' + top.toString() + ' ' + step.toString());

}

ObjFout.writeObject(ser);

ObjFout.close();

} catch (IOException ex) {

ex.printStackTrace();

}

}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

//открыть бин

DefaultTableModel tbl = (DefaultTableModel) jTable1.getModel();

Chooser chsr = new Chooser();

String Filename = chsr.SaveOpen("bin");

ArrayList<String> ser = new ArrayList<String>();

while (tbl.getRowCount() != 0) {

tbl.removeRow(tbl.getRowCount() - 1);

}

while (digits.size() != 0) {

digits.remove(digits.size() - 1);

}

try {

FileInputStream fin = new FileInputStream(Filename);

BufferedInputStream bufferFin = new BufferedInputStream(fin);

ObjectInputStream ObjFin = new ObjectInputStream(bufferFin);

ser = (ArrayList<String>) ObjFin.readObject();

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

String str = ser.get(i),

strTop = " ",

strBottom = " ",

strStep = " ";

int size = str.length();

int j = 0;

while(str.charAt(j) != ' '){

strBottom += str.charAt(j);

j++;

}

j++;

while(str.charAt(j) != ' '){

strTop += str.charAt(j);

j++;

}

j++;

while(j != size){

strStep += str.charAt(j);

j++;

}

tbl.addRow(new Object[]{Double.parseDouble(strBottom), Double.parseDouble(strTop), Double.parseDouble(strStep), null});

RecIntegral node = new RecIntegral(Double.parseDouble(strBottom), Double.parseDouble(strTop), Double.parseDouble(strStep));

digits.add(node);

}

} catch (IOException ex) {

ex.printStackTrace();

} catch (ClassNotFoundException ex) {

ex.printStackTrace();

}

}

/\*\*

\* @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(LW\_UI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(LW\_UI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(LW\_UI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LW\_UI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new LW\_UI().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.JButton jButton6;

private javax.swing.JButton jButton7;

private javax.swing.JButton jButton8;

private javax.swing.JButton jButton9;

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;

private javax.swing.JTextField jTextField4;

// End of variables declaration

}

**CalculateIntegral.java**

package lab1;

import static java.lang.Math.pow;

import static java.lang.Math.sin;

public class CalculateIntegral {

private double Bottom;

private double Top;

private double step;

private double result;

public CalculateIntegral(double Bottom, double Top, double step) {

this.Bottom = Bottom;

this.Top = Top;

this.step = step;

}

public CalculateIntegral(RecIntegral data) {

this.Bottom = data.Bottom;

this.Top = data.Top;

this.step = data.Step;

}

public double calculate() {

boolean isReversed = Bottom > Top;

if (isReversed) {

double tempBorder = Bottom;

Bottom = Top;

Top = tempBorder;

}

double currentBottom = Bottom;

long count = (long)((Top - Bottom) / step);

double sum = 0;

for (long j = 0; j < count; j++) {

sum += ((step / 2) \* (sin(pow(currentBottom,2)) + sin(pow((currentBottom + step), 2))));

currentBottom += step;

}

if((Top - Bottom) / step > count) {

currentBottom -= step;

double lastStepWeigth = Top - (currentBottom);

sum += ((lastStepWeigth / 2) \* (sin(pow(currentBottom, 2)) + sin(pow(Top, 2))));

}

return isReversed ? -sum : sum;

}

}

**DoubleRenderer.java**

package lab1;

import java.text.DecimalFormat;

import javax.swing.table.DefaultTableCellRenderer;

class DoubleRenderer extends DefaultTableCellRenderer {

private DecimalFormat formatter = new DecimalFormat("#.##########");

@Override

protected void setValue(Object value) {

if (value instanceof Double) {

setText(formatter.format(value));

} else {

super.setValue(value);

}

}

}

**ExceptionforRecIntegral.java**

package lab1;

public class ExceptionforRecIntegral extends Exception {

public ExceptionforRecIntegral(String description) {

super(description);

}

}

**InputValidator.java**

package lab1;

public class InputValidator {

public static RecIntegral validateAndParse(String lower, String upper, String step) throws ExceptionforRecIntegral {

double lowerVal = parseValue(lower);

double upperVal = parseValue(upper);

double stepVal = parseValue(step);

validateRange(lowerVal, upperVal, stepVal);

return new RecIntegral(lowerVal, upperVal, stepVal);

}

private static double parseValue(String input) throws ExceptionforRecIntegral {

try {

double value = Double.parseDouble(input);

if (value == 0 || value < 0.000001 || value > 1000000) {

throw new ExceptionforRecIntegral("Значение должно быть от 0.000001 до 1000000");

}

return value;

} catch (NumberFormatException e) {

throw new ExceptionforRecIntegral("Некорректный числовой формат");

}

}

private static void validateRange(double lower, double upper, double step) throws ExceptionforRecIntegral {

if (step <= 0) {

throw new ExceptionforRecIntegral("Шаг должен быть положительным");

}

if (Math.abs(upper - lower) < step) {

throw new ExceptionforRecIntegral("Интервал должен быть не меньше шага");

}

}

}

**Lab1.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package lab1;

public class Lab1 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

LW\_UI frame = new LW\_UI();

frame.show();

}

}

**RecIntegral.java**

package lab1;

import java.io.Serializable;

public class RecIntegral implements Serializable {

public double Top, Bottom, Step, Res;

public RecIntegral(double bottom, double top, double step) {

this.Top = top;

this.Bottom = bottom;

this.Step = step;

}

public RecIntegral() {}

public double getBottom() {

return Bottom;

}

public void setBottom(double Bottom) {

this.Bottom = Bottom;

}

public double getTop() {

return Top;

}

public void setTop(double Top) {

this.Top = Top;

}

public double getStep() {

return Step;

}

public void setStep(double Step) {

this.Step = Step;

}

public double getRes() {

return Res;

}

public void setRes(double Res) {

this.Res = Res;

}

@Override

public String toString() {

return "RecIntegral{" +

"bottom=" + Bottom +

", Top=" + Top +

", Step=" + Step +

", Res=" + Res +

'}';

}

}

**Вывод:** В ходе выполнения лабораторной работы изучили библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.