Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по практической работе №4

по курсу «Разработка кроссплатформенных приложений»

## на тему «Графические интерфейсы»

**Выполнили:**

студенты группы 20ВВ1.1

Вяльмисов М.А.

Репин И.В.

**Приняли:**

Юрова О.В.

Пенза 2023

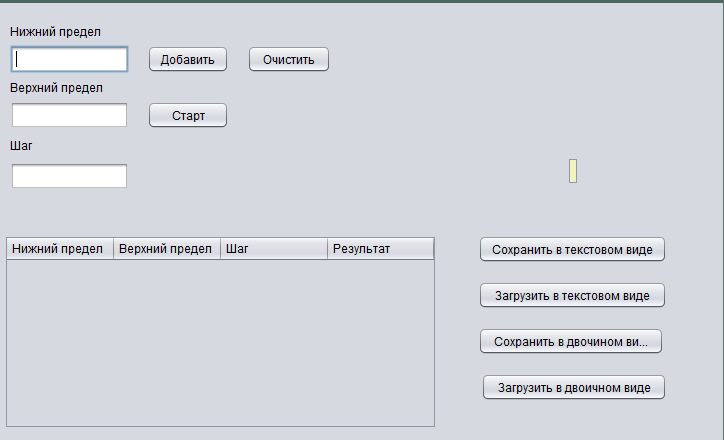
**Цель работы:** изучить работу с файлами и механизмы сериализации данных.

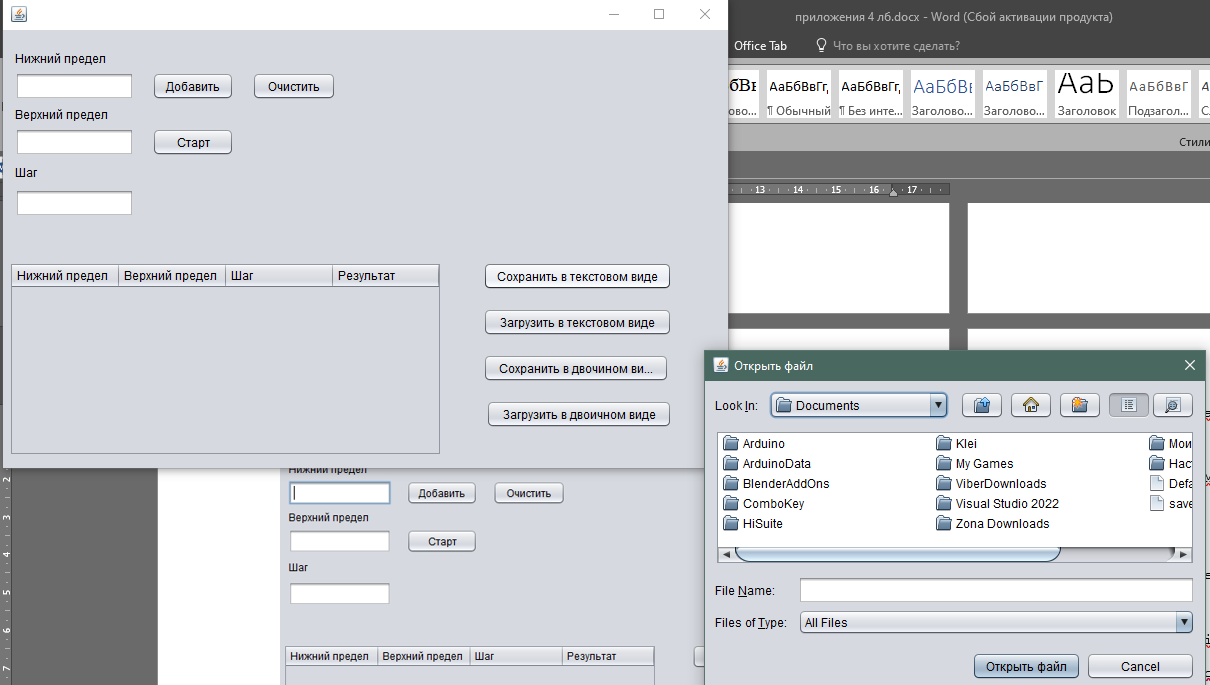
**Вариант №1**

**Задание:**

Модифицировать приложение из предыдущей лабораторной работы, реализовав сохранение в файл и загрузку данных из файла. Предусмотреть сохранение данных, как в текстовом виде, так и в двоичном (с использованием механизма сериализации). Для этого нужно добавить 4 кнопки для сохранения и загрузки в текстовом и двоичном виде соответственно. Кроме того, в программе нужно предусмотреть использование стандартного диалога открытия файла (JFileChooser). Оформление лабораторной работы должно быть выполнено в соответствии с требованиями, приведенными в Приложении 2.

**Скриншоты программы:**





**Листинг программы:**

**Class RecIntegral**

/\*

\* 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 my.numberaddition;

import java.io.Serializable;

import my.numberaddition.Result;

/\*\*

\*

\* @author maksv

\*/

public class RecIntegral extends Thread implements Serializable {

private float bottomBorder;

private float upperBorder;

private float step;

private Result<Float> result = new Result();

public RecIntegral(float bottomBorder, float upperBorder, float step) throws InException {

if(bottomBorder < 0.000001 || bottomBorder > 1000000 || upperBorder < 0.000001 || upperBorder > 1000000 || bottomBorder > upperBorder){

throw new InException();

}

this.bottomBorder = bottomBorder;

this.upperBorder = upperBorder;

this.step = step;

}

public RecIntegral(String str){

String[] strs = str.split(",");

this.bottomBorder = Float.parseFloat(strs[0]);

this.upperBorder = Float.parseFloat(strs[1]);

this.step = Float.parseFloat(strs[2]);

try{

this.result.obj = Float.parseFloat(strs[3]);

}catch(Exception e){

this.result.obj = null;

}

}

public void run() {

CalculateResult();

}

public void CalculateResult() {

if ((this.bottomBorder > this.upperBorder) || (this.bottomBorder == 0) || (this.upperBorder == 0) || (this.step == 0)) {

return;

}

if ((this.bottomBorder < 0) && (this.upperBorder > 0)) {

this.result.str = "Infinite";

} else {

float next\_step = this.bottomBorder;

float prev\_step = this.bottomBorder;

float result = 0;

do {

next\_step = next\_step + this.step;

if (next\_step > this.upperBorder) {

next\_step = this.upperBorder;

}

float iteration\_result = 1 / next\_step + 1 / prev\_step;

iteration\_result = iteration\_result \* (next\_step - prev\_step) / 2;

result += iteration\_result;

prev\_step = next\_step;

} while (next\_step != this.upperBorder);

this.result.obj = result;

}

}

public float GetBottomBorder() {

return this.bottomBorder;

}

public float GetUpperBorder() {

return this.upperBorder;

}

public float GetStep() {

return this.step;

}

public Result<Float> GetResult() {

return this.result;

}

@Override

public String toString(){

return String.valueOf(this.bottomBorder) +"," + String.valueOf(this.upperBorder)+ "," + String.valueOf(this.step) + "," + String.valueOf(this.result.obj) + ";";

}

}

**Main**

/\*

\* 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.numberaddition;

import javax.swing.table.DefaultTableModel;

import java.util.ArrayList;

import java.io.\*;

import javax.swing.\*;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.StandardOpenOption;

/\*\*

\*

\* @author ivanr

\*/

public class MainWindow extends javax.swing.JFrame {

/\*\*

\* Creates new form NewJFrame

\*/

public MainWindow() {

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() {

jLabel212e12e = new javax.swing.JLabel();

jPanel1 = new javax.swing.JPanel();

bottomBorderField = new javax.swing.JTextField();

upperBorderField = new javax.swing.JTextField();

stepField = new javax.swing.JTextField();

calculateButton = new javax.swing.JButton();

resetButton = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

addButton = new javax.swing.JButton();

saveInTextTypeButton = new javax.swing.JButton();

loadInTextTypeButton = new javax.swing.JButton();

saveInBinaryButton = new javax.swing.JButton();

loadInBinaryButton = new javax.swing.JButton();

jLabel212e12e.setText("jLabel2");

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

jPanel1.setToolTipText("");

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

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

bottomBorderFieldActionPerformed(evt);

}

});

calculateButton.setText("Старт");

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

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

calculateButtonActionPerformed(evt);

}

});

resetButton.setText("Очистить");

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

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

resetButtonActionPerformed(evt);

}

});

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

new Object [][] {

},

new String [] {

"Нижний предел", "Верхний предел", "Шаг", "Результат"

}

) {

boolean[] canEdit = new boolean [] {

false, false, false, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jTable1.setToolTipText("");

jTable1.getTableHeader().setReorderingAllowed(false);

jScrollPane1.setViewportView(jTable1);

jLabel1.setText("Верхний предел");

jLabel2.setText("Шаг");

jLabel3.setText("Нижний предел");

addButton.setText("Добавить");

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

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

addButtonActionPerformed(evt);

}

});

saveInTextTypeButton.setText("Сохранить в текстовом виде");

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

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

saveInTextTypeButtonActionPerformed(evt);

}

});

loadInTextTypeButton.setText("Загрузить в текстовом виде");

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

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

loadInTextTypeButtonActionPerformed(evt);

}

});

saveInBinaryButton.setText("Сохранить в двочином виде");

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

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

saveInBinaryButtonActionPerformed(evt);

}

});

loadInBinaryButton.setText("Загрузить в двоичном виде");

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

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

loadInBinaryButtonActionPerformed(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.TRAILING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

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

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 111, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 111, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel1Layout.createSequentialGroup()

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

.addComponent(stepField, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(upperBorderField, javax.swing.GroupLayout.DEFAULT\_SIZE, 119, Short.MAX\_VALUE)

.addComponent(bottomBorderField))

.addGap(18, 18, 18)

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

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

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

.addGap(18, 18, 18)

.addComponent(resetButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 84, javax.swing.GroupLayout.PREFERRED\_SIZE)))

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

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 433, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(41, 41, 41)

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

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

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

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

.addComponent(saveInBinaryButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 186, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(loadInBinaryButton, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 186, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(0, 50, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

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

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(14, 14, 14)

.addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

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

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

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

.addComponent(addButton)

.addComponent(resetButton))

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

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

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

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

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

.addComponent(calculateButton))

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

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(7, 7, 7)

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

.addGap(45, 45, 45)

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

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(saveInTextTypeButton)

.addGap(18, 18, 18)

.addComponent(loadInTextTypeButton)

.addGap(18, 18, 18)

.addComponent(saveInBinaryButton)

.addGap(18, 18, 18)

.addComponent(loadInBinaryButton))

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 194, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap())

);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

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

.addContainerGap()

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

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

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

);

pack();

}// </editor-fold>

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

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

int rowCount = myTable.getRowCount();

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

myTable.removeRow(0);

}

recIntegrals.removeAll(recIntegrals);

}

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

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

int row\_count = myTable.getRowCount();

for (int i = 0; i < row\_count; i++) {

RecIntegral recIntegral = recIntegrals.get(i);

try {

recIntegral.run();

recIntegral.join();

} catch (Exception ex) {

ex.printStackTrace();

}

if (recIntegral.GetResult().obj != null) {

myTable.setValueAt(recIntegral.GetResult().obj, i, 3);

} else {

myTable.setValueAt(recIntegral.GetResult().str, i, 3);

}

}

}

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

// TODO add your handling code here

}

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

RecIntegral recIntegral;

try {

float bottomBorder = Float.parseFloat(bottomBorderField.getText());

float upperBorder = Float.parseFloat(upperBorderField.getText());

float step = Float.parseFloat(stepField.getText());

recIntegral = new RecIntegral(bottomBorder, upperBorder, step);

} catch (InException e) {

JFrame jFrame = new JFrame();

JOptionPane.showMessageDialog(jFrame, "Некоректные данные! Введите другие!");

return;

}

recIntegrals.add(recIntegral);

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

myTable.addRow(new Object[]{recIntegral.GetBottomBorder(), recIntegral.GetUpperBorder(), recIntegral.GetStep()});

}

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

JFileChooser fileopen = new JFileChooser();

int ret = fileopen.showDialog(null, "Открыть файл");

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

File file = fileopen.getSelectedFile();

try {

FileWriter myfile = new FileWriter(file.getAbsolutePath());

String str = new String();

for (RecIntegral ri : recIntegrals) {

str += ri.toString();

}

System.out.print(str);

myfile.write(str);

myfile.flush();

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

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

JFileChooser fileopen = new JFileChooser();

int ret = fileopen.showDialog(null, "Открыть файл");

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

File file = fileopen.getSelectedFile();

try {

FileReader myfile = new FileReader(file);

char recIntegralsBuf[] = new char[1000];

myfile.read(recIntegralsBuf);

System.out.println(recIntegralsBuf);

String recIntegralsStr = new String(recIntegralsBuf);

String[] recIntegralsStrList = recIntegralsStr.split(";");

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

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

if (i == recIntegralsStrList.length - 1) {

break;

}

String riStr = recIntegralsStrList[i];

if (!riStr.isEmpty() && !riStr.isBlank()) {

RecIntegral ri = new RecIntegral(riStr);

recIntegralsList.add(ri);

}

}

recIntegrals = recIntegralsList;

updateTable();

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

private void updateTable() {

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

for (RecIntegral recIntegral : recIntegrals) {

float bottomBorder = recIntegral.GetBottomBorder();

float upperBorder = recIntegral.GetUpperBorder();

float step = recIntegral.GetStep();

if (recIntegral.GetResult().obj != null) {

myTable.addRow(new Object[]{bottomBorder, upperBorder, step, recIntegral.GetResult().obj});

} else {

myTable.addRow(new Object[]{bottomBorder, upperBorder, step, recIntegral.GetResult().str});

}

}

}

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

JFileChooser fileopen = new JFileChooser();

int ret = fileopen.showDialog(null, "Открыть файл");

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

File file = fileopen.getSelectedFile();

try {

FileOutputStream outputStream = new FileOutputStream(file, false);

ObjectOutputStream out = new ObjectOutputStream(outputStream);

out.writeObject(recIntegrals);

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

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

JFileChooser fileopen = new JFileChooser();

int ret = fileopen.showDialog(null, "Открыть файл");

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

File file = fileopen.getSelectedFile();

try {

ObjectInputStream in = new ObjectInputStream(new FileInputStream(file));

recIntegrals.removeAll(recIntegrals);

recIntegrals = (ArrayList<RecIntegral>) in.readObject();

updateTable();

} catch (Exception 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(MainWindow.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new MainWindow().setVisible(true);

}

});

}

private ArrayList<RecIntegral> recIntegrals = new ArrayList<RecIntegral>();

// Variables declaration - do not modify

private javax.swing.JButton addButton;

private javax.swing.JTextField bottomBorderField;

private javax.swing.JButton calculateButton;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel212e12e;

private javax.swing.JLabel jLabel3;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JButton loadInBinaryButton;

private javax.swing.JButton loadInTextTypeButton;

private javax.swing.JButton resetButton;

private javax.swing.JButton saveInBinaryButton;

private javax.swing.JButton saveInTextTypeButton;

private javax.swing.JTextField stepField;

private javax.swing.JTextField upperBorderField;

// End of variables declaration

}

**Вывод:** изучили работу с файлами и механизмы сериализации данных.