Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №4

по курсу «Программирование на языке Java»

на тему «Обработка исключительных ситуаций»

Вариант №9

Выполнили

студенты группы 20ВВП2:

Халилов А.

Кузин Д.

Приняли:

Юрова О. В.

Карамышева Н. С.

Пенза 2023

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

**Лабораторное 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.

**Результат выполнения программы:**

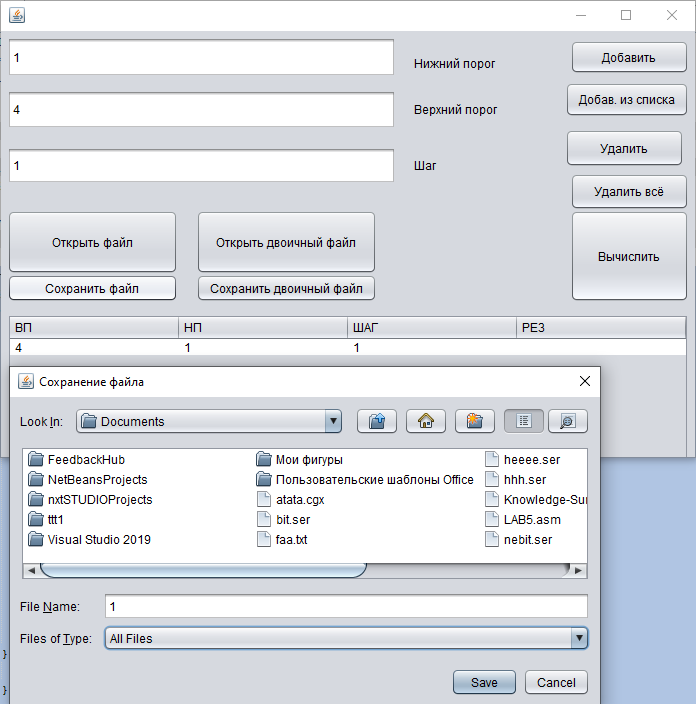


Рисунок 1 — Сохранение в файл .txt.

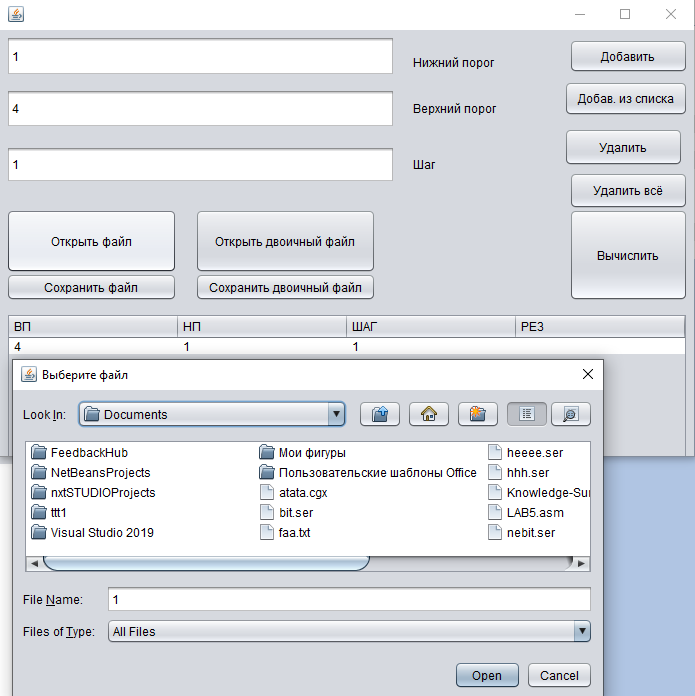


Рисунок 2 — Открытие файла.

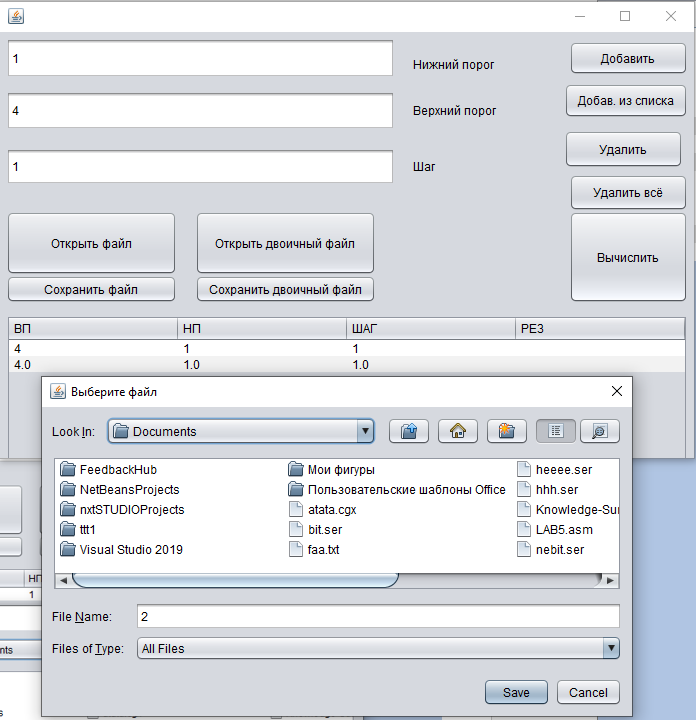


Рисунок 3 — Сохранение в двоичный файл.

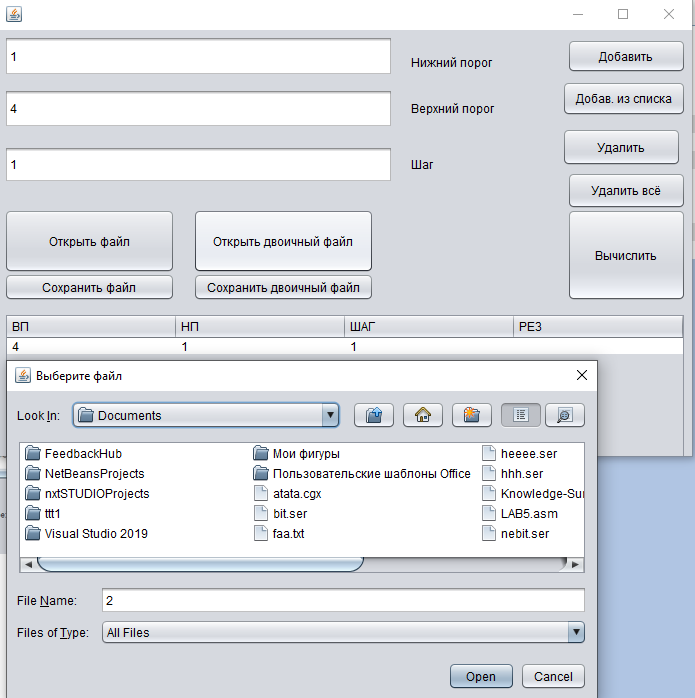


Рисунок 4 — Открытие двоичного файла.

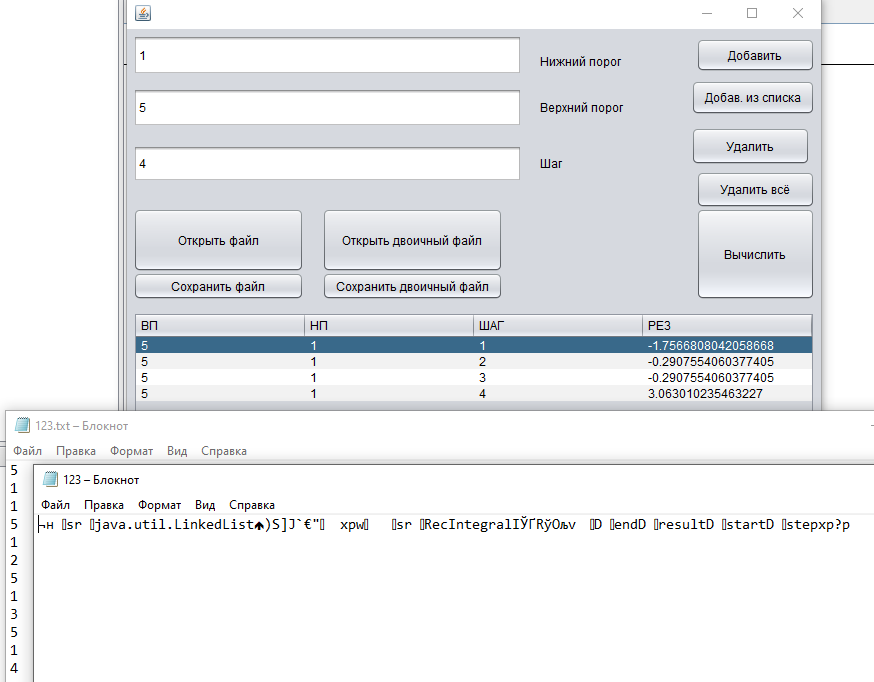


Рисунок 5 —Результат сохранения.

**Вывод:** в ходе выполнения данной лабораторной работы были изучены способы работы с файлами и механизмы сериализации данных.

**Листинг:**

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

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

jFileChooser1.setDialogTitle("Сохранение файла");

if(jFileChooser1.showSaveDialog(null) == jFileChooser1.APPROVE\_OPTION){

File file = new File(jFileChooser1.getSelectedFile() + ".txt");

try{

if(!file.exists()){

file.createNewFile();

}

PrintWriter out = new PrintWriter(file);

try {

for(int i = 0; i < model.getRowCount(); i++){

for(int j = 0; j < 3; j++){

out.println(model.getValueAt(i, j));

}

}

}finally {

out.close();

}

}catch (IOException ex){

throw new RuntimeException(ex);

}

}

}

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

try{

int ret = jFileChooser1.showSaveDialog(null);

File file = null;

if(ret == jFileChooser1.APPROVE\_OPTION){

file = jFileChooser1.getSelectedFile();

} else{

file = new File(jFileChooser1.getSelectedFile() + ".ser");

}

FileOutputStream fileStream = new FileOutputStream(file);

ObjectOutputStream os = new ObjectOutputStream(fileStream);

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

os.writeObject(list);

}

}catch (Exception ex){

ex.printStackTrace();

}

}

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

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

jFileChooser1.setDialogTitle("Выберите файл");

int result = jFileChooser1.showOpenDialog(frame1.this);

if(result == jFileChooser1.APPROVE\_OPTION){

if(model.getRowCount() > 0){

JOptionPane.showMessageDialog(null, "Таблица перезаписана", "message",0);

list.clear();

model.setRowCount(0);

}

File fileReader = jFileChooser1.getSelectedFile();

ObjectInputStream obj\_in = null;

try{

obj\_in = new ObjectInputStream(new BufferedInputStream(new FileInputStream( fileReader.getAbsolutePath())));

list = (LinkedList<RecIntegral>) obj\_in.readObject();

}catch (IOException ex){

ex.printStackTrace();

}catch (ClassNotFoundException exception){

}

// clear.doClick();

// fill.doClick();

}

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

model.addRow(new Object[]{list.get(row).start,list.get(row).end,list.get(row).step});

};

}

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

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

jFileChooser1.setDialogTitle("Выберите файл");

if(jFileChooser1.showOpenDialog(null) == jFileChooser1.APPROVE\_OPTION){

if(model.getRowCount() > 0){

JOptionPane.showMessageDialog(null, "Таблица перезаписана", "message",0);

list.clear();

model.setRowCount(0);

}

String file = jFileChooser1.getSelectedFile().toString();

long lines = 0;

try {

lines = Files.lines(jFileChooser1.getSelectedFile().toPath()).count();

System.out.println(lines);

} catch (IOException e) {

e.printStackTrace();

}

try{

BufferedReader in = new BufferedReader(new FileReader(file));

String[] s = new String[3];

try {

for(int i = 0; i < lines/3; i++){

for(int j = 0; j < 3; j++){

s[j] = in.readLine();

}

model.insertRow(model.getRowCount(), s);

}

}finally {

in.close();

}

}catch (IOException ex){

throw new RuntimeException(ex);

}

}

}