Министерство науки и образования Российской Федерации  
Пензенский государственный университет  
Кафедра “Вычислительная техника”

**ОТЧЁТ**по лабораторной работе №4  
на тему “Работа с файлами”  
по курсу “Программирование на языке Java”  
Вариант 3

Выполнили:  
ст. гр. 21ВВП1  
Алёшина А.В.  
Нефёдова Е.Д.  
Сорокина Е.А.

Приняли:  
Юрова О.В.  
Карамышева Н.С.

Пенза 2024

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

**Ход работы:**

1. Преобразовали класс RecIntegral так, чтобы мы могли сохранять и восстанавливать объекты класса

public class RecIntegral implements Serializable {

private List<String> record;

public static RecIntegral fromString(String inputData) {

StringBuilder sb =new StringBuilder(inputData);

sb.delete(0,20);

sb.delete(sb.length()-2,sb.length());

List<String> localstring = List.of(sb.toString().split(", "));

return new RecIntegral(localstring);

}

@Override

public String toString() {

return "RecIntegral{" +

"record=" + record +

'}';

}

public List<String> getRecord() {

return record;

}

public void setRecord(List<String> record) {

this.record = record;

}

public void setDataByIndex(int index, String record) {

this.record.set(index, record);

}

public RecIntegral(List<String> record){

this.record = new ArrayList<>(record);

}

}

1. Добавили 4 кнопки:

Save Binary: позволяет сохранять данные таблицы в файл в бинарном формате

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

// TODO add your handling code here:

//DefaultTableModel Table = (DefaultTableModel)jTable1.getModel();

ObjectOutputStream out = null;

int temp = jTable1.getSelectedRow();

if (temp == -1) {

return;

}

try {

out = new ObjectOutputStream(new BufferedOutputStream(

new FileOutputStream("BinaryStringNumber" + temp + ".txt")));

out.writeObject(array.get(temp));

out.close();

} catch (IOException ignored) {

}

}

Load Binary: позволяет загружать в таблицу данные из файла, записанные в бинарном формате

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

// TODO add your handling code here:

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

JFileChooser fileChooser = new JFileChooser(FileSystemView.getFileSystemView().getHomeDirectory());

fileChooser.showOpenDialog(null);

ObjectInputStream in = null;

RecIntegral restObj = null;

int temp = jTable1.getRowCount();

if (temp != -1 ){

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

Table.removeRow(0);

}}

try {

in = new ObjectInputStream(new BufferedInputStream(

new FileInputStream(fileChooser.getSelectedFile().getAbsolutePath())));

restObj = (RecIntegral) in.readObject();

array.add(restObj);

Table.addRow(restObj.getRecord().toArray());

}

catch (IOException | ClassNotFoundException ex ) {

//ex.printStackTrace();

}

}

Save Text: позволяет сохранять данные таблицы в файл в текстовом формате

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

// TODO add your handling code here:

ObjectOutputStream out = null;

int temp = jTable1.getSelectedRow();

if (temp == -1) {

return;

}

try {

out = new ObjectOutputStream(new BufferedOutputStream(

new FileOutputStream("TextStringNumber" + temp + ".txt")));

out.writeObject(array.get(temp).toString());

out.close();

} catch (IOException ignored) {

}

}

Load Text: позволяет загружать в таблицу данные из файла, записанные в текстовом формате

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

// TODO add your handling code here:

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

JFileChooser fileChooser = new JFileChooser(FileSystemView.getFileSystemView().getDefaultDirectory());

fileChooser.showOpenDialog(null);

ObjectInputStream in = null;

RecIntegral restObj = null;

int temp = jTable1.getRowCount();

if (temp != -1 ){

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

Table.removeRow(0);

}

}

try {

in = new ObjectInputStream(new BufferedInputStream(

new FileInputStream(fileChooser.getSelectedFile().getAbsolutePath())));

restObj = RecIntegral.fromString((String) in.readObject());

array.add(restObj);

Table.addRow(restObj.getRecord().toArray());

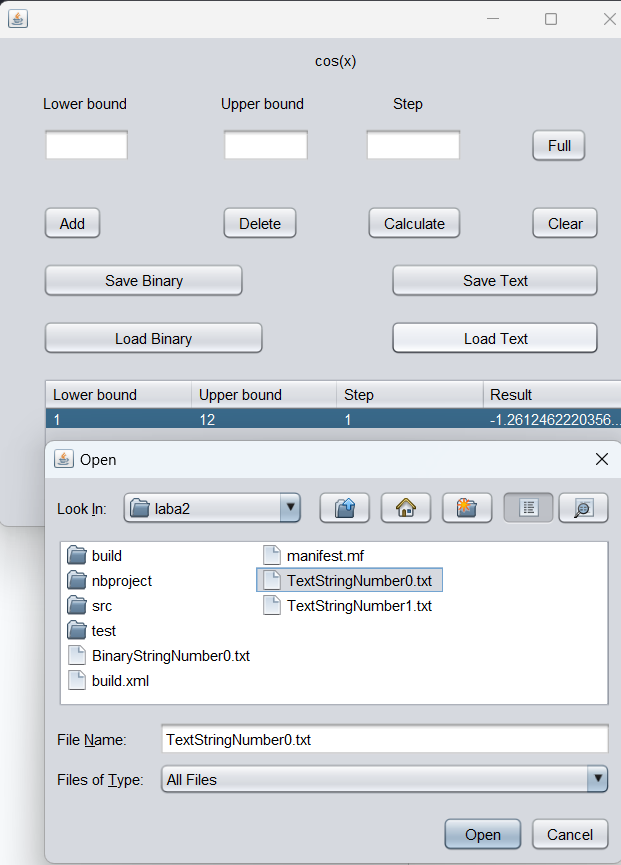
}

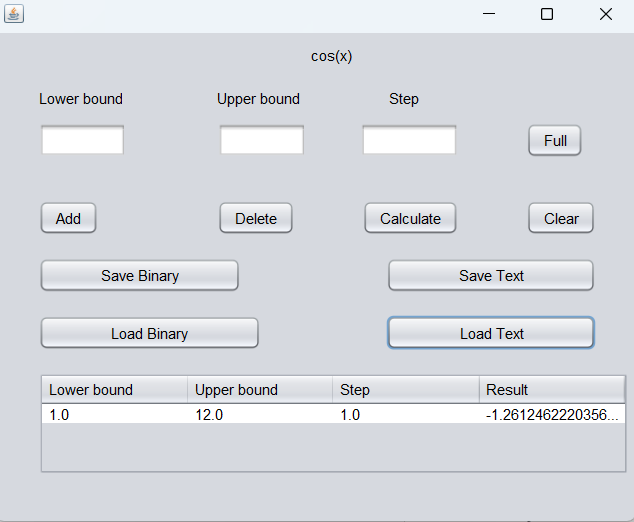
catch (IOException | ClassNotFoundException ex ) {

//ex.printStackTrace();

}

**Результат выполнения работы:**





**Вывод:** в ходе данной лабораторной работы мы изучили работу с файлами и механизмы сериализации данных.