Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 6

з дисципліни: «Кросплатформні засоби програмування»

на тему: «Файли»

**Виконав:**

студент групи КІ-35

Скіпар І.В.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2022

**Мета роботи:** оволодіти навиками використання засобів мови Java для роботи з потоками і файлами.

**Завдання (варіант № 10)**

1. Створити клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі №5. Написати програму для тестування коректності роботи розробленого класу.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання:
5. Охарактеризуйте клас Scanner.
6. Який клас мови Java використовується для здійснення довільного доступу до файлів.

**Текст програми**

Equations.java

package KI35.Skipar.Lab6;  
  
import static java.lang.Math.\*;  
  
*/\*\*  
 \* Class <code>Equations</code> implements method for (tg(x)^2) expression  
 \* calculation  
 \** ***@author*** *Illia  
 \** ***@version*** *1.0  
 \*/*public class Equations {  
 */\*\*  
 \* Method calculate the (tg(x)^2) expression  
 \** ***@param*** *deg angle in degrees  
 \** ***@return*** *expression result  
 \** ***@throws*** *ArithmeticException check correct input  
 \*/* public double calculate(double deg) throws ArithmeticException  
 {  
 double res, rad;  
 rad = *toRadians*(deg);  
  
 try {  
 res = *tan*(rad)\* *tan*(rad);  
 if ((int)(deg/90)%2!=0.0) throw new ArithmeticException();  
 }  
 catch (ArithmeticException e)  
 {  
 throw new ArithmeticException("Invalid X");  
 }  
  
 return res;  
 }  
}

File\_IO.java

package KI35.Skipar.Lab6;  
  
import java.io.\*;  
  
*/\*\*  
 \* Class <code>File\_IO</code> implements input and output to .bin a .txt files  
 \** ***@author*** *Illia  
 \** ***@version*** *1.0  
 \*/*public class File\_IO {  
  
 private FileInputStream in;  
 private BufferedInputStream bin;  
 private FileOutputStream out;  
 private BufferedOutputStream bout;  
  
 */\*\*  
 \* Method for writing result to .txt files  
 \** ***@param*** *fName file name  
 \** ***@param*** *res result  
 \** ***@throws*** *IOException input/output exception  
 \*/* public void writeResTxt(String fName, double res) throws IOException {  
 out = new FileOutputStream(fName);  
 bout = new BufferedOutputStream(out);  
 bout.write(String.*valueOf*(res).getBytes());  
 bout.flush();  
 bout.close();  
 out.close();  
 }  
  
 */\*\*  
 \* Method for reading value from .txt files  
 \** ***@param*** *fName file name  
 \** ***@return*** *values from file  
 \** ***@throws*** *FileNotFoundException when can't found file  
 \*/* public double readResTxt(String fName) throws FileNotFoundException {  
 StringBuilder num = new StringBuilder();  
 var i = 0;  
 try {  
 /\*FileInputStream in = new FileInputStream(fName);\*/  
 in = new FileInputStream(fName);  
 bin = new BufferedInputStream(in);  
 while(( i = bin.read())!=-1) {  
 num.append((char) i);  
 }  
 bin.close();  
 in.close();  
 return Double.*parseDouble*(num.toString());  
 } catch (Exception e) {  
 System.*out*.println(e);  
 }  
 return 0;  
 }  
  
 */\*\*  
 \* Method for writing result to .bin files  
 \** ***@param*** *fName file name  
 \** ***@param*** *res result  
 \** ***@throws*** *IOException input/output exception  
 \*/* public void writeResBin(String fName, double res) throws IOException  
 {  
 out = new FileOutputStream(fName);  
 bout = new BufferedOutputStream(out);  
 DataOutputStream dot = new DataOutputStream(bout);  
 dot.writeDouble(res);  
 dot.flush();  
 dot.close();  
 bout.close();  
 out.close();  
 }  
  
 */\*\*  
 \* Method for reading value from .bin files  
 \** ***@param*** *fName file name  
 \** ***@return*** *result  
 \** ***@throws*** *IOException input/output exception  
 \*/* public double readResBin(String fName) throws IOException  
 {  
 double num;  
 try {  
 in = new FileInputStream(fName);  
 bin = new BufferedInputStream(in);  
 DataInputStream din = new DataInputStream(bin);  
 num = din.readDouble();  
 return num;  
 }  
 catch (Exception e)  
 {  
 System.*out*.print(e);  
 }  
 return 0;  
 }  
}

File\_IO\_App.java

import KI35.Skipar.Lab6.Equations;  
import KI35.Skipar.Lab6.File\_IO;  
  
import java.io.IOException;  
import java.io.RandomAccessFile;  
import java.util.Scanner;  
  
import static java.lang.System.*out*;  
  
public class File\_IO\_App {  
  
 public static void main(String[] args) {  
 try {  
 File\_IO file\_io = new File\_IO();  
 *out*.print("Enter file name: ");  
 Scanner in = new Scanner(System.*in*);  
 String fName = in.nextLine();  
 double deg\_txt = file\_io.readResTxt(fName+".txt");  
 /\*file\_io.writeResBin("file.bin", 44);\*/  
 double deg\_bin = file\_io.readResBin(fName+".bin");  
 try {  
 try {  
 Equations equations = new Equations();  
 file\_io.writeResTxt("result.txt",  
 equations.calculate(deg\_txt));  
 file\_io.writeResBin("result.bin",  
 equations.calculate(deg\_bin));  
 } finally {  
 *out*.print("End program\n");  
 }  
 }  
 catch (ArithmeticException e) {  
 *out*.print(e.getMessage());  
 }  
 *out*.println(deg\_txt);  
 *out*.println(file\_io.readResTxt("result.txt"));  
 *out*.println(deg\_bin);  
 *out*.println(file\_io.readResBin("result.bin"));  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
 }  
}

**Результат виконання програми**

Консоль

Enter file name: file

End program

46.0

1.0723230307791956

44.0

0.9325548097883876

file.txt

46

file.bin

@F

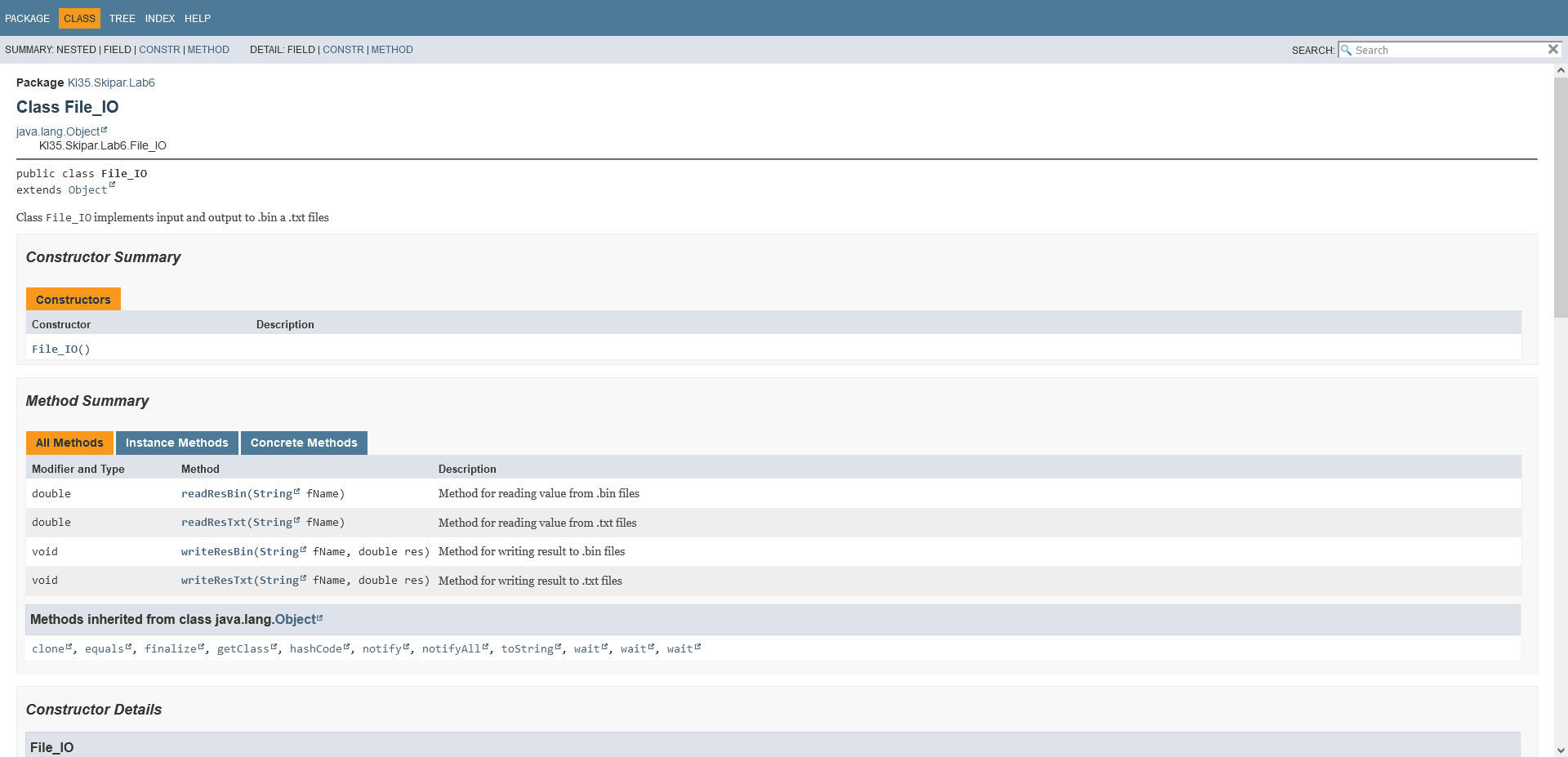
result.txt

1.0723230307791956

result.bin

?нЧ}/8�”

**Фрагмент згенерованої документації**



**Відповіді на контрольні запитання**

1. Клас Scanner найкраще підходить для читання текстових потоків. Він має багато методів, які можуть читати як рядки, так і окремі примітивні тими з подальшим їх перекодуванням до цих типів.

2. Керування файлами з можливістю довільного доступу здійснюється за допомогою класу RandomAccessFile. Його конструктор приймає два параметри: посилання на файл та режим відкривання файлу

**Висновок**

В результаті виконання лабораторної роботи я навчився працювати з потоками вводу/виводу в мові Java.