

Міністерство освіти і науки України  
Національний університет “Львівська політехніка”

Кафедра ЕОМ



### **Звіт**

з лабораторної роботи №6

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Файли»

Виконав: ст.гр. КІ-34

Степанов А. О.

Прийняв:

викл. каф. ЕОМ

Іванов Ю. С.

**Львів 2022**

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

**Завдання:**

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. Для розробленої програми згенерувати документацію.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагмент згенерованої документації.
4. Дати відповідь на контрольні запитання.

**Варіант 21**

$$21. y = \sin(3x - 5) / \operatorname{ctg}(2x)$$

**Лістинг програми:**

**Файл FioApp.java**

```
/**
 * lab 6 package
 */
package KI34.Stepanov.Lab6;

import java.io.*;
import java.util.*;

/**
 * Class <code>FioApp</code> Implements driver for CalcWFio class
 * @author Andriy Stepanov
 * @version 1.0
 */

public class FioApp {
    /**
     * @param args function's parameters
     */
    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        try {
            CalcWFio obj = new CalcWFio();
            Scanner s = new Scanner(System.in);
            System.out.print("Enter data: ");
            int data = s.nextInt();
            obj.calculate(data);
            System.out.println("Result is: " + obj.getResult());

            obj.writeResTxt("textRes.txt");
            obj.writeResBin("BinRes.bin");

            obj.readResBin("BinRes.bin");
            System.out.println("Result is: " + obj.getResult());
            obj.readResTxt("textRes.txt");
            System.out.println("Result is: " + obj.getResult());
        } catch (CalcException ex) {
            // Блок перехоплює помилки обчислень виразу
            System.out.print(ex.getMessage());
        }
    }
}
```

```
}  
}
```

## Файл CalcWFio.java

```
/**  
 * lab 6 package  
 */  
package KI34.Stepanov.Lab6;  
  
import java.io.*;  
import java.util.Locale;  
import java.util.Scanner;  
  
/**  
 * Class <code>CalcWFio</code> implements method for  $\sin(3x-5)/\cos(2x)$   
 expression calculation and methods for writing and reading files  
 */  
 * @author Andriy Stepanov  
 * @version 1.0  
 */  
public class CalcWFio {  
    private double result;  
  
    /**  
     * Method writes txt file  
     *  
     * @param fName File's name  
     * @throws FileNotFoundException error  
     */  
    public void writeResTxt(String fName) throws FileNotFoundException {  
        PrintWriter f = new PrintWriter(fName);  
        f.printf("%f", result);  
        f.close();  
    }  
  
    /**  
     * Method read txt file  
     *  
     * @param fName File's name  
     */  
    public void readResTxt(String fName) {  
        try {  
            File f = new File(fName);  
            if (f.exists()) {  
                Scanner s = new Scanner(f);  
                s.useLocale(Locale.getDefault());  
                result = s.nextDouble();  
                s.close();  
            } else {  
                throw new FileNotFoundException("File " + fName + "not found");  
            }  
        } catch (FileNotFoundException ex) {  
            System.out.print(ex.getMessage());  
        }  
    }  
  
    /**  
     * Method writes bin file  
     *  
     * @param fName File's name  
     * @throws IOException error  
     */  
    public void writeResBin(String fName) throws IOException {  
        DataOutputStream f = new DataOutputStream(new FileOutputStream(fName));  
        f.writeDouble(result);  
    }  
}
```

```

        f.close();
    }

    /**
     * Method reads bin file
     *
     * @param fName File's name
     * @throws IOException error
     */
    public void readResBin(String fName) throws IOException {
        DataInputStream f = new DataInputStream(new FileInputStream(fName));
        result = f.readDouble();
        f.close();
    }

    /**
     * Method calculates the  $\sin(3x-5)/\text{ctg}(2x)$  expression
     *
     * @param x Angle in degrees
     * @throws CalcException
     */
    public void calculate(int x) throws CalcException {
        double radSin, radCtg;
        radSin = (3 * x - 5) * Math.PI / 180.0;
        radCtg = 2 * x * Math.PI / 180.0;
        try {
            result = Math.sin(radSin) / (1.0 / Math.tan(radCtg));

            // Якщо результат не є числом, то генеруємо виключення
            if (result == Double.NaN || result == Double.NEGATIVE_INFINITY ||
                result == Double.POSITIVE_INFINITY || x % 90 == 0 || x == 45
                || (2 * x) % 270.0 == 0)
                throw new ArithmeticException();
        } catch (ArithmeticException ex) {
            // створимо виключення вищого рівня з поясненням причини
            // виникнення помилки
            if (radCtg % Math.PI / 2.0 == 0)
                throw new CalcException("Exception reason: Illegal value of X
for cotangent calculation");
            else if (x == 45 || (2 * x) % 270.0 == 0)
                throw new CalcException("Exception reason: X = 0");
            else
                throw new CalcException("Unknown reason of the exception during
exception calculation");
        }
    }

    /**
     * Method get the result of the expression
     */
    public double getResult() {
        return result;
    }
}

```

## Файл CalcException.java

```
/**
 * lab 6 package
 */
package KI34.Stepanov.Lab6;

/**
 * Class <code>CalcException</code> more precises ArithmeticException
 *
 * @author Andriy Stepanov
 * @version 1.0
 */
public class CalcException extends ArithmeticException {
    public CalcException() {
    }

    public CalcException(String cause) {
        super(cause);
    }
}
```

### Результат виконання програми:

```
Enter X: 90
Exception reason: Illegal value of X for cotangent calculation
Process finished with exit code 0
```

*Виключення для неприпустимого значення для котангеса*

```
Enter X: 45
Exception reason: X = 0
Process finished with exit code 0
```

*Виключення при діленні на нуль*

```
Enter data: 35
Result is: 2.7057370639048854
Result is: 2.7057370639048854
Result is: 2.705737
Process finished with exit code 0
```

*Успішне виконання програми*



textRes.txt - Notepad

File Edit Format View Help

2.705737

*Успішний запис результату у текстовий файл*



BinRes.bin - Notepad

File Edit Format View Help

@YyH\_S

Успішний запис результату у бінарний файл

## Package KI34.Stepanov.Lab6

package KI34.Stepanov.Lab6

All Classes and Interfaces	Classes	Exception Classes
Class	Description	
CalcException	Class CalcException more precises ArithmeticException	
CalcWFio	Class CalcWFio implements method for $\sin(3x-5)/\text{ctg}(2x)$ expression calculation and methods for writing and reading files	
FioApp	Class FioApp Implements driver for CalcWFio class	

Згенерована документація

Package KI34.Stepanov.Lab6

### Class CalcException

java.lang.Object<sup>Ⓜ</sup>  
  java.lang.Throwable<sup>Ⓜ</sup>  
    java.lang.Exception<sup>Ⓜ</sup>  
      java.lang.RuntimeException<sup>Ⓜ</sup>  
        java.lang.ArithmeticException<sup>Ⓜ</sup>  
          KI34.Stepanov.Lab6.CalcException

All Implemented Interfaces:  
Serializable<sup>Ⓜ</sup>

public class CalcException  
extends ArithmeticException<sup>Ⓜ</sup>

Class CalcException more precises ArithmeticException

Version:

1.0

Author:

Andriy Stepanov

See Also:

Serialized Form

#### Constructor Summary

##### Constructors

Constructor	Description
CalcException()	
CalcException(String <sup>Ⓜ</sup> cause)	

#### Method Summary

##### Methods inherited from class java.lang.Throwable<sup>Ⓜ</sup>

addSuppressed<sup>Ⓜ</sup>, fillInStackTrace<sup>Ⓜ</sup>, getCause<sup>Ⓜ</sup>, getLocalizedMessage<sup>Ⓜ</sup>, getMessage<sup>Ⓜ</sup>, getStackTrace<sup>Ⓜ</sup>, getSuppressed<sup>Ⓜ</sup>, initCause<sup>Ⓜ</sup>, printStackTrace<sup>Ⓜ</sup>, printStackTrace<sup>Ⓜ</sup>, printStackTrace<sup>Ⓜ</sup>, setStackTrace<sup>Ⓜ</sup>, toString<sup>Ⓜ</sup>

Інформація про клас CalcException

Class CalcWFio

java.lang.Object<sup>Ⓓ</sup>  
KI34.Stepanov.Lab6.CalcWFio

public class CalcWFio  
extends Object<sup>Ⓓ</sup>

Class CalcWFio implements method for sin(3x-5)/ctg(2x) expression calculation and methods for writing and reading files

Version:

1.0

Author:

Andriy Stepanov

Constructor Summary

Constructors	
Constructor	Description
CalcWFio()	

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	calculate(int x)	Method calculates the sin(3x-5)/ctg(2x) expression
double	getResult()	Method get the result of the expression
void	readResBin(String <sup>Ⓓ</sup> fName)	Method reads bin file
void	readResTxt(String <sup>Ⓓ</sup> fName)	Method read txt file
void	writeResBin(String <sup>Ⓓ</sup> fName)	Method writes bin file
void	writeResTxt(String <sup>Ⓓ</sup> fName)	Method writes txt file

Methods inherited from class java.lang.Object <sup>Ⓓ</sup>
equals <sup>Ⓓ</sup> , getClass <sup>Ⓓ</sup> , hashCode <sup>Ⓓ</sup> , notify <sup>Ⓓ</sup> , notifyAll <sup>Ⓓ</sup> , toString <sup>Ⓓ</sup> , wait <sup>Ⓓ</sup> , wait <sup>Ⓓ</sup> , wait <sup>Ⓓ</sup>

Інформація про клас CalcWFio

Constructor Details

CalcWFio
public CalcWFio()

Method Details

writeResTxt
public void writeResTxt(String <sup>Ⓓ</sup> fName) throws FileNotFoundException <sup>Ⓓ</sup>
Method writes txt file
Parameters: fName - File's name
Throws: FileNotFoundException <sup>Ⓓ</sup> - error
readResTxt
public void readResTxt(String <sup>Ⓓ</sup> fName)
Method read txt file
Parameters: fName - File's name

Інформація про клас CalcWFio

Package KI34.Stepanov.Lab6

Class FioApp

java.lang.Object  
KI34.Stepanov.Lab6.FioApp

public class FioApp  
extends Object

Class FioApp Implements driver for CaleWFio class

Version:

1.0

Author:

Andriy Stepanov

Constructor Summary

Constructors	
Constructor	Description
FioApp()	

Method Summary

All Methods		
Static Methods		
Concrete Methods		
Modifier and Type	Method	Description
static void	main(String[] args)	

Methods inherited from class java.lang.Object	
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Інформація про клас FioApp

Constructor Details

FioApp
public FioApp()

Method Details

main
public static void main(String[] args) throws IOException
Parameters: args - function's parameters
Throws: IOException

Інформація про клас FioApp



## **Відповіді на контрольні запитання:**

### **1. Розкрийте принципи роботи з файловою системою засобами мови Java.**

Для створення файлових потоків і роботи з ними у Java є 2 класи, що успадковані від `InputStream` і `OutputStream` це - `FileInputStream` і `FileOutputStream`. Як і їх суперкласи вони мають методи лише для байтового небуферизованого блокуючого читання/запису даних та керування потоками. На відміну від, наприклад, мови програмування C, де для виконання усіх можливих операцій з файлами необхідно мати один вказівник на `FILE` у мові Java реалізовано інший набагато складніший і гнучкіший підхід, який дозволяє формувати такі властивості потоку, які найкраще відповідають потребам рішення конкретної задачі. Так у Java розділено окремі функціональні можливості потоків на різні класи. Компонуючи ці класи між собою і досягається необхідна кінцева функціональність потоку.

### **2. Охарактеризуйте клас `Scanner`.**

Для читання текстових потоків найкраще підходить клас `Scanner`. На відміну від `InputStreamReader` і `FileReader`, що дозволяють лише читати текст, він має велику кількість методів, які здатні читати як рядки, так і окремі примітивні типи з подальшим їх перекодуванням до цих типів, робити шаблонний аналіз текстового потоку, здатний працювати без потоку даних та ще багато іншого.

### **3. Наведіть приклад використання класу `Scanner`.**

Приклад читання даних за допомогою класу `Scanner` з стандартного потоку вводу:

```
Scanner sc = new Scanner(System.in);
```

```
int i = sc.nextInt();
```

Приклад читання даних за допомогою класу `Scanner` з текстового файлу:

```
Scanner sc = new Scanner(new File("myNumbers"));
```

```
while (sc.hasNextLong()) {
```

```
    long aLong = sc.nextLong();
```

```
}
```

#### **4. За допомогою якого класу можна здійснити запис у текстовий потік?**

Для буферизованого запису у текстовий потік найкраще використовувати клас `PrintWriter`.

#### **5. Охарактеризуйте клас `PrintWriter`.**

#### **6. Розкрийте методи читання/запису двійкових даних засобами мови `Java`.**

Читання двійкових даних примітивних типів з потоків здійснюється за допомогою класів, що реалізують інтерфейс `DataInput`, наприклад класом `DataInputStream`. Інтерфейс `DataInput` визначає такі методи для читання двійкових даних:

- `readByte`;
- `readInt`;
- `readShort`;
- `readLong`;
- `readFloat`;
- `readDouble`;
- `readChar`;
- `readBoolean`;
- `readUTF`.

Запис двійкових даних примітивних типів у потоки здійснюється за допомогою класів, що реалізують інтерфейс `DataOutput`, наприклад класом `DataOutputStream`. Інтерфейс `DataOutput` визначає такі методи для запису двійкових даних:

- `writeByte`;
- `writeInt`;
- `writeShort`;
- `writeLong`;
- `writeFloat`;
- `writeDouble`;
- `writeChar`;
- `writeBoolean`;
- `writeUTF`.

## **7. Призначення класів `DataInputStream` і `DataOutputStream`.**

Класи `DataOutputStream` і `DataInputStream` дозволяють записувати і зчитувати дані примітивних типів.

Клас `DataOutputStream` представляє потік виведення і призначений для запису даних примітивних типів, таких, як `int`, `double` і т.д. Для запису кожного з примітивних типів призначений свій метод.

Клас `DataInputStream` діє протилежним чином - він зчитує з потоку дані примітивних типів. Відповідно для кожного примітивного типу визначено свій метод для зчитування.

## **8. Який клас мови Java використовується для здійснення довільного доступу до файлів.**

Керування файлами з можливістю довільного доступу до них здійснюється за допомогою класу `RandomAccessFile`.

## **9. Охарактеризуйте клас `RandomAccessFile`.**

Керування файлами з можливістю довільного доступу до них здійснюється за допомогою класу `RandomAccessFile`. Відкривання файлу в режимі запису і читання/запису здійснюється за допомогою конструктора, що приймає 2 параметри – посилання на файл (`File file`) або його адресу (`String name`) та режим відкривання файлу (`String mode`).

Файли, що керуються класом `RandomAccessFile`, оснащені вказівником на позицію наступного байту, що має читатися або записуватися. Для того, щоб перемістити даний вказівник на довільну позицію в межах файлу використовується метод `void seek(long pos)`. Параметр `long pos` визначає номер байту, що має читатися або записуватися.

## **10. Який зв'язок між інтерфейсом `DataOutput` і класом `DataOutputStream`?**

Запис двійкових даних примітивних типів у потоки здійснюється за допомогою класів, що реалізують інтерфейс `DataOutput`, наприклад класом `DataOutputStream`.

## **Висновок:**

На цій лабораторній роботі я оволодів навиками використання засобів мови Java для роботи з потоками і файлами.