Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ

Звіт

З лабораторної роботи №6

Варіант – 21

З дисципліни: «Кросплатформенні засоби програмування»

На тему: «Файли»

Виконав: ст. гр. КІ-35

Сухан Д. В.

Перевірив: доцент кафедри ЕОМ

Іванов Ю.С.

Львів 2022

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

Завдання (варіант № 21)

21. y=sin(3x-5)/ctg(2x)

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.

Хід роботи:

Текст програми

CalcException.java

```
package KI35.Sukhan.Lab6;

public class CalcException extends ArithmeticException {
    public CalcException(){}

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

CalcWFio.java

```
package KI35.Sukhan.Lab6;

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

```
public class CalcWFio {
    private double  result ,sin_rad ,tg_rad;

    public void writeResTxt(String fName) throws
FileNotFoundException
    {
        PrintWriter f = new PrintWriter(fName);
        f.printf("%f ",result);
        f.close();
    }

    public void readResTxt(String fName)
    {
        try
        {
            File f = new File (fName);
            if (f.exists())
            {
                Scanner s = new Scanner(f);
                result = s.nextDouble();
                s.close();
            }
            else
                throw new FileNotFoundException("File " +
fName + "not found");
        }
        catch (FileNotFoundException ex)
        {
            System.out.print(ex.getMessage());
        }
    }

    public void writeResBin(String fName) throws
FileNotFoundException, IOException
    {
```

```
        DataOutputStream f = new DataOutputStream(new
FileOutputStream(fName));
        f.writeDouble(result);
        f.close();

    }

    public void readResBin(String fName) throws
FileNotFoundException, IOException
    {
        DataInputStream f = new DataInputStream(new
FileInputStream(fName));
        result = f.readDouble();
        f.close();
    }

    public double calculate(double x) throws
CalcException{

        sin_rad = Math.toRadians(3*x - 5);
        tg_rad = Math.toRadians(2*x);

        try {
            result = Math.sin(sin_rad)*(
Math.tan(tg_rad));
            if (result==Double.NaN ||
result==Double.NEGATIVE_INFINITY ||
result==Double.POSITIVE_INFINITY || 2*x == 90 || 2*x == -
90 )
                throw new ArithmeticException();
        }
        catch(ArithmeticException ex){
            if (tg_rad == Math.PI/2.0 || tg_rad == -
Math.PI/2.0)
                throw new CalcException("Exception reason:
Illegal value of X for tangent calculation");
            else
```

```
                throw new CalcException("Unknown reason of
the exception during exception calculation");
        }

        System.out.println(result);
        return result;
    }

    public double getResult()
    {
        return result;
    }
}
```

FioApp.java

```
package KI35.Sukhan.Lab6;

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

public class FioApp {
    public static void main(String[] args) throws IOException {
        boolean IsCatched = true;
         CalcWFio obj = new CalcWFio();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter data: ");
        double data = s.nextDouble();

        try
        {
            obj.calculate(data);
        }
        catch (CalcException ex)
        {
```

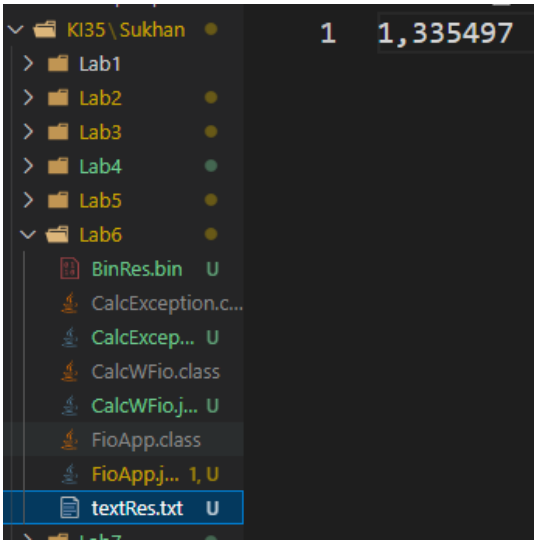
```
            IsCatched =
false;

            System.out.println(ex.getMessage());
        }

        if(IsCatched) {


            System.out.println("Result is: " + obj.getResult());
            obj.writeResTxt("KI35/Sukhan/Lab6/textRes.txt");
            obj.writeResBin("KI35/Sukhan/Lab6/BinRes.bin");

            obj.readResBin("KI35/Sukhan/Lab6/BinRes.bin");
            System.out.println("Result is: " + obj.getResult());
            obj.readResTxt("KI35/Sukhan/Lab6/textRes.txt");
            System.out.println("Result is: " + obj.getResult());

            RandomAccessFile file = new RandomAccessFile("Res.txt", "rw");
            file.write("Sukhan info".getBytes());
            file.close();
        }
 }
}
```

Результат виконання програми

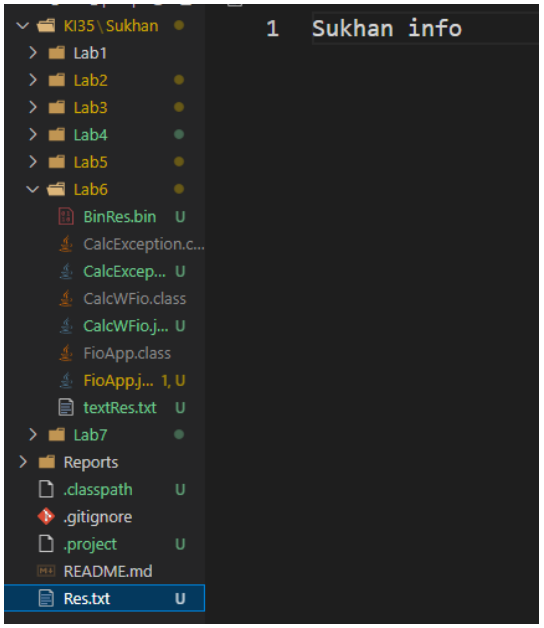
```
PROBLEMS 6   OUTPUT   DEBUG CONSOLE   TERMINAL

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS R:\5_semester\KZP\CPPT_Sukhan_DV_KI-35_2>  & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe'
Enter data: 27
1.335497495156728
Result is: 1.335497495156728
Result is: 1.335497495156728
Result is: 1.335497
PS R:\5_semester\KZP\CPPT_Sukhan_DV_KI-35_2>
```

Вміст файлу BinRes.bin

Вміст файлу textRest.txt

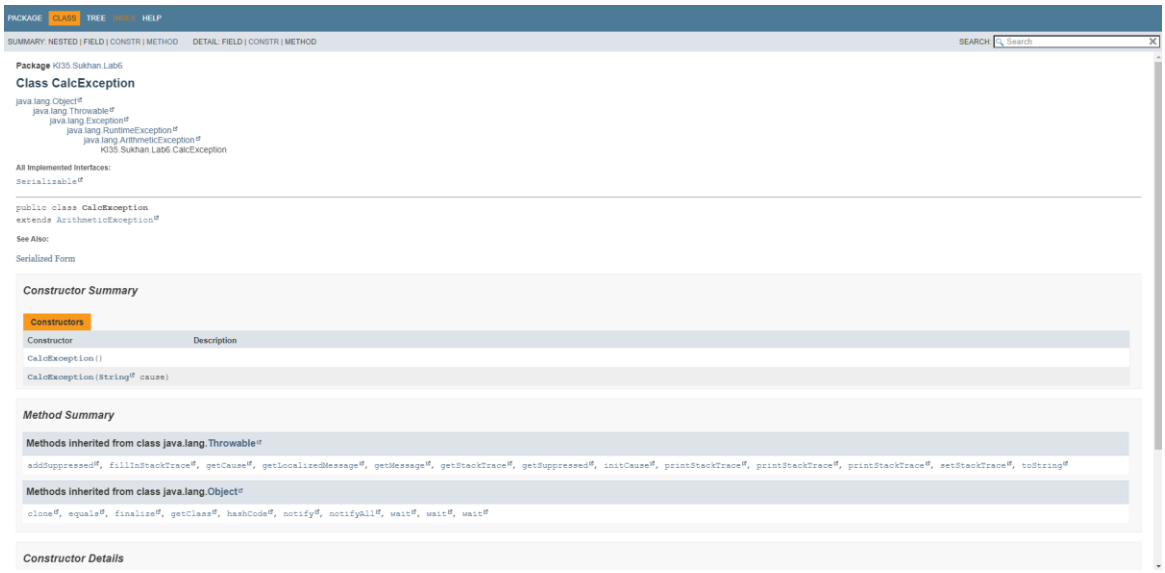


Вміст файлу Res.txt



Фрагмент згенерованої документації

CalcException.html



PACKAGE CLASS TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH: Search

Package KI35.Sukhan.Lab6

Class CalcException

java.lang.Object
java.lang.Throwable
java.lang.Exception
java.lang.RuntimeException
java.lang.ArithmeticException
KI35.Sukhan.Lab6.CalcException

All Implemented Interfaces:
Serializable

public class CalcException
extends ArithmeticException

See Also:
Serialized Form

Constructor Summary

Constructors

Constructor	Description
CalcException()	
CalcException(String cause)	

Method Summary

Methods inherited from class java.lang.Throwable

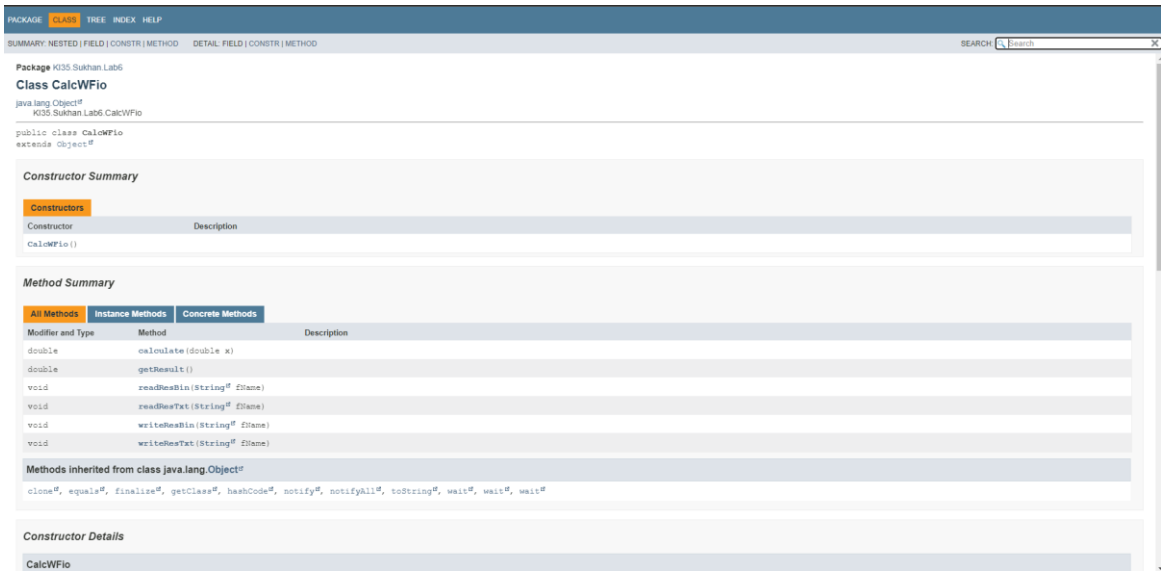
addSuppressed, fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, getSuppressed, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Details

CalcWFio.html



PACKAGE CLASS TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH: Search

Package KI35.Sukhan.Lab6

Class CalcWFio

java.lang.Object
KI35.Sukhan.Lab6.CalcWFio

public class CalcWFio
extends Object

Constructor Summary

Constructors

Constructor	Description
CalcWFio()	

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
double	calculate(double x)	
double	getResult()	
void	readResBin(String fName)	
void	readResTxt(String fName)	
void	writeResBin(String fName)	
void	writeResTxt(String fName)	

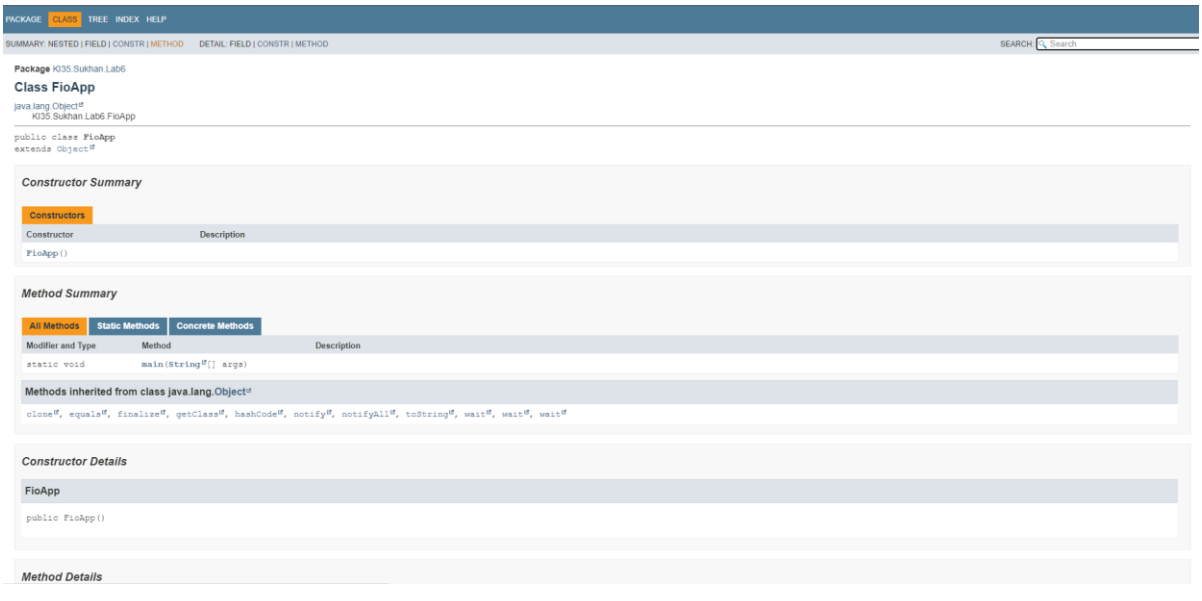
Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

CalcWFio

FioApp.html



PACKAGE CLASS TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH: Search

Package KI35.Sukhan.Lab6

Class FioApp

java.lang.Object
KI35.Sukhan.Lab6.FioApp

public class FioApp
extends Object

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

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

FioApp

public FioApp()

Method Details

Відповідь на контрольні запитання:

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

Відповідь: 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 керуванням потоками.

2. Охарактеризуйте клас Scanner.

Відповідь: 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.

Відповідь: Scanner sc = new Scanner(new File("file"));

```
while (sc.hasNext()) {
 String sentence = sc.nextLine();
}
```

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

Відповідь: Для буферизованого запису у текстовий потік найкраще використовувати клас PrintWriter.

5. Охарактеризуйте клас PrintWriter.

Відповідь: Цей клас має методи для виводу рядків і чисел у текстовому форматі: print, println, printf

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

Відповідь: Читання двійкових даних примітивних типів з потоків здійснюється за допомогою класів, що реалізують інтерфейс DataInput, наприклад класом DataInputStream.

7. Призначення класів DataInputStream і DataOutputStream.

Відповідь: DataInputStream клас з методами для читання двійкових даних. DataOutputStream запис.

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

Відповідь: RandomAccessFile

9. Охарактеризуйте клас RandomAccessFile.

Відповідь: 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):

RandomAccessFile(File file, String mode);

RandomAccessFile(String name, String mode)

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

Відповідь: DataOutputStream імплементує інтерфейс DataOutput

Висновок: Я оволодів навиками використання засобів мови Java для роботи з потоками і файлами.