

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

Лис Б. Л.

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 11
 $Y = \text{ctg}(x)/\text{tg}(x)$

Лістинг програми:

Файл EquationsApp.java

```
/**
 * lab 6 package
 */
package KI34.Lys.Lab6;
import java.util.*;
import java.io.*;

/**
 * Class <code>EquationsApp</code> Implements driver for Equations class
 * @author Lys Bohdan
 * @version 1.0
 */
public class EquationsApp {
    /**
     * @param args function's parameters
     */
    public static void main(String[] args) throws IOException {
        try {
            Equations obj = new Equations();
            Scanner s = new Scanner(System.in);
            System.out.print("Enter data: ");
            double data = s.nextDouble();
            obj.calculate(data);
            System.out.println("Result from program is: " + obj.getResult());

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

            obj.readResBin("BinRes.bin");
            System.out.println("Result from Bin file is: " + obj.getResult());
            obj.readResTxt("textRes.txt");
            System.out.println("Result from txt file is: " + obj.getResult());
        } catch (CalcException | IllegalArgumentException ex) {
            System.out.print(ex.getMessage());
        }
    }
}
```

Файл Equations.java

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

import java.io.*;
import java.util.Locale;
import java.util.Scanner;

/**
 * Class <code>Equations</code> implements method for tan(x)/ctg(x) expression
 * calculation and methods for writing and reading files
 * @author Lys Bohdan
 * @version 1.0
 */

public class Equations {
    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 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 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 calculates the  $\cotg(x)/tg(x)$  expression
     * @param x Angle in degrees
     * @throws CalcException
     */
    public double calculate(double x) throws CalcException {
        try {
            result = Math.tan(x) / (1.0 / Math.tan(x));

            if (result == Double.NaN || result == Double.NEGATIVE_INFINITY ||
                result == Double.POSITIVE_INFINITY || x % 90 == 0)
                throw new ArithmeticException();
        } catch (ArithmeticException ex) {

            if (x % 180 == 0)
                throw new CalcException("Exception reason: Illegal value of X
for cotangent calculation");
            else if (x % 90 == 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");
        }

        return result;
    }

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

```

Файл CalcException.java

```

/**
 * lab 6 package
 */
package KI34.Lys.Lab6;
/**

```

```

* Class <code>CalcException</code> more precises ArithmeticException
* @author Lys Bohdan
* @version 1.0
*/
public class CalcException extends ArithmeticException{
    public CalcException(){}
    public CalcException(String cause)
    {
        super(cause);
    }
}

```

Результат виконання програми:

```

C:\Users\mrpet\.jdk\openjdk-18.0.2\bin\java.exe "-javaagent:C:\
Enter data: 36
Result from program is: 60.06979926008897
Result from Bin file is: 60.06979926008897
Result from txt file is: 60.069799

Process finished with exit code 0

```

Успішне виконання програми

 textRes.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

60,069799

Успішний запис результату у текстовий файл

 BinRes.bin: Блокнот

Файл Редагування Формат Вигляд Довідка

@Nл.Ўю

Успішний запис результату у бінарний файл

Package KI34.Lys.Lab6

package KI34.Lys.Lab6

All Classes and Interfaces	Classes	Exception Classes
Class	Description	
CalcException	Class CalcException more precises ArithmeticException	
Equations	Class Equations implements method for tan(x)/ctg(x) expression calculation and methods for writing and reading files	
EquationsApp	Class EquationsApp Implements driver for Equations class	

Згенерована документація

Package KI34.Lys.Lab6

Class CalcException

java.lang.Object[Ⓜ]
 java.lang.Throwable[Ⓜ]
 java.lang.Exception[Ⓜ]
 java.lang.RuntimeException[Ⓜ]
 java.lang.ArithmeticException[Ⓜ]
 KI34.Lys.Lab6.CalcException

All Implemented Interfaces:

Serializable[Ⓜ]

```
public class CalcException
extends ArithmeticExceptionⓂ
```

Class CalcException more precises ArithmeticException

Version:

1.0

Author:

Lys Bohdan

See Also:

Serialized Form

Constructor Summary

Constructors	
Constructor	Description
<code>CalcException()</code>	
<code>CalcException(String[Ⓜ] cause)</code>	

Інформація про клас CalcException

Class Equations

java.lang.Object[Ⓓ]
KI34.Lys.Lab6.Equations

public class Equations
extends Object[Ⓓ]

Class Equations implements method for tan(x)/ctg(x) expression calculation and methods for writing and reading files

Version:

1.0

Author:

Lys Bohdan

Constructor Summary

Constructors	
Constructor	Description
Equations()	

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
double	calculate(int x)	Method calculates the ctg(x)/tg(x) expression
double	getResult()	Method get the result of the expression
void	readResBin(String [Ⓓ] fName)	Method reads bin file
void	readResTxt(String [Ⓓ] fName)	Method read txt file
void	writeResBin(String [Ⓓ] fName)	Method writes bin file
void	writeResTxt(String [Ⓓ] fName)	Method writes txt file

Інформація про клас Equations

Package `Kl34.Lys.Lab6`

Class `EquationsApp`

`java.lang.Object`

`Kl34.Lys.Lab6.EquationsApp`

```
public class EquationsApp
extends Object
```

Class `EquationsApp` Implements driver for `Equations` class

Version:

1.0

Author:

Lys Bohdan

Constructor Summary

Constructors

Constructor	Description
-------------	-------------

<code>EquationsApp()</code>	
-----------------------------	--

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
-------------------	--------	-------------

<code>static void</code>	<code>main(String[] args)</code>	
--------------------------	----------------------------------	--

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Інформація про клас `EquationsApp`

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.