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.

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.

Bаріант 11
$$Y = ctg(x)/tg(x)$$

Лістинг програми:

Файл EquationsApp.java

```
oublic class EquationsApp {
    * @param args function's parameters
           obj.calculate(data);
```

Файл Equations.java

```
import java.util.Scanner;
* @version 1.0
   public void readResTxt(String fName) {
               result = s.nextDouble();
                s.close();
     * @param fName File's name
```

```
* @param fName File's name
* @throws IOException error
  DataOutputStream f = new DataOutputStream(new FileOutputStream(fName));
* @param x Angle in degrees
* @throws CalcException
```

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);
   }
}
```

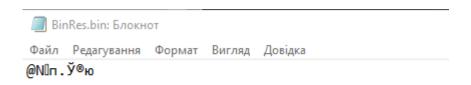
Результат виконання програми:

```
C:\Users\mrpet\.jdks\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



Успішний запис результату у текстовий файл



Успішний запис результату у бінарний файл

Package KI34.Lys.Lab6

package KI34.Lys.Lab6

All Classes and Ir	nterfaces Class	Exception Classes
Class	Description	
CalcException	Class CalcExcep	on more precises ArithmeticException
Equations	Class Equations	$ mplements \ method \ for \ tan(x)/ctg(x) \ expression \ calculation \ and \ methods \ for \ writing \ and \ reading \ files$
EquationsApp	Class Equations	p Implements driver for Equations class

Згенерована документація

Package KI34.Lys.Lab6

java.lang.Object^{td}
java.lang.Throwable^{td}
java.lang.Exception^{td}
java.lang.RuntimeException^{td}
java.lang.ArithmeticException^{td}
Kl34.Lys.Lab6.CalcException

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 CalcException() CalcException(String[™] cause)

Інформація про клас CalcException

Package Kl34.Lys.Lab6

Class Equations

java.lang.Object[®] Kl34.Lys.Lab6.Equations

public class Equations extends $\texttt{Object}^{\, \underline{\sigma}}$

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

Version:

1.0

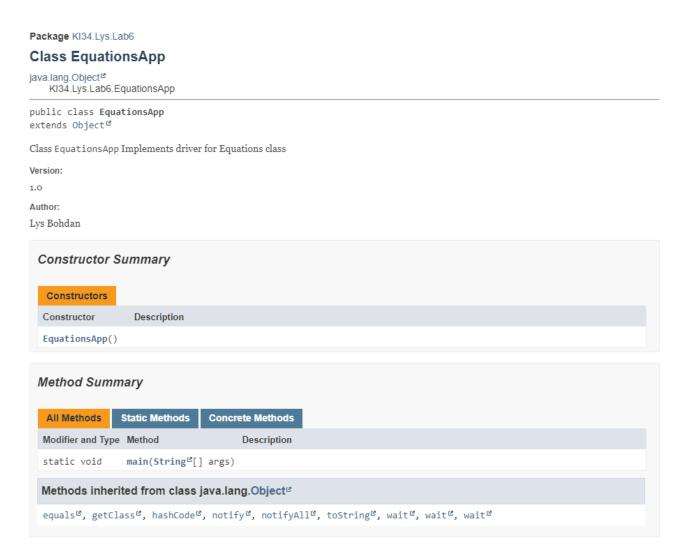
Author:

Lys Bohdan

Constructor Summary Constructors Constructor Description Equations()

lethod Sumn	nary	
All Methods	Instance Methods Concre	te Methods
Modifier and Type	Method	Description
double	<pre>calculate(int x)</pre>	Method ca
double	<pre>getResult()</pre>	Method ge
void	readResBin(String ^{td} fName) Method r
void	readResTxt(String ^{td} fName) Method r
void	writeResBin(String ^ଔ fNam	e) Method w
void	writeResTxt(String [™] fNan	e) Method w

Інформація про клас Equations



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 ϵ 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. На відміну від, наприклад, мови програмування С, 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 i 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.