Lab3- Тексты и файлы в Java

Файлы в Java

В Java множество классов и методов помогают работать с файлами. Некоторые из них перечислены в таблице ниже.

Операция	Метод	Класс
Создание файла	createNewFile()	java.io.File
Чтение файла	read()	java.io.FileReader
Чтение файла построчно	nextLine()	java.util.Scanner;
Запись файла	write()	java.io.FileWriter
Удаление файла	delete()	java.io.File

String vs StringBuilder vs StringBuffer

Строки в Java реализованы в виде объектов класса String. Они финализированы и неизменяемы, вследствие этого при любых манипуляциях с ними всегда создается новая строка, что делает работу со строками весьма ресурсоёмким процессом. Если строки необходимо часто менять, то для этого есть StringBuffer и StringBuilder.

Разница между ними заключается в следующем:

- String финализирован и неизменяемы (final), тогда как StringBuffer и StringBuilder являются изменяемыми классами.
- StringBuffer является потокобезопасным и синхронизированным, тогда как StringBuilder — нет. Вот почему StringBuilder быстрее, чем StringBuffer.

Для манипуляций со строками в среде без многопоточности лучше использовать StringBuilder, иначе использовать класс StringBuffer.

Документацию по этим классам можно найти по следующим ссылкам:

1- String
2- StringBuilder
3- StringBuffer

Пример

```
package com.finu;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.Scanner;

public class Main {
    public final static String emptyFileName="file1.txt";
    public final static String textFileName="file2.txt";
    public static void main(String[] args) {

        /************** File creation and Deletion **************/
        System.out.println(String.format("Creating an empty file
[%s]....",emptyFileName));
        File file = new File(emptyFileName);
        try {
            if (file.createNewFile()) {
                System.out.println("File creation successful!");
```

```
            } else {
                System.out.println("File creation failed!");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        System.out.println(String.format("Deleting file [%s]....",emptyFileName));
        System.out.println("Do you want to delete the file (Y or y for Yes)?");
        Scanner in=new Scanner(System.in);
        char choice=in.nextLine().charAt(0);
        if(choice=='Y' || choice=='y')
        {
            try {
                if (file.delete()) {
                    System.out.println("File deletion successful!");
                } else {
                    System.out.println("File deletion failed!");
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        /****************** Writing to a text file ******************/
        System.out.println(String.format("Writing to file [%s]",textFileName));
        try {
            FileWriter myWriter = new FileWriter(textFileName);
            myWriter.write("First line\nSecond line\nThird line\nForth Line");
            myWriter.close();
            System.out.println("Text saved successfully.");
        } catch (Exception e) {
            System.out.println("Text saving failed.");
            e.printStackTrace();
        }

        /****************** Reading from a text file ************************/
        // using FileReader
        System.out.println(String.format("Reading file [%s] using
FileReader",textFileName));
        StringBuilder sb= new StringBuilder();
        try {
            FileReader myReader = new FileReader(textFileName);
            int character=myReader.read();
            while(character!=-1)
            {
                //System.out.print((char) character);
                sb.append((char) character);
                character=myReader.read();
            }
            myReader.close();
        } catch (Exception e) {
            System.out.println("Text reading failed.");
            e.printStackTrace();
        }
        System.out.println(sb.toString());

        // using Scanner
        System.out.println(String.format("Reading file[%s] using
Scanner",textFileName));
        try {
            File myFile = new File(textFileName);
            Scanner myReader = new Scanner(myFile);
            while (myReader.hasNextLine()) {
```

```
                String line = myReader.nextLine();
                System.out.println(line);
            }
            myReader.close();
        } catch (Exception e) {
            System.out.println("Text reading failed.");
            e.printStackTrace();
        }
    }
}
```

Задача:

Напишите программу позволяющую выполнить следующее:

1- предоставить пользователю возможность ввода текста из текстового файла "Input.txt".

1- подсчитать количество слов в тексте.

2- подсчитать количество прописных и строчных букв в тексте и их сумма.

3- подсчитать количество пробелов в тексте.

4- подсчитать количество целых чисел и чисел с плавающей запятой. В случае обнаружения целых чисел программа должна вывести их в шестнадцатеричном формате. В случае обнаружения чисел с плавающей точкой программа должна вывести их с двумя десятичными знаками (используйте метод String.format()).

5- подсчитать количество знаков препинания (предположим, что там только " , . !?). Используйте класс StringBuilder, чтобы удалить знаки препинания из текста.

6- сохранить текст со всеми результатами в текстовом файле под названием "Results.txt".

8- предоставить пользователю возможность искать слово в текстовом файле "Input.txt". Результат должен быть указан в виде двух чисел для каждого появления слова: индекс первой буквы и индекс последней буквы. Следует учитывать многократное появление одного слова.