

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №2

По дисциплине: «Современные платформы программирования»

Выполнил:
студент 3 курса
группы ПО-8
Таразевич Н.А.

Проверил:
Крощенко А.А.

Брест, 2024

Цель работы: приобрести базовые навыки работы с файловой системой в Java.

Задание 1: Напишите программу, которая выводит слова, располагая их в порядке

убывания частоты их появления. Перед каждым словом должно быть число его появлений.

Код программы:

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

public class Task1 {
    private static String filePath = "src/task1.txt";
    public static void main(String[] args) throws
FileNotFoundException {
        File file = new File(filePath);
        Scanner scanner = new Scanner(file);

        Map<String, Integer> wordCount = new HashMap<>();
        String word;

        while (scanner.hasNext()) {
            word = scanner.next();

            word = word.replaceAll("[^a-zA-Z0-9']", "");
            if (!word.isEmpty()) {
                wordCount.put(word,
                               wordCount.getDefault(word, 0) + 1);
            }
        }
        scanner.close();

        List<Map.Entry<String, Integer>> wordCountSorted = new
ArrayList<>(wordCount.entrySet());

wordCountSorted.sort(Map.Entry.comparingByValue(Collections.reverseOr
der()));

        for (Map.Entry<String, Integer> pair : wordCountSorted) {
            System.out.println(pair.getValue() + " - " +
pair.getKey());
        }
    }
}
```

Входные данные: Hello world, Hello world. Hello Nikita!?

Результат работы программы:

D:\SDK\JDK\bin\java.exe

3 - Hello

2 - world

1 - Nikita

Задание 2: Утилита `nl` выводит переданный файл в стандартный вывод или в другой файл, выполняя нумерацию его строк. Если файл не задан или задан как `-`, читает стандартный ввод.

Формат использования: `nl [-i] [-l] [-n] входной_файл [выходной_файл]`

- `-i` ЧИСЛО Задаёт шаг увеличения номеров строк
- `-l` 1/0 Задаёт флаг нумерации пустых строк
- `-n` ФОРМАТ Использовать заданный формат для номеров строк.

`ln` – номер выровнен по левому краю, без начальных нулей

`rn` – номер выровнен по правому краю, без начальных нулей

`rz` – номер выровнен по правому краю с начальными нулями

Пример использования: `nl -i 2 -l 0 -n ln in.txt`

Обрабатывает файл `in.txt`, выводит результат в стандартный вывод, инкремент счетчика равен двум (`-i 2`), пустые строки не нумеруются.

Код программы:

```
import java.io.File;

import java.io.FileOutputStream;
import java.io.PrintStream;
import java.util.*;

public class Task2 {

    private static final int optionPattern = 0;
    private static final int optionValue = 1;

    public static void main(String[] args) throws Exception {
        if (args.length == 0 || !args[0].equals("nl")) {
            System.err.println("Utility name - nl");
            System.exit(1);
        }

        Map<String, ArrayList<String>> options = new HashMap<>();
        options.put("-i", new ArrayList<>(Arrays.asList("^([1-9][0-9])*", "1")));
        options.put("-l", new ArrayList<>(Arrays.asList("[10]", "1")));
        options.put("-n", new ArrayList<>(Arrays.asList("ln|rn|rz", "ln")));

        ArrayList<String> params = new ArrayList<>();
```

```

for (int i = 1; i < args.length; i++){
    if (args[i].startsWith("-") && args[i].length() >
        1){ if (params.isEmpty()

                &&
                options.containsKey(args[i])
                && i < args.length + 1

                && args[i+1].matches(
                    options.get(args[i]).get(optionPattern)
                ))

        {
            options.put(args[i], new
                ArrayList<>( Arrays.asList(

                    options.get(args[i]).get(optionPattern)
                ), args[++i])

            )
        };

    }

    else {

        System.out.println("Wrong option " +
            args[i]); System.exit(1);

    }

}

else {
    params.add(args[i]
    );
}

}

if (params.isEmpty() || params.size() > 2){
    System.out.println("Wrong params
        count"); System.exit(1);
}

if (params.size() == 2

        && params.getFirst().equals(params.getLast())){
    System.out.println("Input file can't be output
        file"); System.exit(1);
}

Scanner
scanner; File
inFile;

```

```

if (!params.getFirst().equals("-")) {
    inFile = new
    File(params.getFirst()); if
    (!inFile.isFile()){

        System.err.println("Input file doesn't
        exist"); System.exit(1);

    }

    scanner = new
    Scanner(inFile); }
else {

    scanner = new
    Scanner(System.in); }

PrintStream outputStream;
FileOutputStream
fileOutputStream; if
(params.size() == 2){

    File outFile = new
    File(params.getLast()); if
    (!outFile.isFile()){

        if (!outFile.createNewFile()){
            System.err.println("Failed to create
            output
            fi
            le"); System.exit(1);

        }

    }

    fileOutputStream = new
    FileOutputStream(params.getLast()); outputStream = new
    PrintStream(fileOutputStream);

}

else {

    outputStream = new
    PrintStream(System.out); }

int step = Integer.parseInt(options

                                .get("-
i").get(optionValue)); boolean countEmptyLines =
"1".equals(options

                                .get("-l")

```

```

        .get(optionValue));

String alignType = options.get("-
n").get(optionValue); String line;
for (int lineNumber = 1;
    scanner.hasNextLine();) { line =
    scanner.nextLine();

    if
        (line.equals("\
q")) break;

    if (countEmptyLines ||
        !line.isEmpty()) { switch
        (alignType) {

            case "ln":

                line = lineNumber + " -- " +
                line; break;

            case "rn":

                line = line + " -- " +
                lineNumber; break;

            case "rz":

                line = line + " -- "

                    + String.format("%4s",
                        lineNumber) .replace(' ',
                        '0');

                br
                eak; }

            lineNumber +=
            step; }

        outputStream.println(li
        ne); }

    }

}

```

Входные данные:

Файл inFile.txt

```
1 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2 Curabitur vulputate ante sapien, non aliquet risus laoreet ut.
3 Proin molestie quam tortor, consequat rhoncus neque rhoncus eu.
4 Aenean vel lorem in nunc cursus ultrices.
5 Ut rhoncus rutrum fermentum. Ut nec dolor urna.
6 In non congue orci.
7
8 Nulla luctus malesuada erat, ac faucibus felis ornare nec.
9 Integer ac pellentesque odio.
10 Vivamus lacinia cursus leo, non volutpat nibh condimentum in.
11
12 Pellentesque a nisi est. Suspendisse lobortis elit a libero lacinia,
13 quis iaculis nulla pharetra.
```

Файл run.bat

```
1 java -jar spp_lab2.jar nl -i 2 -l 0 -n rz inFile.txt outFile.txt
```

Результат работы программы:

Полученный файл outFile.txt

```
1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. -- 0001
2 Curabitur vulputate ante sapien, non aliquet risus laoreet ut. -- 0003
3 Proin molestie quam tortor, consequat rhoncus neque rhoncus eu. -- 0005
4 Aenean vel lorem in nunc cursus ultrices. -- 0007
5 Ut rhoncus rutrum fermentum. Ut nec dolor urna. -- 0009
6 In non congue orci. -- 0011
7
8 Nulla luctus malesuada erat, ac faucibus felis ornare nec. -- 0013
9 Integer ac pellentesque odio. -- 0015
10 Vivamus lacinia cursus leo, non volutpat nibh condimentum in. -- 0017
11
12 Pellentesque a nisi est. Suspendisse lobortis elit a libero lacinia, -- 0019
13 quis iaculis nulla pharetra. -- 0021
```

Вывод: были приобретены базовые навыки работы с файловой системой в Java.