

Министерство образования Республики Беларусь
Учреждение образования
“Брестский государственный технический университет”
Кафедра ИИТ

Отчёт
По лабораторной работе №2
По дисциплине СПП

Выполнил

Студент группы ПО-3
3-го курса
Куликович И. Т.

Проверил

Крощенко А. А.

Лабораторная работа №2

ВАРИАНТ 13

Задание 1. Напишите программу, которая ищет в тексте похожие слова (слова, которые содержат более 50% подряд идущих букв, совпадающих с соответствующими буквами слова-эталона) и выводит такие слова на экран в порядке “слово-эталон”: “первое похожее слово” “второе похожее слово” и т.д.

Задание 2. Утилита tail выводит несколько (по умолчанию 10) последних строк из файла.

- Формат использования: tail [-n] file
- Ключ -n <количество строк> (или просто <количество строк>) позволяет изменить количество выводимых строк.
- Пример использования:
tail -n 20 app.log
tail 20 app.log
- Выводит 20 последних строк из файла app.log.
- Для решения задачи подойдет класс java.io.RandomAccessFile, реализующий произвольный доступ к файлу (чтение и запись с любой позиции в файле).

Код программы

```
package live.ilyusha.spp2;

import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.*;
import java.io.File;
import java.io.IOException;
import java.util.function.Supplier;
import java.util.stream.Stream;

public class Main {

    public static void main(String[] args) throws IOException {
        System.out.printf("task1 = %s\n", Arrays.toString(task1("text.txt",
"test")));

        System.out.println("task2");

        try {
            Supplier<Stream<String>> lines;
            long skip;
            if (args.length == 1) {
                lines = linesSupplierFactory(Paths.get(args[0]));
                skip = lines.get().count();
            } else if (args.length == 2) {
                lines = linesSupplierFactory(Paths.get(args[1]));
                skip = Long.parseLong(args[0]);
            } else if (args.length == 3 && "-n".equals(args[0])) {
                lines = linesSupplierFactory(Paths.get(args[2]));
                skip = Long.parseLong(args[1]);
            } else if (args.length == 3 && "-n".equals(args[1])) {
                lines = linesSupplierFactory(Paths.get(args[0]));
                skip = Long.parseLong(args[2]);
            } else {
                throw new Exception("Invalid arguments");
            }
            lines.get().skip(lines.get().count() -
skip).forEach(System.out::println);
        }
```

```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static Supplier<Stream<String>> linesSupplierFactory(Path path)
throws Exception {
    return () -> {
        try {
            return Files.lines(path);
        } catch (IOException e) {
            return Stream.empty();
        }
    };
}

    private static String[] task1(String resource, String baseWord) throws
IOException {
        String text = new Scanner(new File("resources/" +
resource)).useDelimiter("\\Z").next();

        LinkedList<String> words = new
LinkedList<String>(Arrays.asList(text.toString().split("[^A-Za-zA-Яa-
я0-9]")));

        words.removeAll(Collections.singleton(""));

        ArrayList<String> result = new ArrayList<>();

        for (String word : words) {
            int lineLength = Math.min(word.length() / 2, baseWord.length());
            if (baseWord.substring(0,
lineLength).equals(word.toLowerCase().substring(0, lineLength)) &&
word.length() > 2) {
                result.add(word);
            }
        }

        return result.toArray(new String[0]);
    }
}

```

Спецификация ввода

```

>java Main [[-n] количество строк] <файл>
>java Main <файл> [[-n] количество строк]

```

Пример

```

>java Main -n 10 resources/text.txt

```

Спецификация вывода

```

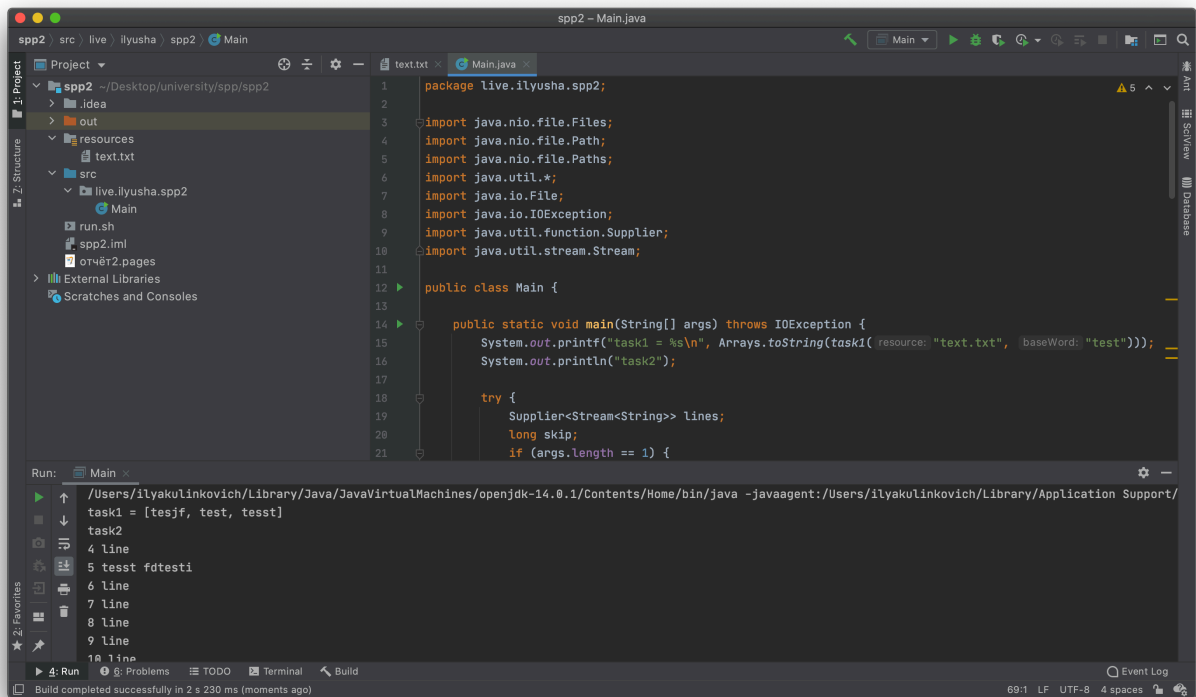
task1 = <набор слов найденный в соответствии с алгоритмом>
task2
<строка 1 найденная в соответствии с алгоритмом>
<строка 2 найденная в соответствии с алгоритмом>
...
<строка N найденная в соответствии с алгоритмом>

```

Пример

```
task1 = [tesjff, test, tesst]
task2
4 line
5 tesst fdtesti
6 line
7 line
8 line
9 line
10 line
11 line
12 line
13 line
```

Рисунки с результатами работы программы



Вывод

В данной лабораторной работе я приобрел базовые навыки работы с файловой системой в Java.