

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Отчёт
по лабораторной работе №2

Выполнила:
студентка группы ПО-9
Матюшик Е.П.

Проверил:
Крощенко А. А.


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

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.

Входные данные:

 spp2_1.txt – Блокнот

Файл Правка Формат Вид Справка
абрикосы
стол
абрикосовый
абдула
абрикосовое

Введите слово-эталон: абрикос

Выходные данные:

Похожие слова для слова "абрикос":
1. абрикосы
2. абрикосовый
3. абрикосовое

Код программы:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class Task1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        String text = readTextFromFile("spp2_1.txt"); // Добавляем чтение текста из файла

        System.out.print("Введите слово-эталон: ");
        String wordToMatch = scanner.nextLine();

        findAndPrintSimilarWords(text, wordToMatch);

        scanner.close();
    }

    // Метод для чтения текста из файла
```

```

public static String readTextFromFile(String fileName) {
    StringBuilder sb = new StringBuilder();
    try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {
        String line;
        while ((line = reader.readLine()) != null) {
            sb.append(line).append("\n");
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return sb.toString();
}

// Метод для поиска и вывода похожих слов
public static void findAndPrintSimilarWords(String text, String wordToMatch) {
    // Разбиваем текст на слова
    String[] words = text.split("[^а-яА-Я]+");

    System.out.println("Похожие слова для слова \"" + wordToMatch + "\"");

    int count = 0;
    // Проверяем каждое слово из текста
    for (String word : words) {
        // Если слово похоже на слово-эталон, выводим его
        if (isSimilar(wordToMatch, word)) {
            count++;
            System.out.println(count + ". " + word);
        }
    }
    // Если не найдено похожих слов, выводим сообщение
    if (count == 0) {
        System.out.println("Похожих слов не найдено.");
    }
}

// Метод для проверки, является ли слово похожим на слово-эталон
public static boolean isSimilar(String wordToMatch, String similarWord) {
    int matchCount = 0;
    // Сравниваем символы слов поочередно
    for (int i = 0; i < wordToMatch.length(); i++) {
        char charToMatch = wordToMatch.charAt(i);
        char similarChar = similarWord.charAt(i);

        // Если символы совпадают или совпадают после приведения к одному регистру,
        // увеличиваем счетчик

```

```

        if (charToMatch == similarChar || Character.toLowerCase(charToMatch) ==
Character.toLowerCase(similarChar)) {
            matchCount++;
        }
    }
    // Проверяем, достигает ли количество совпадений порога в 50%
    return (double) matchCount / wordToMatch.length() > 0.5;
}
}

```

Вариант 4

Задание 2.

Написать консольную утилиту, обрабатывающую ввод пользователя и дополнительные ключи. Проект упаковать в jar-файл, написать bat-файл для запуска. 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

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.

```

Выходные данные:

```

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

```

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> 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 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 file");
                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("-1")
        .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');
                break;
        }
        lineNumber += step;
    }
    outputStream.println(line);
}
}
}

```

Вывод: в результате выполнения лабораторной были приобретены практические навыки работы с файловой системой в Java.