МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Специальность ПО-5(о)

Выполнил

Д. С. Бриштен,

студент группы ПО-5

Проверил

А. А. Крощенко

ст. преп. каф. ИИТ

Брест 2021

Цель работы: Приобрести базовые навыки работы с файловой системой в Java. Вариант 4

Задание 1. Напишите программу, выполняющую чтение текстовых данных из файла и их последующую обработку. Программа выводит слова, располагая их в порядке убывания частоты их появления. Перед каждым словом должно быть число его появлений.

Выполнение:

Код программы

import java.util.Collections;

import java.util.Scanner;

import java.util.concurrent.\*;

import java.io.\*;

public class Task1{

    public static void main(String args[]) throws FileNotFoundException{

        Scanner in = new Scanner(new File("file.txt"));

        String s = "";

        while (in.hasNextLine()) {

            s += in.nextLine() + " ";

        }

        String[] words = s.split(" ");

        ConcurrentMap<String, Integer> m = new ConcurrentHashMap<>();

        for(String word: words){

            m.compute(word, (k, v) -> v == null ? 1 : v + 1);

        }

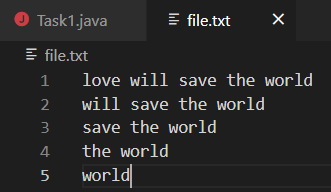
        m.entrySet().stream().sorted(Collections.reverseOrder((n1,n2)->n1.getValue().compareTo(n2.getValue()))).forEach(n->System.out.println(n));

        in.close();

    }

}

Файл для чтения file.txt



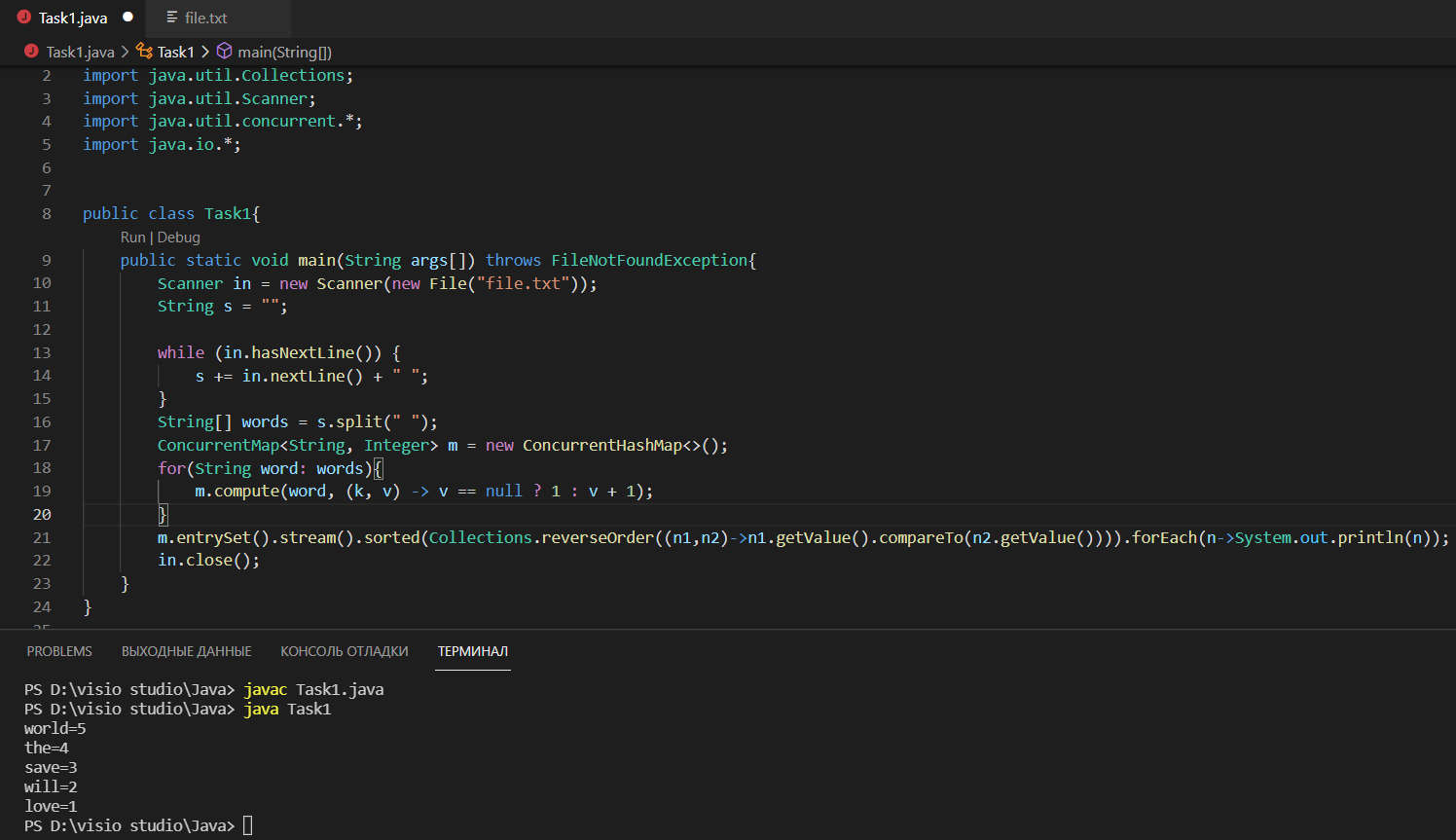
Спецификация вывода

1 итерация forEach: n = <1-й элемент карты>; 2-я: n = <2-й элемент карты>…

Пример:

n1 = ‘world=5’, n2 = ‘the=4’, n3 = ‘save=3’, n4 = ‘will=2’, n5 = ‘love=1’

Рисунок с результатами работы программы



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 Задает шаг увеличения номеров строк  
• -l 1/0 Задает флаг нумерации пустых строк  
• -n ФОРМАТ Использовать заданный формат для номеров строк.  
ln – номер выравнен 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.

Выполнение:

Код программы

import java.io.\*;

import java.util.\*;

import java.util.stream.Collectors;

public class Core{

    public static void main(String[] args){

        Scanner in = new Scanner(System.in);

        String str = in.nextLine();

        String regex = "\\s\*(nl)\\s+(-i)\\s+([0-9]+)\\s+(-l)\\s+([01])\\s+(-n)\\s+(ln|rn|rz)(\\s+((-|[a-z]+.txt)+\\s\*([a-z]+.txt)\*)\*)\*\\s\*";

        Boolean result = str.matches(regex);

        while(!result){

            System.out.println("\nInput template: nl [-i] [-l] [-n] input\_file [output\_file]\n" +

            "Example: nl -i 2 -l 0 -n ln in.txt\n\n|  1 - Continue  |  2 - Exit   |  3 - Help    |\n");

            byte lever = in.nextByte();

            if(lever == 2){

                in.close();

                return;

            }

            else if(lever == 3){

                System.out.println(

                "-i NUMBER Specifies the row number increment step\n" +

                "-l [1/0] Specifies the numbering flag for empty rows\n" +

                "-n FORMAT Use the specified format for line numbers.\n" +

                "ln - number is aligned to the left with no initial zeros\n" +

                "rn - number is aligned to the right with no initial zeros\n" +

                "rz - number is aligned to the right with initial zeros\n"

                );

            }

            in.nextLine();

            System.out.print("Enter: ");

            str = in.nextLine();

            result = str.matches(regex);

        }

        String[] commands = str.split(" ");

        int shag = Integer.parseInt(commands[2]);

        byte flag = Byte.parseByte(commands[4]);

        String alignment = commands[6];

        int len\_com = commands.length;

        List<String> sentences = new ArrayList<>();

        if(len\_com == 7){

            sentences = Input\_Console();

            Output\_Console(sentences, shag, flag, alignment);

        }

        else if(len\_com == 8){

            String fileName = commands[7];

            if(fileName.equals("-")){

                sentences = Input\_Console();

                Output\_Console(sentences, shag, flag, alignment);

            }

            else{

                sentences = Input\_File(fileName, shag, flag, alignment);

                Output\_Console(sentences, shag, flag, alignment);

            }

        }

        else if(len\_com == 9){

            String fileName = commands[7];

            String OutFile = commands[8];

            if(fileName.equals("-")){

                sentences = Input\_Console();

                Output\_File(OutFile, sentences, shag, flag, alignment);

            }

            else{

                sentences = Input\_File(fileName, shag, flag, alignment);

                Output\_File(OutFile, sentences, shag, flag, alignment);

            }

        }

        in.close();

    }

    public static List<String> Input\_Console() {

        Scanner input = new Scanner(System.in);

        System.out.println("Enter text (character of a new string - [%n]):");

        String text = input.nextLine();

        List<String> sentences = Arrays.stream(text.split("%n"))

        .map(String::trim) // Обрезаем строки в потоке (stream)

        .collect(Collectors.toList()); // Собираем их в список

        input.close();

        return sentences;

    }

    public static List<String> Input\_File(String fileName, int shag, Byte flag, String alignment) {

        List<String> sentences = new ArrayList<>();

        try(Scanner file = new Scanner(new File(fileName)))

        {

            while (file.hasNextLine()) {

                sentences.add(file.nextLine());

            }

            file.close();

        }

        catch(IOException ex){

            System.out.println(ex.getMessage());

        }

        return sentences.stream().map(String::trim).collect(Collectors.toList());

    }

    public static void Output\_Console(List<String> sentences, int shag, Byte flag, String alignment) {

        int num = 1;

        if(alignment.equals("ln")){

            for (String str : sentences) {

                if (flag == 0 && str.isEmpty()){ // && - для более быстрой проверки, так как

                    System.out.printf("%-2s %s %n", ' ', str); // & - проверяет оба условия

                    continue;

                }

                System.out.printf("%-2d %s %n", num, str);

                num += shag;

            }

        }

        else if(alignment.equals("rn")){

            for (String str : sentences) {

                if (flag == 0 && str.isEmpty()){

                    System.out.printf("%15s %s %n", ' ', str);

                    continue;

                }

                System.out.printf("%15d %s %n", num, str);

                num += shag;

            }

        }

        else if(alignment.equals("rz")){

            for (String str : sentences) {

                if (flag == 0 && str.isEmpty()){

                    System.out.printf("%015d %s %n", flag, str);

                    continue;

                }

                System.out.printf("%015d %s %n", num, str);

                num += shag;

            }

        }

    }

    public static void Output\_File(String OutFile, List<String> sentences,

    int shag, Byte flag, String alignment) {

        try(PrintStream writer = new PrintStream(new File(OutFile))){

                int num = 1;

                if(alignment.equals("ln")){

                    for (String str : sentences) {

                        if (flag == 0 && str.isEmpty()){

                            writer.printf("%-2s %s %n", ' ', str);

                            continue;

                        }

                        writer.printf("%-2d %s %n", num, str);

                        num += shag;

                    }

                }

                else if(alignment.equals("rn")){

                    for (String str : sentences) {

                        if (flag == 0 && str.isEmpty()){

                            writer.printf("%15s %s %n", ' ', str);

                            continue;

                        }

                        writer.printf("%15d %s %n", num, str);

                        num += shag;

                    }

                }

                else if(alignment.equals("rz")){

                    for (String str : sentences) {

                        if (flag == 0 && str.isEmpty()){

                            writer.printf("%015d %s %n", flag, str);

                            continue;

                        }

                        writer.printf("%015d %s %n", num, str);

                        num += shag;

                    }

                }

                writer.close();

            }

        catch(IOException ex){

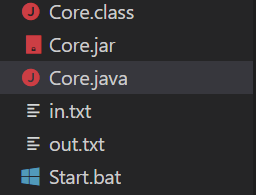
            System.out.println(ex.getMessage());

        }

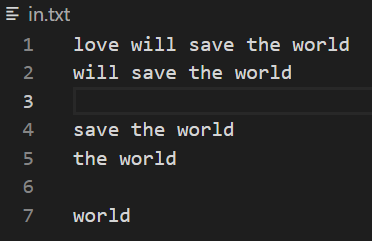
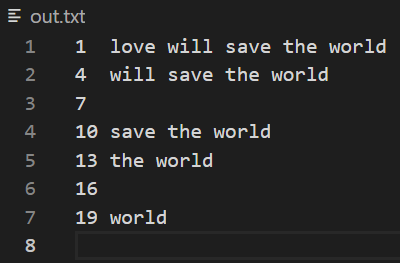
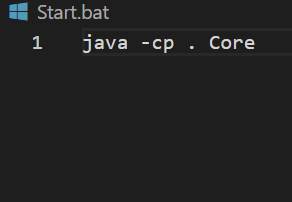
    }

}

Имеем следующие файлы в каталоге



in.txt out.txt Start.bat

Файл чтения Файл записи Файл запуска

Спецификация ввода:

Запуск программы производится через файл Start.bat в командной строке

str = <утилита и дополнительные ключи>

lever = <выбор действия>

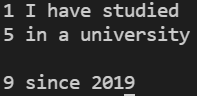
text = <текст для обработки>

sentences (файловый ввод) = <1 элемент списка (1 строка файла)> <2 (2 строка)>...

Спецификация вывода:

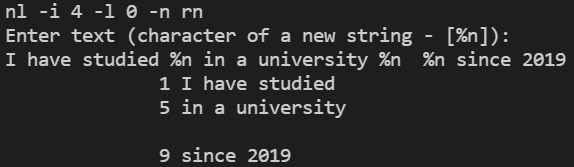
<1 элемент списка> <2> <3> …

Пример:

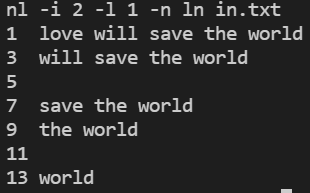


Рисунки с результатами работы программы

Используя только консоль

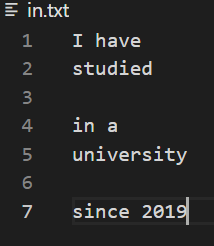
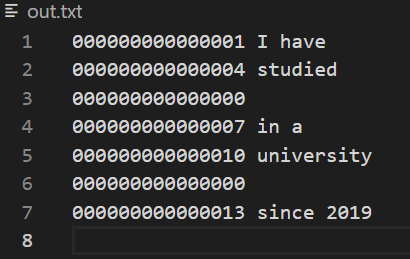


Используя файловый ввод и консольный вывод



Используя файловый ввод-вывод



Вывод: научился выполнять простейшие операции с файлами.