

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ
КАФЕДРА ИИТ

Отчет по лабораторной работе №2

Выполнил:
Студент 4 курса
Группы АС-50
Куц Д.А.
Проверил:
Крощенко А.А.

Брест 2020

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

Задание 1. Напишите программу, выполняющую чтение текстовых данных из файла и их последующую обработку:

9) Напишите программу, которая использует генерацию случайных чисел для создания предложений. Программа должна использовать 4 массива строк, называемые noun (существительные), adjective (прилагательные), verb (глаголы) и preposition (предлоги). Указанные массивы должны считываться из файла. Программа должна создавать предложение, случайно выбирая слова из каждого массива в следующем порядке: noun, verb, preposition, adjective, noun. Слова должны быть разделены пробелами. При выводе окончательного предложения, оно должно начинаться с заглавной буквы и заканчиваться точкой. Программа должна генерировать 20 таких предложений.

Код:

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

public class task1 {
    public static String[] ReadWrite(String str, BufferedReader bufReader,
    FileReader fileReader, ArrayList<String> arList, String[] string) throws
    IOException {
        bufReader = new BufferedReader(fileReader);
        arList = new ArrayList<>();

        while((str = bufReader.readLine()) != null ) {
            if (!str.isEmpty()) arList.add(str);
        }

        string = arList.toArray(new String[0]);
        return string;
    }

    public static void main(String[] args) throws IOException {
        FileReader nounReader = new FileReader("task1_files/noun.txt"),
        verbReader = new FileReader("task1_files/verb.txt"),
        prepReader = new FileReader("task1_files/preposition.txt"),
        adjReader = new FileReader("task1_files/adjective.txt");

        BufferedReader bufNounReader = new BufferedReader(nounReader),
        bufVerbReader = new BufferedReader(verbReader),
        bufPrepReader = new BufferedReader(prepReader),
        bufAdjReader = new BufferedReader(adjReader);

        ArrayList<String> nounList = new ArrayList<>(),
        verbList = new ArrayList<>(),
        prepList = new ArrayList<>(),
        adjList = new ArrayList<>();

        String nounStr = "", verbStr = "", prepStr = "", adjStr = "";

        String[] nouns = new String[0],
        verbs = new String[0],
        prepositions = new String[0],
        adjectives = new String[0];
```

```

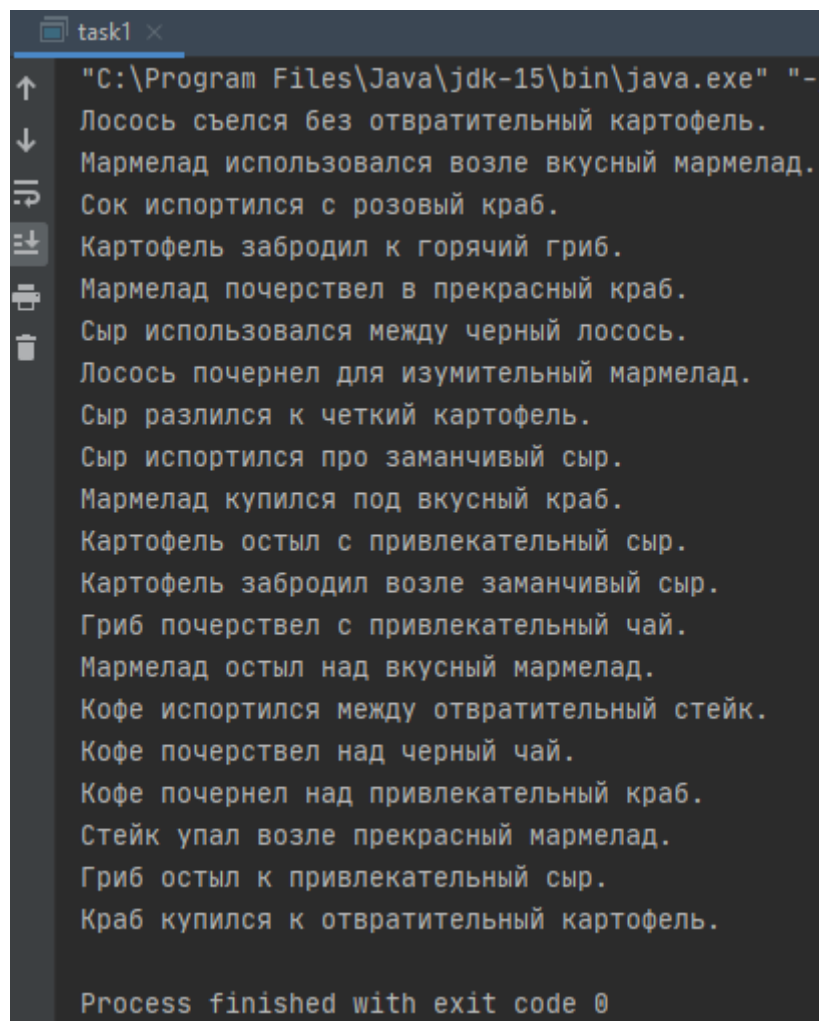
        String [] noun = ReadWrite(nounStr, bufNounReader, nounReader,
nounList, nouns),
        verb = ReadWrite(verbStr, bufVerbReader, verbReader,
verbList, verbs),
        preposition = ReadWrite(prepreStr, bufPrepreReader, prepreReader,
prepList, prepositions),
        adjective = ReadWrite(adjStr, bufAdjReader, adjReader,
adjList, adjectives);

        for (int i=0; i<20; i++) {
            int num1 = (int) (Math.random()*10),
            num2 = (int) (Math.random()*10),
            num3 = (int) (Math.random()*10),
            num4 = (int) (Math.random()*10),
            num5 = (int) (10 + Math.random()*10);

            System.out.println(noun[num1] + " "
+ verb[num2]+ " "
+ preposition[num3] + " "
+ adjective[num4]+ " "
+ noun[num5]+ ".");
        }
    }
}

```

Результат:



```

"C:\Program Files\Java\jdk-15\bin\java.exe" "-
Лосось съелся без отвратительный картофель.
Мармелад использовался возле вкусный мармелад.
Сок испортился с розовый краб.
Картофель забродил к горячий гриб.
Мармелад почерствел в прекрасный краб.
Сыр использовался между черный лосось.
Лосось почернел для изумительный мармелад.
Сыр разлился к четкий картофель.
Сыр испортился про заманчивый сыр.
Мармелад купился под вкусный краб.
Картофель остыл с привлекательный сыр.
Картофель забродил возле заманчивый сыр.
Гриб почерствел с привлекательный чай.
Мармелад остыл над вкусный мармелад.
Кофе испортился между отвратительный стейк.
Кофе почерствел над черный чай.
Кофе почернел над привлекательный краб.
Стейк упал возле прекрасный мармелад.
Гриб остыл к привлекательный сыр.
Краб купился к отвратительный картофель.

Process finished with exit code 0

```

Задание 2. Написать консольную утилиту, обрабатывающую ввод пользователя и дополнительные ключи. Проект упаковать в jar-файл, написать bat-файл для запуска.

9) Утилита `join` объединяет строки двух упорядоченных текстовых файлов на основе наличия общего поля. По своему функционалу схоже с оператором `JOIN`, используемого в языке `SQL` для реляционных баз данных, но оперирует с текстовыми файлами. Команда `join` принимает на входе два текстовых файла и некоторое число аргументов. Если не передаются никакие аргументы командной строки, то данная команда ищет пары строк в двух файлах, обладающие совпадающим первым полем (последовательностью символов, отличных от пробела), и выводит строку, состоящую из первого поля и содержимого обоих строк. Ключами `-1` или `-2` задаются номера сравниваемых полей для первого и второго файла, соответственно. Если в качестве одного из файлов указано `-` (но не обоих сразу!), то в этом случае вместо файла считывается стандартный ввод.

Формат использования:

`join [-1 номер_поля] [-2 номер_поля] файл1 файл2 [файл3]`

Параметры:

- `-1 field_num` Задаёт номер поля в строке для первого файла, по которому будет выполняться соединение.
- `-2 field_num` Задаёт номер поля в строке для второго файла, по которому будет выполняться соединение.

Аргументы:

- `файл1`, `файл2` – входные файлы
- `файл3` – выходной файл, куда записывается результат работы программы.

Примеры использования:

Пусть задан файл `1.txt` со следующим содержимым:

1 abc

2 lmn

3 pqr

и файл `2.txt` со следующим содержимым:

1 abc

3 lmn

9 orq

Тогда, выполнение команды **`join 1.txt 2.txt`** даст следующий результат:

1 abc abc

3 pqr lmn

Поскольку в обоих файлах есть строки, чьё первое поле совпадает (1, 3), выполнение команды **join -1 2 -2 2 1.txt 2.txt** даст результат

abc 1 1

lmn 2 3

Поскольку теперь сравнение выполняется по 2-му полю для первого и второго файла соответственно.

Код:

```
import java.io.FileWriter;
import java.io.IOException;
import java.nio.charset.Charset;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;

public class task2 {

    public static void main(String[] args) throws IOException {
        List<String> firstFile =
Files.readAllLines(Paths.get("task2_files/f1.txt"),
Charset.defaultCharset()),
        secondFile =
Files.readAllLines(Paths.get("task2_files/f2.txt"),
Charset.defaultCharset());

        FileWriter writer = new FileWriter("task2_files/f3.txt");

        int firstField = Integer.parseInt(args[1]), secondField =
Integer.parseInt(args[2]),
        length = firstFile.size(), length1 = 0, length2 = 0;

        String field1, field2;

        int index1, index2;

        char[] st1_11 = new char[10], st1_12 = new char[10], st1_21 = new
char[10], st1_22 = new char[10],
        st2_11 = new char[10], st2_12 = new char[10], st2_21 =
new char[10], st2_22 = new char[10],
        st3_11 = new char[10], st3_12 = new char[10], st3_21 =
new char[10], st3_22 = new char[10],
        st4_11 = new char[10], st4_12 = new char[10], st4_21 =
new char[10], st4_22 = new char[10];

        for (String str : args) System.out.print(str + " ");
        System.out.println("");

        if (firstField == 1 && secondField == 1) {
            for (int i = 0; i < length; i++) {
                field1 = firstFile.get(i);
                field2 = secondFile.get(i);
                length1 = field1.length();
                length2 = field2.length();
                index1 = field1.indexOf(' ');
```

```

        index2 = field2.indexOf(' ');
        field1.getChars(0, index1, st1_11, 0);
        field2.getChars(0, index2, st1_21, 0);
        field1.getChars(index1, length1, st1_12, 0);
        field2.getChars(index2, length2, st1_22, 0);
        if (Arrays.equals(st1_11, st1_21)) {
            System.out.print(st1_11);
            System.out.print(" ");
            System.out.print(st1_12);
            System.out.print(" ");
            System.out.print(st1_22);
            writer.write(st1_11);
            writer.write(st1_12);
            writer.write(st1_22);
            writer.write("\n");
        }
    }

    if (firstField == 1 && secondField == 2) {
        for (int i = 0; i < length; i++) {
            field1 = firstFile.get(i);
            field2 = secondFile.get(i);
            length1 = field1.length();
            length2 = field2.length();
            index1 = field1.indexOf(' ');
            index2 = field2.indexOf(' ');
            field1.getChars(0, index1, st2_11, 0);
            field2.getChars(0, index2, st2_21, 0);
            field1.getChars(index1 + 1, length1, st2_12, 0);
            field2.getChars(index2 + 1, length2, st2_22, 0);
            if (Arrays.equals(st2_11, st2_22)) {
                System.out.print(st2_11);
                System.out.print(" ");
                System.out.print(st2_12);
                System.out.print(" ");
                System.out.print(st2_21);
                writer.write(st2_11);
                writer.write(st2_12);
                writer.write(st2_22);
                writer.write("\n");
            }
        }
    }

    if (firstField == 2 && secondField == 1) {
        for (int i = 0; i < length; i++) {
            field1 = firstFile.get(i);
            field2 = secondFile.get(i);
            length1 = field1.length();
            length2 = field2.length();
            index1 = field1.indexOf(' ');
            index2 = field2.indexOf(' ');
            field1.getChars(0, index1, st3_11, 0);
            field2.getChars(0, index2, st3_21, 0);
            field1.getChars(index1 + 1, length1, st3_12, 0);
            field2.getChars(index2, length2, st3_22, 0);
            if (Arrays.equals(st3_12, st3_21)) {
                System.out.print(st3_12);
                System.out.print(" ");
                System.out.print(st3_11);
                System.out.print(" ");
                System.out.print(st3_22);
                writer.write(st3_12);
            }
        }
    }
}

```

```

        writer.write(st3_11);
        writer.write(st3_22);
        writer.write("\n");
    }
}

if (firstField == 2 && secondField == 2) {
    for (int i = 0; i < length; i++) {
        field1 = firstFile.get(i);
        field2 = secondFile.get(i);
        length1 = field1.length();
        length2 = field2.length();
        index1 = field1.indexOf(' ');
        index2 = field2.indexOf(' ');
        field1.getChars(0, index1, st4_11, 0);
        field2.getChars(0, index2, st4_21, 0);
        field1.getChars(index1 + 1, length1, st4_12, 0);
        field2.getChars(index2 + 1, length2, st4_22, 0);
        if (Arrays.equals(st4_12, st4_22)) {
            System.out.print(st4_12);
            System.out.print(" ");
            System.out.print(st4_11);
            System.out.print(" ");
            System.out.print(st4_21);
            writer.write(st4_12);
            writer.write(st4_11);
            writer.write(st4_21);
            writer.write("\n");
        }
    }
}

writer.close();

Scanner scan = new Scanner(System.in);
String s = scan.nextLine();
System.out.println(s);
}
}

```

Результат:

The image shows four terminal windows, each titled 'task2', displaying the output of a file join operation. The first window shows 'join 1 1' with output '1 hiii damn'. The second window shows 'join 1 2' with output 'heyy nice 8'. The third window shows 'join 2 1' with output 'yooo 5 9999'. The fourth window shows 'join 2 2' with output 'what 3 4'. Below these is a screenshot of a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The prompt shows the command 'java -jar ssp_lab2.jar join 2 2' being executed, with the output 'join 2 2' and 'what 3 4' displayed on separate lines.

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