

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

Лабораторная работа №2
По дисциплине «Современные платформы программирования»

Выполнила:
Студентка 3 курса
Группы ПО-3
Дмитрук В.Г.
Проверил:
Крощенко А.А.

Брест 2020 г.

Вариант 9

Цель работы:

Приобрести практические навыки работы с файловой системой в Java.

Ход работы

Задание 1:

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.

Текст программы:

```
public static void task1() {
    String nouns =
fileReader("/Users/valeriadmitruk/Desktop/SPP/src/com/company/nouns.txt");
    String adjectives =
fileReader("/Users/valeriadmitruk/Desktop/SPP/src/com/company/adjectives.txt"
);
    String verbs =
fileReader("/Users/valeriadmitruk/Desktop/SPP/src/com/company/verbs.txt");
    String prepositions =
fileReader("/Users/valeriadmitruk/Desktop/SPP/src/com/company/prepositions.tx
t");

    String[] nounsArray = nouns.split(" ");
    String[] adjectivesArray = adjectives.split(" ");
    String[] verbsArray = verbs.split(" ");
    String[] prepositionsArray = prepositions.split(" ");

    String result = "";
    for(int i=0; i<20; i++) {
        String noun = nounsArray[(int) (Math.random()*nounsArray.length)];
        result += noun.substring(0, 1).toUpperCase() + noun.substring(1) + "
";

        String verb = verbsArray[(int) (Math.random()*verbsArray.length)];
        result += verb + " ";
        String preposition = prepositionsArray[(int)
(Math.random()*prepositionsArray.length)];
        result += preposition + " ";
        String adjective = adjectivesArray[(int)
(Math.random()*adjectivesArray.length)];
```

```

        result += adjective + " ";
        String nounLast = nounsArray[(int)
(Math.random()*nounsArray.length)];
        result += nounLast + ".";
        System.out.println(result);
        result = "";
    }
}

```

Рисунок с результатом работы программы:

```

Небо кричать около красивый дожди.
Солнце ехать перед уродливый малышка.
Тучи плакать на уродливый облака.
Облака кричать на злой солнце.
Небо следовать под красивый солнце.
Море делать в добрый тучи.
Дьявол делать под дьявольский дьявол.
Тучи делать перед уродливый небо.
Дьявол следовать под уродливый дожди.
Малышка кричать под красивый облака.
Небо делать на дьявольский море.
Дожди следовать под уродливый небо.
Малышка ехать на дьявольский дожди.
Дьявол делать на добрый малышка.
Дьявол плакать перед уродливый дьявол.
Тучи плакать под злой солнце.
Дьявол следовать под красивый море.
Солнце кричать около злой море.
Малышка делать на красивый небо.
Небо следовать около дьявольский малышка.

```

Задание 2:

- 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.

-

Текст программы:

```

public static String fileReader(String filename) {
    String text = "";
    try (FileReader reader = new FileReader(filename))
    {
        int c;
        while((c=reader.read())!=-1){
            text += (char)c;
        }
    }
}

```

```

    }
    catch(IOException ex){
        System.out.println(ex.getMessage());
    }
    return text;
}

public static void fileWriter(String filename, String str){
    try {
        FileWriter myWriter = new FileWriter(filename);
        myWriter.write(str);
        myWriter.close();
        System.out.println("Successfully wrote to the file.");
    } catch (IOException e) {
        System.out.println("An error occurred.");
        e.printStackTrace();
    }
}

public static Boolean compareStrings(String[] str1, String[] str2, int val1,
int val2) {
    return str1[val1].equals(str2[val2]);
}

public static void join(String values,
                        String filepathOne,
                        String filepathTwo,
                        String filepathThree
) {
    int val1=0;
    int val2=0;
    String[] val = values.split(" ");
    if (val.length>0) {
        if (val.length == 2) {
            if (val[0].equals("-1")) {
                val1 = Integer.parseInt(val[1]) - 1;
            } else if (val[0].equals("-2")){
                val2 = Integer.parseInt(val[1]) - 1;
            }
        } else if (val.length == 4) {
            if (val[0].equals("-1")) {
                val1 = Integer.parseInt(val[1]) - 1;
            }
            if (val[2].equals("-2")) {
                val2 = Integer.parseInt(val[3]) - 1;
            }
        }
    }

    if(val1>2 || val1<0) { val1 = 0; }
    if(val2>2 || val2<0) { val2 = 0; }
}

int oppositeOne = 0;
int oppositeTwo = 0;
if (val1 == 0) { oppositeOne = 1; } else { oppositeOne = 0; }
if (val2 == 0) { oppositeTwo = 1; } else { oppositeTwo = 0; }
String answer = "";
//считываем 2 файла
String one = fileReader(filepathOne);
String two = fileReader(filepathTwo);
//переводим в 2 массива строк
String[] str1 = one.split("\n");
String[] str2 = two.split("\n");
// в цикле разбиваем на сплит и сравниваем
for(int i=0; i<str1.length; i++) {

```

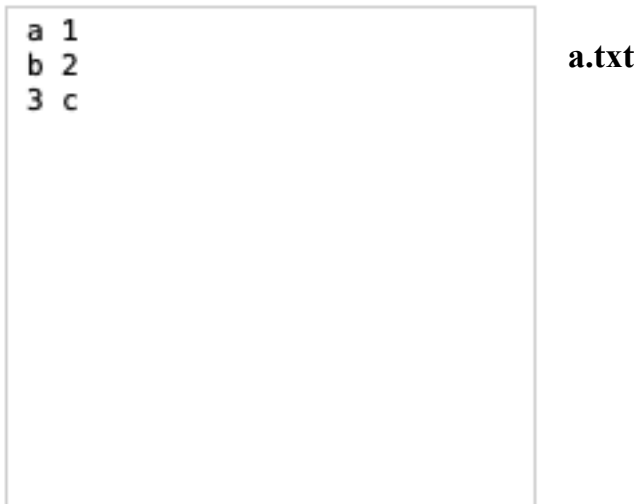
```

        for(int j=0; j<str2.length; j++) {
            String[] str11 = str1[i].split(" ");
            String[] str22 = str2[j].split(" ");
            if(compareStrings(str11, str22, val1, val2) == true) {
                answer += str11[val1];
                answer += " ";
                answer += str11[oppositeOne];
                answer += " ";
                answer += str22[oppositeTwo];
                answer += "\n";
            }
        }
    }
    fileWriter(filepathThree, answer);
}

public static void main(String[] args) {
    join("",
"/Users/valeriadmitruk/Desktop/SPP/out/production/SPP/com/company/b.txt",
"/Users/valeriadmitruk/Desktop/SPP/out/production/SPP/com/company/a.txt",
    "answer.txt");
}

```

Рисунок с результатом работы программы:



a 1
b 2
3 c

a.txt

```
1 a
2 b
3 c
```

b.txt

```
3 c c
```

answer.txt

Выводы:

Приобрела практические навыки работы с файловой системой Java.