# Министерство образования Республики Беларусь Учреждение образования

«Брестский государственный технический университет» Кафедра ИИТ

# Лабораторная работа №2

Выполнила: Студент 3 курса Группы ПО-5 Соколов А.А. Проверил:

Крощенко А.А.

Брест 2021

**Вариант - 9**

**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:

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 программы:**

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.Scanner;

**public** **class** Next {

**public** **static** **void** main(String[] args){

String way\_noun= "One.txt"; //существительные

String way\_adjective= "Two.txt"; //прилагательные

String way\_verb= "Three.txt"; //глаголы

String way\_preposition= "Four.txt"; //предлоги

**int** a,b,c,d;

a=*number*(way\_noun);

System.***out***.println("Количество существительных в файле - " + a );

String[] noun=**new** String[a];

*filling*(way\_noun,noun);

b=*number*(way\_adjective);

System.***out***.println("Количество прилагательных в файле - " + b );

String[] adjective=**new** String[b];

*filling*(way\_adjective,adjective);

c=*number*(way\_verb);

System.***out***.println("Количество глаголов в файле - " + c );

String[] verb=**new** String[c];

*filling*(way\_verb,verb);

d=*number*(way\_preposition);

System.***out***.println("Количество предлогов в файле - " + d );

String[] preposition=**new** String[d];

*filling*(way\_preposition,preposition);

**for**(**int** i=0;i<20;i++) { //Составление предложений

**int** aa =(**int**)(Math.*random*() \* a );

**int** aaa =(**int**)(Math.*random*() \* a );

**int** bb =(**int**)(Math.*random*() \* b );

**int** cc =(**int**)(Math.*random*() \* c );

**int** dd =(**int**)(Math.*random*() \* d );

String name;

name = noun[aa].substring(0,1).toUpperCase() + noun[aa].substring(1).toLowerCase(); //Первый элемент с заглавной буквы

System.***out***.println("Предложение "+ (i+1) +": " + name+ " " + verb[cc]+ " " + preposition[dd]+ " " + adjective[bb]+ " " + noun[aaa]+ "." );

}

}

**public** **static** **void** filling(String directory, String word[]) { //Заполнение массива данными из файла

String line = **null**;

**int** i = 0,k=0;

**try** {

File file=**new** File(directory);

Scanner scanner= **new** Scanner(file, "UTF8");

**while**(scanner.hasNextLine()) {

line=scanner.nextLine();

String[] mass =line.split(" ");

k=mass.length;

**for**(**int** t=0;t<k;t++) {

word[i]=mass[t];

i++;

}

}

scanner.close();

} **catch**(IOException e){

System.***out***.println("Error: " +e);

}

}

**public** **static** **int** number(String directory) { //Определение количества слов в файле

String line = **null**;

**int** i = 0;

**try** {

File file=**new** File(directory);

Scanner scanner= **new** Scanner(file, "UTF8");

**while**(scanner.hasNextLine()) {

line=scanner.nextLine();

String[] mass =line.split(" ");

i+=mass.length;

}

scanner.close();

} **catch**(IOException e){

System.***out***.println("Error: " +e);

}

**return** i;

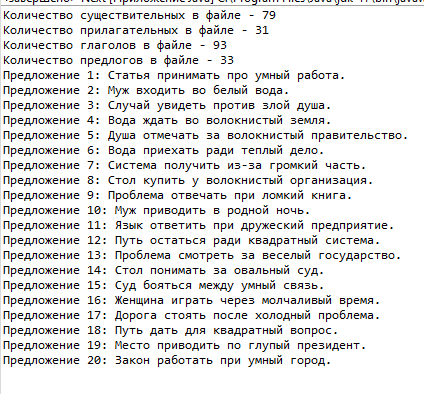
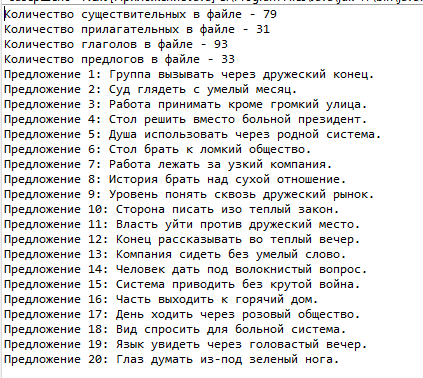
}

## }

## Существительные

## Прилагательные

## Глаголы Предлоги Результат:

## 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 fiedl\_num Задает номер поля в строке для первого файла, по которому будет выполняться соединение.

• - 2 field\_num Задает 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 opq

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 программы:**

**package** kkk;

**import** java.io.File;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.util.Scanner;

**public** **class** Join {

**public** **static** **void** main(String[] args){

**int** a=1,b=1,join=0;

**int** lenght=args.length-1;

System.***out***.println("Начинается проверка переданных значений!");

**if** (lenght<2) { //Проверка если переданных значений меньше чем нужно, то завершает программу

System.***out***.println("Переданных значений не достаточно");

**return**;

}

**else** {

System.***out***.println("Переданных значений достаточно");

}

**for** (**int** i=0;i<args.length;i++) {

**if**(args[i].equals("Join")) { //Убедимся что утилита Join именно вызывается

join=1;

}

**if**(args[i].equals("-1")) { //Проверка на ключ для первого файла

a=Integer.*parseInt*(args[i+1]);

}

**if**(args[i].equals("-2")){ //Проверка на ключ для второго файла

b=Integer.*parseInt*(args[i+1]);

}

}

**if** (join==1) {

System.***out***.println("Утилита Join выполняется с файлами:");

}

**else** {

System.***out***.println("Неизвестная утилита");

**return**;

}

**if**(args[lenght-2].equals("-")) {

}

String way\_first;

String way\_two;

String way\_result;

way\_first=args[lenght-2];

System.***out***.println("Путь до первого файла - " + way\_first);

way\_two=args[lenght-1];

System.***out***.println("Путь до второго файла - " + way\_two);

way\_result=args[lenght];

System.***out***.println("Путь до файла с результатами - " + way\_result);

**int** number\_first, number\_two;

number\_first=*number*(way\_first,a);

number\_two=*number*(way\_two,b);

String[] content\_first=**new** String[number\_first];

String[] join\_content=**new** String[number\_first\*number\_two];

**int**[] Coincidence=**new** **int**[number\_first\*number\_two];

*first*(way\_first,a,content\_first); //Запись в массив всех слов в колонке "а"

**int** number\_Coincidence=*Two*(way\_two, b, content\_first,Coincidence,join\_content); //Чтение второго файла и поиск схожести с первым файлом колонок

*first\_last*(way\_first,a,Coincidence, join\_content, number\_Coincidence);

*result*(join\_content,number\_Coincidence, way\_result);

System.***out***.println("Вывод результата на консоль: ");

**for**(**int** i=0;i<number\_Coincidence;i++) {

System.***out***.println(join\_content[i]);

}

}

**public** **static** **void** first\_last(String directory\_first,**int** a, **int** Coincidence[], String [] join, **int** number\_Coincidence) {

**int** t=0; //t- это номер предложения [0,1,2..n]

**try** {

File file\_first=**new** File(directory\_first);

Scanner scanner\_first= **new** Scanner(file\_first, "UTF8");

**while**(scanner\_first.hasNextLine()) {

String line=scanner\_first.nextLine();

String[] mass\_first =line.split(" ");

**int** i=mass\_first.length;

**for**(**int** j=0;j< number\_Coincidence;j++) { //Записываем в for для того чтобы найти повторы в совпадениях

**if**(t==Coincidence[j]) { //Предложение "t" сравнивается с числом, в котором было найдено совпадение

**for**(**int** g=0;g<i;g++) {

**if**(g==(a-1)) { //Исключаем то слово которое сравнивали и переходим к следующему

g++;

}

**if**(g!=i) {

join[j]= join[j]+" "+mass\_first[g]; //Запись в массив предложения из первого файла

}

}

}

}

t++; //Переход к следующему предложению

}

scanner\_first.close();

} **catch**(IOException e){

System.***out***.println("Error: " +e);

}

}

**public** **static** **int** number(String directory,**int** a) { //Определение количества слов в ряде "a"

String line = **null**;

**int** i,number=0;

**try** {

File file=**new** File(directory);

Scanner scanner= **new** Scanner(file, "UTF8");

**while**(scanner.hasNextLine()) {

line=scanner.nextLine();

String[] mass =line.split(" ");

i=mass.length;

**if**(a<=i) {

number++;

}

}

scanner.close();

} **catch**(IOException e){

System.***out***.println("Error: " +e);

}

**return** number;

}

**public** **static** **void** first(String directory\_first, **int** a,String[] content) {//Запись в массив всех слов в колонке "а"

**int** t=0;

**try** {

File file\_first=**new** File(directory\_first);

Scanner scanner\_first= **new** Scanner(file\_first, "UTF8");

**while**(scanner\_first.hasNextLine()) {

String line=scanner\_first.nextLine();

String[] mass\_first =line.split(" ");

**int** i=mass\_first.length;

**if**(a<=i) {

content[t]=mass\_first[a-1];

t++;

}

}

scanner\_first.close();

} **catch**(IOException e){

System.***out***.println("Error: " +e);

}

}

**public** **static** **int** Two(String directory\_two, **int** b,String[] content\_first,**int**[] Coincidence, String[] join) {

**int** t=0;

String content\_two;

**try** {

File file\_first=**new** File(directory\_two);

Scanner scanner\_two= **new** Scanner(file\_first, "UTF8");

**while**(scanner\_two.hasNextLine()) {

String line=scanner\_two.nextLine();

String[] mass\_two =line.split(" ");

**int** i=mass\_two.length;

**if**(b<=i) {

content\_two=mass\_two[b-1];

**for**(**int** a=0;a<content\_first.length;a++) {

**if**(content\_first[a].equals(content\_two)) {

Coincidence[t]=a;

join[t]=content\_first[a];

**for**(**int** z=0;z<i;z++) {

**if**(z==(b-1)) {

z++;

}

**if**(z!=i) {

join[t]= join[t]+" "+mass\_two[z];

}

}

t++;

}

}

}

}

scanner\_two.close();

} **catch**(IOException e){

System.***out***.println("Error: " +e);

}

**return** t;

}

**public** **static** **void** result(String[] join, **int** number, String way\_result) {

**try** {

File file\_result=**new** File(way\_result);

PrintWriter pw=**new** PrintWriter(file\_result);

**for**(**int** i=0;i<number;i++) {

pw.println(join[i]);

}

pw.close();

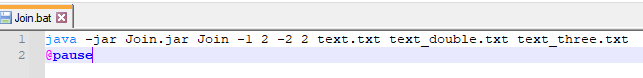
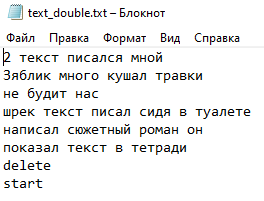
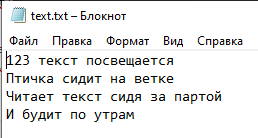
} **catch**(IOException e){

System.***out***.println("Error: " +e);

}

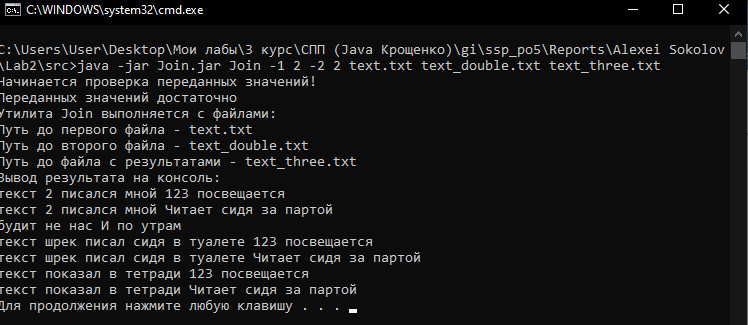
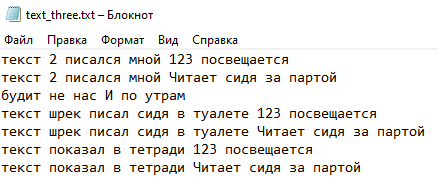
}

}

## 

## Результат:

**Вывод:** приобрёли базовые навыки работы с файловой системой в Java.