

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Специальность ПОЗ

Выполнила Р.И.
Гаврилюк, студентка
группы ПОЗ

Проверил А.А.
Крощенко,
ст. преп. кафедры ИИТ,
«—» ————— 2020 г.

Брест 2020

Вариант 6

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

Задание 1. Напишите программу сравнения двух файлов, которая будет печатать первую строку и позицию символа, где они различаются. В противном случае должно выводиться сообщение об эквивалентности содержимого файлов.

Код программы

```
import java.io.*;

class Lab_2 {

    public void task1()
    {
        try {
            File file1=new File("D:\\User\\Documents\\3 курс\\спп\\лаба 2\\file 1.txt");
            FileReader fr1 = new FileReader(file1);
            BufferedReader reader1 = new BufferedReader(fr1);
            String line1 = reader1.readLine();

            File file2=new File("D:\\User\\Documents\\3 курс\\спп\\лаба 2\\file 2.txt");
            FileReader fr2 = new FileReader(file2);
            BufferedReader reader2 = new BufferedReader(fr2);
            String line2 = reader2.readLine();

            boolean fileEquals = true;
            int ind = 0;
            while (line1 != null || line2 != null) {
                if(!line1.equals(line2)){
                    fileEquals = false;
                    System.out.println("Строки различающиеся:");
                    System.out.println(line1);
                    System.out.println(line2);

                    int N = line1.toCharArray().length;
                    if(line1.toCharArray().length <= line2.toCharArray().length){
                        N = line2.toCharArray().length;
                    }
                    for(int i = 0; i < N; i++){
                        ind++;
                        if(line1.toCharArray()[i] != line2.toCharArray()[i]){
                            System.out.println("Символы различающиеся:");
                            System.out.println(line1.toCharArray()[i]);
                            System.out.println(line2.toCharArray()[i]);
                            System.out.println("Индекс символа:");
                            System.out.println(ind);
                            break;
                        }
                    }
                }
            }
        }
    }
}
```

```

        else{
            ind += line1.toCharArray().length;
        }
        line1 = reader1.readLine();
        line2 = reader2.readLine();

        if(line1 != null && line2 == null){
            fileEquals = false;
            System.out.println("Строки различающиеся:");
            System.out.println(line1);
            System.out.println(line2);
            System.out.println("Символы различающиеся:");
            System.out.println(line1.toCharArray()[0]);
            System.out.println("Индекс символа:");
            System.out.println(ind + 1);
            break;
        }
        else if(line1 == null && line2 != null){
            fileEquals = false;
            System.out.println("Строки различающиеся:");
            System.out.println(line1);
            System.out.println(line2);
            System.out.println("Символы различающиеся:");
            System.out.println(line2.toCharArray()[0]);
            System.out.println("Индекс символа:");
            System.out.println(ind + 1);
            break;
        }
    }
    reader1.close();
    reader2.close();
    if(fileEquals){
        System.out.println("Файлы идентичны");
    }

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    Lab_2 lab = new Lab_2();
    lab.task1();
}
}

```

Рисунки с результатами работы программы

```
Строки различающиеся:
null
kkk
Символы различающиеся:
k
Индекс символа:
28
```

file 1.txt – Блокнот	file 2.txt – Блокнот
Файл Правка Формат	Файл Правка Формат
hello	hello
world	world
it	it
is	is
file	file
1	1
good buy	good buy
	kkk

file 1.txt – Блокнот	file 2.txt – Блокнот
Файл Правка Формат	Файл Правка Формат
hello	hello
world	world
it	it
is	is
file	file
1	1
good buy	good buy

Файлы идентичны

Задание 2. Утилита `split` копирует и разбивает файл на отдельные файлы заданной длины. В качестве аргументов ей надо указать имя исходного файла и префикс имен выходных файлов. Если файл не задан или задан как `-`, программа читает стандартный ввод.

По умолчанию размер части разбиения равен 10 строк, а префикс равен `x`. Имена выходных файлов будут состояться из этого префикса и двух дополнительных букв `aa`, `ab`, `ac` и т. д. (без пробелов и точек между префиксом и буквами). Если префикс имен файлов не задан, то по умолчанию используется `x`, так что выходные файлы будут называться `хаа`, `хаb` и т. д.

Формат использования: `split [-b | -l] [-d] [входной_файл [префикс_выходных_файлов]]` где ключи имеют следующее значение:

- `-b` , `--bytes=num` Записывать в каждый выходной файл заданное число `num` байт. При задании числа байт можно использовать суффиксы: `b` означает байты, `k` – `1kb` , `m` – `1Mb`.
- `-l` , `--lines=num` Записывать в каждый выходной файл `num` строк.
- `-d` , `--numericssuffixes` Использовать числовые, а не алфавитные суффиксы, начинающиеся с `00`. Суффиксы файлов будут иметь вид: `00`, `01`, `02` и т. д.

Код программы

```
import java.io.*;
import java.util.Scanner;
import java.util.LinkedList;
import java.util.Queue;
import java.util.Arrays;
```

```

import java.math.*;
import javax.management.Query;

class Lab_2 {
    //размещение с повторением для генерации имени файла
    public Object[][] getVariations(Object[] source, int variationLength) {
        int srcLength = source.length;
        int permutations = (int) Math.pow(srcLength, variationLength);

        Object[][] table = new Object[permutations][variationLength];

        for (int i = 0; i < variationLength; i++) {
            int t2 = (int) Math.pow(srcLength, i);
            for (int p1 = 0; p1 < permutations; p1++) {
                for (int a1 = 0; a1 < srcLength; a1++) {
                    for (int p2 = 0; p2 < t2; p2++) {
                        table[p1][i] = source[a1];
                        p1++;
                    }
                }
            }
        }

        return table;
    }

    public String setFileName(String prefix, String nameElements, int numFileName){
        String fileName = prefix;
        int m = nameElements.length(); //количество символов в массиве возможных элементов для названия
        int n; //количество необходимых элементов в размещении
        for(int i = 1; ; ++i){
            if(numFileName < Math.pow(m, i)){
                n = i;
                break;
            }
        }

        int numVariations = 0;
        for(int i = 0; i < n - 1; ++i){
            numVariations += Math.pow(m, i);
        }

        String[] elements = new String[nameElements.length()];
        for(int i = 0; i < nameElements.length(); ++i){
            elements[i] = Character.toString(nameElements.charAt(i));
        }

        Object[][] variations = this.getVariations(elements, n);
        int i = numFileName - numVariations - 1;
        for(int j = 0; j < variations[i].length; ++j){
            fileName += variations[i][j];
        }

        return fileName;
    }
}

```

```

}

public void task2(String[] args){
    if(args[0].equals("split")){
        int numLines = 10;
        int numBytes = -1;
        String preffix = "x";
        String nameElements = "abcdefghijklmnopqrstuvwxy";
        String fileName = "";

        //parsing
        for(int i = 1; i < args.length; i++) {
            if(args[i].contains("-b") || args[i].contains("--bytes=")){
                String size;
                if(args[i].contains("-b") ){
                    size = args[i].replace("-b=", "");
                }
                else {
                    size = args[i].replace("--bytes=", "");
                }

                if(size.contains("kb") ){
                    numBytes = (int)Math.pow(10, 3) * Integer.parseInt(size.replace(
"kb", ""));
                }
                else if(size.contains("Mb") ){
                    numBytes = (int)Math.pow(10, 6) * Integer.parseInt(size.replace(
"Mb", ""));
                }
                else if(size.contains("b") ){
                    numBytes = Integer.parseInt(size.replace("b", ""));
                }
                else{
                    numBytes = Integer.parseInt(size);
                }
            }
            else if(args[i].contains("-l") || args[i].contains("--lines")){
                if(args[i].contains("-l") ){
                    numLines = Integer.parseInt(args[i].replace("-l=", ""));
                }
                else {
                    numLines = Integer.parseInt(args[i].replace("--lines=", ""));
                }
            }
            else if(args[i].contains("-d") || args[i].contains("--
numeric suffixes")){
                nameElements = "0123456789";
            }
            else{
                if(fileName == ""){
                    fileName = args[i];
                }
                else{
                    preffix = args[i];
                }
            }
        }
    }
}

```

```

    }
}

//files
Queue<String> arr = new LinkedList<>();
if(fileName.equals("")){
    fileName = "x.txt";

    Scanner in = new Scanner(System.in);
    String str = in.nextLine();
    while(!str.contains("#")){
        arr.offer(str);
        str = in.nextLine();
    }
    in.close();
}

try{
    File file = new File(fileName);
    file.createNewFile();

    if(file.exists()){
        //если чтение из консоли
        if(!arr.isEmpty()){
            FileWriter fw = new FileWriter(file);
            String it;
            while ((it = arr.poll()) != null) {
                fw.write(it);
                fw.write("\n");
                fw.flush();
            }
            fw.close();
        }

        FileReader fr = new FileReader(file);
        BufferedReader reader = new BufferedReader(fr);

        if(numBytes == -1){
            //работа со строками
            String line = null;
            int numFile = 0;
            do{
                arr.clear();
                for(int i = 0; i < numLines; ++i){
                    line = reader.readLine();
                    arr.offer(line);
                }
                ++numFile;

                String fileOddName = this.setFileName(prefix, nameElements,
numFile);

                File oddFile = new File(fileOddName);
                oddFile.createNewFile();

```

```

        FileWriter fw = new FileWriter(oddFile);
        String it;
        while ((it = arr.poll()) != null) {
            fw.write(it);
            fw.write("\n");
            fw.flush();
        }
        fw.close();
    }while(line != null);
}
else{
    //побайтовое чтение и запись
    int numFile = 0;
    int offset = 0;

    do{
        char[] buffer = new char[numBytes];

        for(int i = 0; i < numFile; ++i){
            reader.read(buffer, 0, numBytes);
        }
        offset += numBytes;
        ++numFile;

        String fileOddName = this.setFileName(prefix, nameElements,
numFile);

        File oddFile = new File(fileOddName);
        oddFile.createNewFile();

        FileWriter fw = new FileWriter(oddFile);
        fw.write(buffer);
        fw.close();
    }while(offset < file.length());
}

    reader.close();
    fr.close();
}
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

}

public static void main(String[] args) {

    Lab_2 lab = new Lab_2();
    //lab.task1();
    lab.task2(args);
}
}

```


Рисунки с результатами работы программы

```
'Lab_2' 'split' '-d' '-b=10'
```

```
dhdhdhdhdhd  
efiejifoje  
wdsd  
#
```

x.txt

x0

x1

x2

x.txt – Блокнот

Файл Правка Формат

```
dhdhdhdhdhd  
efiejifoje  
wdsd
```

x0 – Блокнот

Файл Правка Формат

```
dhdhdhdhdhd
```

x1 – Блокнот

Файл Правка Формат Вид

```
d  
efiejifoje
```

x2 – Блокнот

Файл Правка Формат Вид

```
e  
wdsd
```

```
'Lab_2' 'split' 'a.txt' 'a' '--numericssuffixes' '-l=1'
```

a.txt – Блокнот

Файл Правка Формат Вид Справка

```
int numVariations = 0;  
    for(int i = 0; i < n - 1; ++i){  
        numVariations += Math.pow(m, i);  
    }
```

a0 – Блокнот

Файл Правка Формат Вид Справка

```
int numVariations = 0;
```

a1 – Блокнот

Файл Правка Формат Вид Справка

```
    for(int i = 0; i < n - 1; ++i){
```

a2 – Блокнот

Файл Правка Формат Вид Справка

```
        numVariations += Math.pow(m, i);
```

a3 – Блокнот

Файл Правка Формат Вид Справка

```
    }
```

Вывод: приобрела базовые навыки работы с файловой системой в Java.