

Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №2
По дисциплине: «СПП»
Тема: «Файловая система компьютера и ЯП Java»
Вариант №6

Выполнил:
Студент 3 курса
Группы ПО-7
Комиссаров А.Е.
Проверила:
Демидович А.Г.

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

Задание 1: 6) 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.

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

6) 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`, `хаб` 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.

Ход работы:

Файл `task1.java`

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

public class Main{
    public static void main(String[] args) throws IOException{

        String path_to_files = "C:\\Users\\Student\\IdeaProjects\\LaboratornayaSPP\\src\\";
        BufferedReader file1 = new BufferedReader(new FileReader(path_to_files + "file1.txt"));
        BufferedReader file2 = new BufferedReader(new FileReader(path_to_files + "file2.txt"));

        String line1 = file1.readLine();
        String line2 = file2.readLine();
        boolean equal = true;
        int line = 1;
        int character = 1;
        while (line1 != null || line2 != null)
        {
            if(line1 == null || line2 == null)
            {
                equal = false;
                break;
            }
        }
    }
}
```

```

        else if(! line1.equalsIgnoreCase(line2))
        {
            equal = false;
            break;
        }
        line1 = file1.readLine();
        line2 = file2.readLine();
        line++;
    }

    if(equal){
        System.out.println("No differences found.");
    }else{
        for (int i = 0; i < line1.length(); i++){
            if (line1.charAt(i) != line2.charAt(i)){
                character = i + 1; break;
            }
        }
        System.out.println("Files are different on line : " + line + " , character : " + character);
        System.out.println("File1 : " + line1 + "\nFile2 : " + line2);

        for (int i = 0; i < character + 7; i++){ System.out.print(" "); }
        System.out.print("^");

    }
    file1.close();
    file2.close();
}
}

```

Содержимое файла file1.txt

A bunch of random words that have absolutely no meaning or sense at all. They're literally exactly identical. NO DIFFERENCES!!

Содержимое файла file2.txt

A bunch of random words that have absolutely no meaning or sense at all. Theyre literally exactly identical. NO DIFFERENCES!!

Результат работы программы:

```

Files are different on line : 3 , character : 19
File1 : sense at all. They're literally
File2 : sense at all. Theyre literally
                        ^
Process finished with exit code 0

```

Рис. 1 – Результат работы программы (1)

Файл task2.java

```
import java.io.*;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.NoSuchFileException;
import java.nio.file.Paths;

public class Main{
    public static void main(String[] args) throws IOException {

        boolean arg_bytes, arg_lines, arg_numeric;
        String filename, prefix, file, newfilename;
        int num;

        arg_bytes = arg_lines = arg_numeric = false;
        num = 0;
        filename = prefix = "";

        if (args.length == 0){
            System.out.println("Arguments parse fail : no arguments (use --help to learn about
usable parameters).");
            System.exit(1); }

        for (int i = 0; i<args.length; i++){
            ////help////////////////////////////////////
            if (args[i].equalsIgnoreCase("-h") | args[i].equalsIgnoreCase("--help")){
                System.out.println("Usage : lab2 [-b OR -l <num>] [-d] [<filename.txt> [prefix]]");
                System.out.println("\nParameters : ");
                System.out.println("\t-b (--bytes) - divide the original file into multiple files that
consist of <num> bytes from the original file.");
                System.out.println("\t-l (--lines) - split the file into multiple files that contain a
specified amount of lines from the original.");
                System.out.println("\t-d (--numericprefixes) - the files will use a numeric prefix
(00_ex.txt, 01_ex.txt, 02_ex.txt and so on).");
                System.out.println("\nExample : lab2 -b 60 file1.txt -d");
                System.exit(0);
            }
            ////bytes////////////////////////////////////
            if (args[i].equalsIgnoreCase("-b") | args[i].equalsIgnoreCase("--bytes")) {
                arg_bytes = true;
                if (i+1 < args.length) {
                    try {
                        num = Integer.parseInt(args[i+1]);
                    } catch (NumberFormatException nfe){
                        System.out.println("Arguments parse fail : first argument after -b (--bytes)
arg is not an integer.");
                        System.exit(1);
                    }
                } else {
                    System.out.println("Arguments parse fail : no arguments after -b (--bytes).");
                    System.exit(1);
                }
            }
        }
    }
}
```

```

////////lines////////////////////////////////////
if (args[i].equalsIgnoreCase("-l") | args[i].equalsIgnoreCase("--lines")) {
    arg_lines = true;
    if (i+1 < args.length){
        try {
            num = Integer.parseInt(args[i+1]);
        } catch (NumberFormatException nfe){
            System.out.println("Arguments parse fail : first argument after -l (--lines)
arg is not an integer.");
            System.exit(1);
        }
    } else {
        System.out.println("Arguments parse fail : no arguments after -l (--lines).");
        System.exit(1);
    }
}
////////numeric suffixes////////////////////////////////////
if (args[i].equalsIgnoreCase("-d") | args[i].equalsIgnoreCase("--numericprefixes")) {
    arg_numeric = true;
}
////////filename////////////////////////////////////
if (!args[i].startsWith("-") & args[i].endsWith(".txt")){
    filename = args[i];
}
////////prefix////////////////////////////////////
if (!args[i].startsWith("-") & !args[i].endsWith(".txt") & !args[i].equals("" + num)){
    prefix = args[i];
}
////////////////////////////////////

}

    if (!arg_numeric & prefix.isEmpty()){
        System.out.println("Prefix not specified, using \"x\".");
        prefix = "x"; }

    if (filename.isEmpty()){
        System.out.println("Filename not specified, using default.txt instead.");
        filename = "default.txt"; }

    file = readFile(filename);

    if (file.isEmpty()){
        System.out.print("File is empty.");
        System.exit(1); }

    if(arg_bytes | arg_lines){
        if(num > 0){

if (!arg_numeric){
    newfilename = prefix + "aa";
} else { newfilename = prefix + "00"; }

if(arg_bytes){
    byte[] bytes_arr = file.getBytes(StandardCharsets.UTF_8);
    for (int i = 0; i < bytes_arr.length; i++){
        try{
            FileWriter fstream = new FileWriter(newfilename + ".txt",true);

```

```

        BufferedWriter out = new BufferedWriter(fstream);
        out.write(bytes_arr[i]);
        out.close();
    }catch (Exception e){
        System.err.println("Error while writing to file: " +
            e.getMessage());
    }
    if (i % num == 0){
        newfilename = getNextPrefix(prefix, newfilename, arg_numeric);
    }
}
} else {
    String[] file_arr = file.split("\n");
    for (int i = 0; i < file_arr.length; i++){
        try{
            FileWriter fstream = new FileWriter(newfilename + ".txt",true);
            BufferedWriter out = new BufferedWriter(fstream);
            out.write(file_arr[i]);
            out.close();
        }catch (Exception e){
            System.err.println("Error while writing to file: " +
                e.getMessage());
        }
        if (i % num == 0){
            newfilename = getNextPrefix(prefix, newfilename, arg_numeric);
        }
    }
}

    } else {
        System.out.println("Arguments parse fail : received -l (--lines) or -b (--
bytes) with negative amount.");
        System.exit(1);
    }
}

}

public static String getNextPrefix(String prefix, String previous, boolean arg_numeric){
    String result;
    String old_prefix = previous.substring(prefix.length());
    int buff_int;

    if(!arg_numeric){
        if (old_prefix.charAt(1) < 'z'){
            buff_int = old_prefix.charAt(1);
            buff_int ++;
            result = prefix + old_prefix.charAt(0) + (char)buff_int;
        } else if (old_prefix.charAt(0) < 'z'){
            buff_int = old_prefix.charAt(0);
            buff_int ++;
            result = prefix + (char)buff_int + "a";
        } else {
            result = prefix + "aa";
        }//////////
    } else { //numeric prefix
        buff_int = Integer.parseInt(old_prefix);
        buff_int++;
        result = prefix + buff_int;
    }
}

```

```

    }
    return result;
}
public static String readFile(String filename) throws IOException {
    String file = "";
    try {
        file = Files.readString(Paths.get(filename), StandardCharsets.UTF_8);
        System.out.println("\nFile contents : \n" + file);
    } catch (NoSuchFileException nsfe){
        System.out.println("File opening fail : No such file " + filename);
        System.exit(1);
    }
    return file;
}
}

```

Содержимое файла default.txt

What is Lorem Ipsum?

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Why do we use it?

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

Упаковка проекта в jar-файл:

```

C:\Files\Github repos\Laboratorki\Labi\СПП\Lab2\src\task2\src>javac Main.java
C:\Files\Github repos\Laboratorki\Labi\СПП\Lab2\src\task2\src>jar -cvf Main.jar Main.class
added manifest
adding: Main.class(in = 5664) (out= 2995)(deflated 47%)

```

Рис. 2 – Упаковка проекта

Содержимое файла split.bat:

```

@ECHO OFF
java -cp Main.jar Main %*

```

Параметры, использованные при запуске программы:

```

C:\Files\Github repos\Laboratorki\Labi\СПП\Lab2\src\task2\src>split -l 3 myfile_

```

Результат работы программы:

myfile_aa.txt	02.11.2022 21:48	Текстовый документ	1 КБ
myfile_ab.txt	02.11.2022 21:48	Текстовый документ	1 КБ
myfile_ac.txt	02.11.2022 21:48	Текстовый документ	1 КБ
myfile_ad.txt	02.11.2022 21:48	Текстовый документ	1 КБ
myfile_ae.txt	02.11.2022 21:48	Текстовый документ	1 КБ

Рис. 3 – Текстовые документы, созданные программой

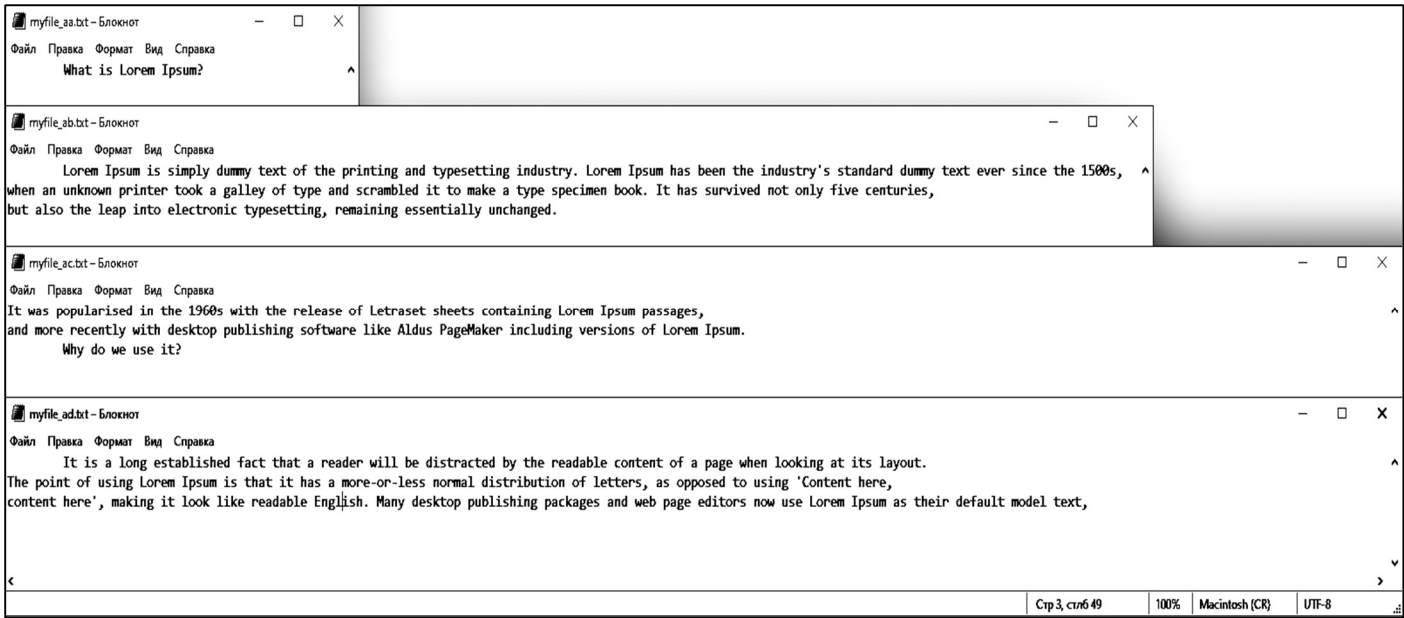


Рис. 4 – Результат работы программы (2)

Вывод: я приобрёл базовые навыки работы с файловой системой в Java.