Министерство образования Республики Беларусь
Учреждение образования
“Брестский государственный технический университет”
Кафедра ИИТ

Лабораторная работа №2
по дисциплине «ССП»

Выполнил:
Студент 4 курса
Факультета ЭИС
Группы АС – 50
Мендель Д.А.
Проверил:
Крощенко А.А.

Брест 2020

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

Вариант 6

Задание 1

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

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 составляться 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 xaa, xab 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 , --numericsuffixes Использовать числовые, а не алфавитные суффиксы, начинающиеся с 00. Суффиксы файлов будут иметь вид: 00, 01, 02 и т. д.

Код программы:

```
package com.company;

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

public class Main {

   public static void main(String[] args) {
      readFile("f1.txt", "f2.txt");
   }

   private static void readFile(String path1, String path2) {
      try {
         File file1 = new File(path1);
         File file2 = new File(path2);
         FileReader fr1 = new FileReader(file1);
         FileReader fr2 = new FileReader(file2);
         BufferedReader br1 = new BufferedReader(fr1);
         BufferedReader br2 = new BufferedReader(fr2);
```

```
            String line1 = br1.readLine();
            String line2 = br2.readLine();
            int count = 1;
            while ((line1 != null) && (line2 != null)) {
                if (!line1.equals(line2)) {
                    char[] array1 = line1.toCharArray();
                    char[] array2 = line2.toCharArray();
                    for (int i = 0; i < array1.length; i++) {
                        if (array1[i] != array2[i]) {
                            System.out.println("line number: " + count + "\nchar position: " + (i + 1));
                            return;
                        }
                    }
                }
                line1 = br1.readLine();
                line2 = br2.readLine();
                count++;
            }
            System.out.println("files are equal");
        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}
```

```
package com.company;

import java.io.*;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Main {

    private static boolean dIsExist = false;

    private static String readFile(String path) throws IOException {
        File file = new File(path);
        FileReader fr = new FileReader(file);
        BufferedReader bf = new BufferedReader(fr);
        StringBuilder stringBuilder = new StringBuilder();
        String line = bf.readLine();
        while (line != null) {
            stringBuilder.append(line).append(System.lineSeparator());
            line = bf.readLine();
        }
        return stringBuilder.toString();
```

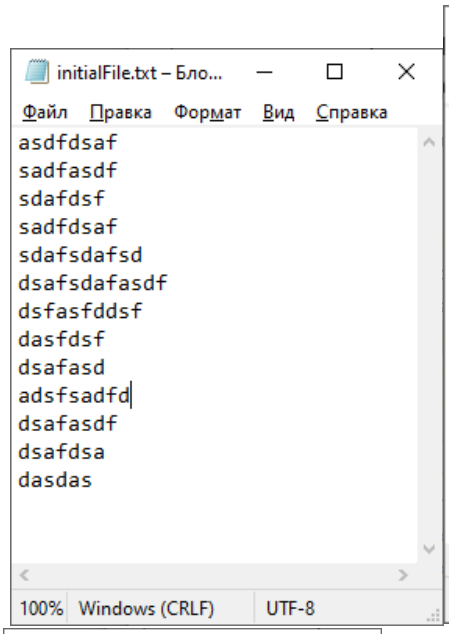
```
}

private static void writeFile(String[] content, String prefix) throws IOException {
   char c1 = 'a';
   char c2 = 'a';
   int i1 = 0;
   int i2 = 0;
   FileWriter fileWriter;
   BufferedWriter bufferedWriter;
   for (String string : content) {
      if (dIsExist)
         fileWriter = new FileWriter(prefix + i1 + i2 + ".txt");
      else
         fileWriter = new FileWriter(prefix + c1 + c2 + ".txt");
      bufferedWriter = new BufferedWriter(fileWriter);
      bufferedWriter.append(string);
      bufferedWriter.flush();
      if (i2 == 9) {
         i1++;
         i2 = 0;
      } else i2++;
      if (c2 == 'z') {
         c2 = 'a';
         c1++;
      } else
         c2++;

   }
}

private static String[] initializeStringsForB(String fileContent, int length, int number, int factor) {
   String[] strings;
   int count = 0;
   int begin = 0;
   int end = 0;
   if (length < number * factor) {
      strings = new String[]{fileContent};
   } else {
      strings = new String[(int) Math.ceil(1.0 * length / number / factor)];
      end = number * factor;
      while (begin < fileContent.length()) {
         strings[count] = fileContent.substring(begin, end);
         begin = end;
         end += number * factor;
         if (end > fileContent.length())
            end = fileContent.length();
         count++;
      }
   }
   return strings;
```

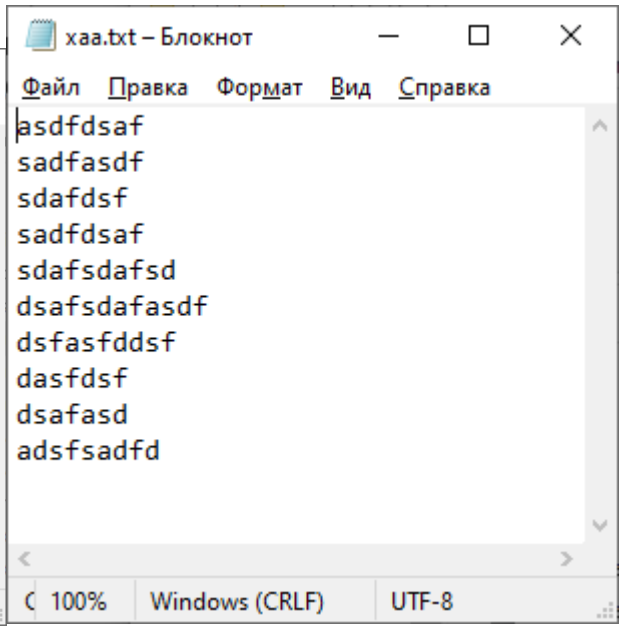
```
}

private static String[] initializeStringsForL(String fileContent, int limit) {
   String[] tempStrings = fileContent.split("\r\n");
   String[] resultStrings = new String[(int) Math.ceil(1.0 * tempStrings.length / limit)];
   int count = 0;
   int i = 0;
   StringBuilder stringBuilder = new StringBuilder();
   for (String string : tempStrings) {
      stringBuilder.append(string).append(System.lineSeparator());
      count++;
      if (count >= limit) {
         count = 0;
         resultStrings[i] = stringBuilder.toString();
         i++;
         stringBuilder = new StringBuilder();
      }
      System.out.println(string);
   }
   if (!stringBuilder.toString().isEmpty())
      resultStrings[i] = stringBuilder.toString();
   return resultStrings;
}

private static String setPrefix(String[] params) {
   if (Pattern.matches("\\w", params[params.length - 1])) {
      return params[params.length - 1];
   }
   return "x";
}

private static String initializeFileContent(String[] params) throws IOException {
   String fileContent;
   if (Pattern.matches("\\w+\\.txt", params[3])) {
      fileContent = readFile(params[3]);
   } else {
      fileContent = readFile(params[4]);
      dIsExist = true;
   }
   return fileContent;
}

private static String initializeFileContentWithoutFlags(String[] params) throws IOException {
   String fileContent;
   if (Pattern.matches("\\w+\\.txt", params[1])) {
      fileContent = readFile(params[1]);
   } else {
      dIsExist = true;
      fileContent = readFile(params[2]);
   }
```

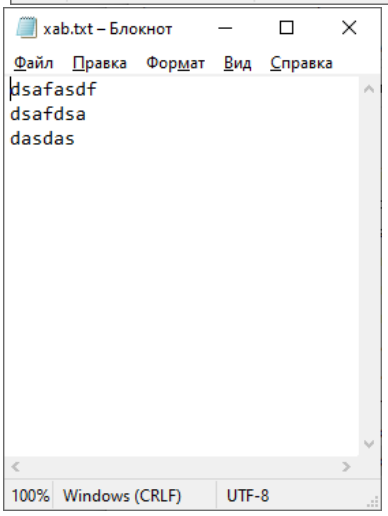
```
    return fileContent;
}

private static String initializeContentWithoutFile() {
    Scanner scanner = new Scanner(System.in);
    String tempString = scanner.nextLine();
    StringBuilder stringBuilder = new StringBuilder();
    while (!tempString.equals("")) {
        stringBuilder.append(tempString).append(System.lineSeparator());
        tempString = scanner.nextLine();
    }
    scanner.close();
    return stringBuilder.toString();
}

public static void main(String[] argss) throws IOException {
    String[] args = new String[1];
    args[0] = "split -l 1 g";
    String[] params = args[0].split(" ");
    String fileContent;

    String prefix;

    if (Arrays.stream(params).noneMatch(param -> Pattern.matches("(\\w+\\.txt)", param))) {
        String content = initializeContentWithoutFile();
        prefix = setPrefix(params);
        if (params[1].equals("-b")) {
            runWithB(params, content, prefix);
        } else if (params[1].equals("-l")) {
            writeFile(initializeStringsForL(content, Integer.parseInt(params[2])), prefix);
        } else
            writeFile(initializeStringsForL(content, 10), prefix);
    } else if (Pattern.matches("split (-b \\d+[bBkKmM]\\s)(-d\\s)?(\\w+\\.txt)(\\s\\w)?", args[0])) {
        prefix = setPrefix(params);
        fileContent = initializeFileContent(params);

        runWithB(params, fileContent, prefix);
    } else if (Pattern.matches("split (-l \\d+\\s)(-d\\s)?(\\w+\\.txt)(\\s\\w)?", args[0])) {
        prefix = setPrefix(params);
        fileContent = initializeFileContent(params);

        writeFile(initializeStringsForL(fileContent, Integer.parseInt(params[2])), prefix);
    } else {
        prefix = setPrefix(params);
        fileContent = initializeFileContentWithoutFlags(params);
        writeFile(initializeStringsForL(fileContent, 10), prefix);
    }
}

private static void runWithB(String[] params, String fileContent, String prefix) throws IOException {
```

```
        String size = params[2].substring(params[2].length() - 1);
        int number = Integer.parseInt(params[2].replace(size, ""));
        String[] strings;


        switch (size.toLowerCase()) {
            case "b":
                int factor = 1;
                strings = initializeStringsForB(fileContent, fileContent.length(), number, factor);
                writeFile(strings, prefix);
                break;

            case "k": {
                factor = 1024;
                strings = initializeStringsForB(fileContent, fileContent.length(), number, factor);
                writeFile(strings, prefix);
                break;
            }
            case "m":
                factor = 1024 * 1024;
                strings = initializeStringsForB(fileContent, fileContent.length(), number, factor);
                writeFile(strings, prefix);
                break;
        }
    }
}
```

Тестовые данные:







Результат :