

Laboratório 17 - Leitura e Escrita de Ficheiros I (28/10/2014)

1.

```
package lab17;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

/**
 * This program read the grades in fileIn and creates a new file, fileOut with only the positive
 * Grades
 * @author FilipaG
 */

public class PositiveGrades {
    public static void copyApproved(String fileIn, String fileOut) throws
        FileNotFoundException, IOException {
        BufferedReader br = new BufferedReader(new FileReader(fileIn));
        //o ficheiro fileIn é lido e armazenado provisoriamente pelo buffer

        PrintWriter pw = new PrintWriter(fileOut);
        //criação do ficheiro fileOut onde será inserida a informação

        String line = null;
        while ((line = br.readLine()) != null) {
            //faz a leitura de cada linha até chegar à linha null (fim do ficheiro)

            int pos = line.lastIndexOf(" ");
            //posição em que começa a leitura da informação

            int grade = Integer.parseInt(line.substring(pos + 1));
            //inteiro encontrado em cada string (linha)

            if (grade >= 10) {
                pw.println(line);
                //imprimir notas apenas se forem positivas (>=10)
            }
        }
        br.close();pw.close(); //fecho das streams de leitura e escrita
    }

    public static void main (String[]args) throws FileNotFoundException, IOException
    {
        String Grades= "C:\\Users\\Filipa\\workspace\\sessão21_10\\src\\lab17\\Grades.txt";
        //localização do ficheiro Grade (input)
        String GradesAprov =
        "C:\\Users\\Filipa\\workspace\\sessão21_10\\src\\lab17\\GradesAprov.txt";
        //localização do ficheiro de output a ser criado
        copyApproved(Grades, GradesAprov);
    }
}
```

GradesAprov.txt

Nelson Gonçalves 14
Bernardo Silva 17
Gonçalo Baptista 14
Ana Moreira 10

Laboratório 17 - Leitura e Escrita de Ficheiros I (28/10/2014)

2.

```
package sessão21_10;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
/**
 * This class reads the file fileIn and creates a new file fileOut with the same text,
 * but with a new word, Jose, wherever occurs the specific word Ze.
 * @author FilipaG
 */

public class IOUtils {
    /**
     * This method reads the fileIn file and copy the file text to fileOut, but
     * substitutes with newWord whenever a specific word WordToFind occur. The method
     * WordToFind tests whether the word is part of a longer word, analyzing the
     * character immediately before the beginning of the word and immediately following
     the
     * end of wordToFind.
     * @param fileIn
     * @param fileOut
     * @param wordToFind
     * @param newWord
     * @return
     */
    public static boolean copyAndReplace(String fileIn, String fileOut, String
wordToFind, String newWord) {

        try (BufferedReader br = new BufferedReader(new FileReader(fileIn));
            PrintWriter pw = new PrintWriter(fileOut)) {
            String line = null;
            String newLine="";
            while ((line = br.readLine()) != null) {

                // verifica se a linha tem a palavra referente à wordToFind
                while(line.contains(wordToFind)) {

//verificar se a string wordTofind não faz parte de nenhuma palavra, ou seja, se o
caracter que a antecede e a sucede são " "

//obter o caracter que está depois da palavra WordToFind
                char afterWordToFind =
line.charAt(line.indexOf(wordToFind)+wordToFind.length());

//índice da posição onde inicia a nova string (na primeira posição após a palavra até
ao fim da linha)
                int beginIndex = line.indexOf(wordToFind)+wordToFind.length();

//índice da posição onde finaliza a string (na primeira posição da palavra wordToFind)
                int endIndex = line.indexOf(wordToFind);

// Se WordToFind é a primeira palavra da linha, verificar se o caracter após o
wordToFind é letra
                if ((line.indexOf(wordToFind)==0)&& (!isALetter(afterWordToFind)))
                {
                    newLine = newWord;
                }

                // Se wordToFind contém o último caracter ficheiro
                else if ((beginIndex==line.length())&&(!isALetter(line.charAt(endIndex-
1))))
                {
                    newLine += newWord;
                }
            }
        }
    }
}
```

Laboratório 17 - Leitura e Escrita de Ficheiros I (28/10/2014)

```
    }

    // beforeWordToFind=line.charAt(endIndex-1))
    else if ((!isALetter(afterWordToFind)) && (!isALetter(line.charAt(endIndex-
1))))
    {
        newLine += line.substring(0,endIndex) + newWord ;
// A linha já obtida, substituir a palavra "wordToFind" pela palavra "newWord" e
adicionar
    }
    else
// Se a palavra "wordToFind" fizer parte de uma palavra maior, não se faz qualquer
substituição
    {
        newLine += line.substring(0,beginIndex);
    }

    line = line.substring(beginIndex); //a linha a analisar passa a
ser apenas da primeira posição após a ocorrência da palavra wordToFind até ao fim da
linha.//é esta nova linha que será novamente analisada para verificar a ocorrência da
palavra wordToFind
    }
    newLine +=line;
    pw.println(newLine); //impressão das linhas de texto no ficheiro
fileOut
    }

    br.close(); // fechar o BufferedReader
    pw.close(); //fechar o PrintWriter

} catch (FileNotFoundException e) // se o ficheiro não for
encontrado
{
    System.out.println("File not found or could not be opened");
    e.printStackTrace();

} catch (IOException e) {
    System.out.println("Error reading from file");
    e.printStackTrace();
}
return true;
}

/**
 * This method verifies if the string wordToFind is not a part of any word, that
is, if the character that
 * precedes or succeeds is not a letter.
 * @param char1
 * @return true if the character is a letter, and false otherwise
 */
public static boolean isALetter(char char1){
    if ((char1 > 96 && char1 < 123 )||(char1 > 64 && char1 < 91))// testar se
está entre a e z ou A e Z
    {
        return true;
    }
    return false;
}

public static void main (String [] args){

    String fileIn =
"C:\\Users\\Filipa\\workspace\\sessão21_10\\src\\lab17\\ZeZambeze.txt";
    String fileOut =
"C:\\Users\\Filipa\\workspace\\sessão21_10\\src\\lab17\\JoseZambeze.txt";
    String wordToFind = "Ze";
```

Laboratório 17 - Leitura e Escrita de Ficheiros I (28/10/2014)

```
String newWord = "Jose";  
  
    copyAndReplace(fileIn, fileOut, wordToFind, newWord);  
}  
}
```

ZeZambeze.txt

O Ze foi fazer um Safari ao Zambeze. Na sua viagem avistou montes de Zebras e Elefantes. Além dos animais, o objectivo do Ze era visitar as tribos locais. Numa delas, ensinaram-no a cozer carne de Javali, da qual o Ze gostou muito. O Ze contou a sua viagem aos amigos, que ficaram tao fascinados, que muitos deles vao faze-la também.

JoseZambeze.txt

O Jose foi fazer um Safari ao Zambeze. Na sua viagem avistou montes de Zebras e Elefantes. Além dos animais, o objectivo do Jose era visitar as tribos locais. Numa delas, ensinaram-no a cozer carne de Javali, da qual o Jose gostou muito. O Jose contou a sua viagem aos amigos, que ficaram tao fascinados, que muitos deles vao faze-la também.