

## TP10

**ASSUNTO – Ficheiros de texto e Estruturas de Dados indexadas**

### Objetivos Específicos:

- Mediante a apresentação dum problema, os alunos deverão ser capazes de o analisar, conceber e descrever o algoritmo estruturado em módulos e utilizar *ficheiros de texto e arrays*.

### Conteúdo da aula

#### Exercício 1

Analise o seguinte programa e diga qual a sua funcionalidade:

```
public class TP10_1 {

    static final String FILE1 = "textFile.txt";
    static final String FILE2 = "textFile2.txt";
    static final int MAX_WORDS = 20;

    public static void main(String[] args) throws FileNotFoundException {
        Scanner in1 = new Scanner(System.in);

        String[] arr = new String[MAX_WORDS];
        String str = in1.next();
        readFromFile(arr, FILE1, str);
        writeToFile(arr, FILE2);
    }

    public static void readFromFile(String[] arr, String file, String str)
        throws FileNotFoundException {

        Scanner in2 = new Scanner(new File(file));
        int words = 0;
        while (in2.hasNext() && words < MAX_WORDS) {
            String word = in2.next();
            if (word.startsWith(str)) {
                arr[words++] = word;
            }
        }
        in2.close();
    }

    public static void writeToFile(String[] arr, String file)
        throws FileNotFoundException {

        PrintWriter out = new PrintWriter(new File(file));
        for (int i = 0; arr[i] != null; i++) {
            out.println(arr[i]);
        }
        out.close();
    }

}
```

## TP10

### Input:

a

the objective of the exercise is to practice  
the manipulation of some data structures and  
text files. Part of the exercise is about  
text files and another part is about arrays  
unidimensional arrays

### Output:

and  
about  
and  
another  
about  
arrays  
arrays

### Exercício 2

A informação da taxa de crescimento de vários países, referente aos últimos 7 anos, encontra-se armazenada no ficheiro de texto "growthRate.txt". Cada linha do ficheiro contém o nome do país e as respetivas taxas de crescimento dos últimos 7 anos, separados por vírgulas, por exemplo: França, 0.53, 0.5, -0.5, 0.47, -0.45, 0.41, 0.39

Pretende-se executar um conjunto de operações sobre esta informação. Neste sentido, elabore um programa modular que implemente as seguintes funcionalidades:

- Ler para memória a informação do ficheiro de texto;
- Mostrar todos os países e respetivas taxas de crescimento;
- Mostrar o nome dos países onde não houve qualquer taxa de crescimento positiva;
- Remover da memória a informação de um país;
- Gravar no ficheiro de texto "growthRate2.txt" a informação em memória.

### Uma proposta de resolução

```
public class TP10_2 {

    static final int MAX_ROWS = 10;
    static final int MAX_COLS = 7;
    static final String FILE_IN = "growthRates.txt";
    static final String FILE_OUT = "growthRates2.txt";

    public static void main(String[] args) throws FileNotFoundException {
        String[] countries = new String[MAX_ROWS];
        double[][] growthRates = new double[MAX_ROWS][MAX_COLS];
        int rows = 0;

        //a)
        rows = readInfoFromFile(countries, growthRates, FILE_IN);
        //b)
        printInfo(countries, growthRates, rows);
        //c)
        displayCountriesWithGrowthRateN(countries, growthRates, rows);
        //d)
        if (deleteCountryInfo(countries, growthRates, rows, "Alemanha")) {
            rows--;
        }
        //e)
        writeInfoToFile(countries, growthRates, rows, FILE_OUT);
    }
}
```

## TP10

```
public static int readInfoFromFile(String[] arrCountries,
                                   double[][] arrGrowthRates,
                                   String file) throws FileNotFoundException
{
    Scanner in = new Scanner(new File(file));
    int rows = 0;
    while (in.hasNextLine() && rows < MAX_ROWS) {
        String[] itens = in.nextLine().split(",");
        arrCountries[rows] = itens[0];
        for (int item = 1; item < itens.length; item++) {
            arrGrowthRates[rows][item - 1] = Double.parseDouble(itens[item]);
        }
        rows++;
    }
    in.close();
    return rows;
}
//===== b)

public static void printInfo(String[] arrCountries,
                              double[][] arrGrowthRates,
                              int rows) {
    System.out.println("-----");
    for (int row = 0; row < rows; row++) {
        System.out.printf("%-10s", arrCountries[row]);
        for (int col = 0; col < MAX_COLS; col++) {
            System.out.printf("%6.2f", arrGrowthRates[row][col]);
        }
        System.out.println("");
    }
    System.out.println("-----");
}
//===== c)

public static void displayCountriesWithGrowthRateN(String[] arrCountries,
                                                    double[][] arrGrowthRates,
                                                    int rows) {
    System.out.println("Countries Without Growth Rate (last 7 years)");
    for (int row = 0; row < rows; row++) {
        if (hasNotGrowthRate(arrGrowthRates[row])) {
            System.out.println(arrCountries[row]);
        }
    }
}
//===== c.1)

public static boolean hasNotGrowthRate(double[] arrRates) {
    for (int column = 0; column < MAX_COLS; column++) {
        if (arrRates[column] > 0.0) {
            return false;
        }
    }
    return true;
}
```

## TP10

```
//===== d)

public static boolean deleteCountryInfo(String[] arrCountries,
                                       double[][] arrGrowthRates,
                                       int rows,
                                       String country) {
    int pos = findCountryIndex(arrCountries, rows, country);
    if (pos < 0) {
        return false;
    }

    //-----<remove row>
    for (int row = pos; row < rows - 1; row++) {
        //.....<country line>
        arrCountries[row] = arrCountries[row + 1];
        //.....<rate line>
        arrGrowthRates[row] = arrGrowthRates[row + 1];
    }
    //-----</remove row>
    return true;
}
//===== d.1)

public static int findCountryIndex(String[] arrCountries,
                                   int rows,
                                   String country) {
    for (int row = 0; row < rows; row++) {
        if (arrCountries[row].equalsIgnoreCase(country)) {
            return row;
        }
    }
    return -1;
}
//===== e)

public static void writeInfoToFile(String[] arrCountries,
                                   double[][] arrGrowthRates,
                                   int rows,
                                   String file)
    throws FileNotFoundException {
    Formatter out = new Formatter(new File(file));

    for (int row = 0; row < rows; row++) {
        //.....<country>
        out.format("%s", arrCountries[row]);
        //.....<rates>
        for (int col = 0; col < MAX_COLS; col++) {
            out.format(",%.2f", arrGrowthRates[row][col]);
        }
        out.format("%n");
    }
    out.close();
}
}
```