

Aula prática nº 10

Tópicos

- Ficheiros de texto

Exercícios

1.

O seguinte excerto de código permite ler todas as palavras (palavra a palavra) de um ficheiro de texto, que terá de estar localizado na pasta do projeto Eclipse/NetBeans. Pode criar este ficheiro com um editor de texto ou usar um qualquer ficheiro de código java.

```
public static void main(String[] args) {
    Scanner input;
    try {
        input = new Scanner(new File("words.txt"));
        while (input.hasNext()) {
            String word = input.next();
            System.out.println(word);
        }
    } catch (FileNotFoundException e) {
        System.out.println("Ficheiro não existente.");
    }
}
```

- Teste o excerto de código de forma a listar o conteúdo do ficheiro.
- Guarde numa estrutura de dados adequada todas as palavras com mais do que 2 caracteres
- Liste todas as palavras terminadas em 's'.
- Remova da estrutura todas as palavras que contenham outros caracteres que não letras.
- Liste novamente o resultado final.

2.

Construa um programa que leia um ficheiro de texto e que conte todos os pares de palavras encontrados no ficheiro e o número de ocorrências de cada par. Despreze todas as palavras de tamanho inferior a 3 e considere como separadores os seguintes caracteres:

`\t\n.,:!"';?!*{}+=&/()[]""'"\`

Se utilizar a primeira frase deste problema o resultado no ficheiro de saída deverá ser o seguinte:

```
cada={par=1}
como={separadores=1}
considere={como=1}
construa={programa=1}
conte={todos=1}
despreze={todas=1}
encontrados={ficheiro=1}
ficheiro={numero=1, texto=1}
inferior={considere=1}
leia={ficheiro=1}
numero={ocorrencias=1}
ocorrencias={cada=1}
```

```

palavras={encontrados=1, tamanho=1}
par={despreze=1}
pares={palavras=1}
programa={que=1}
que={conte=1, leia=1}
seguintes={caracteres=1}
separadores={seguintes=1}
tamanho={inferior=1}
texto={que=1}
todas={palavras=1}
todos={pares=1}

```

Para cada palavra encontrada no ficheiro o programa deverá associar um conjunto de todas as palavras seguintes contando igualmente quantas vezes cada par ocorre.

Teste o programa com o ficheiro “major.txt”. O resultado deverá ser semelhante a:

```

1864={tratava=1}
abacateiros={entoando=1, mangueiras=1, oitenta=1, suas=1, troncos=1}
abacates={ora=1, pouco=1}
abacaxis={coroados=1, que=1}
abafada={comprimida=1, tens=1}
...
voltava={aos=1, biblioteca=1, com=1, ficava=1, idéia=1, olhava=1, para=1, seu=1}
voltavam={mesmo=1, oficial=1, para=1, sorridentes=1}
voltou={acidente=1, agarrou=1, alegria=1, aos=1, bonde=1, general=1, instante=1, lhe=1}

```

3.

No dossier da disciplina encontra, para além deste guião, dois ficheiros de texto: *voos.txt* e *companhias.txt*.

O primeiro representa o voos que chegaram ao aeroporto do Porto no dia 24 de Maio. A estrutura deste ficheiro é a seguinte (os campos são separados por *tab*):

Hora	Voo	Origem	Atraso
00:50	TP 1944	Lisboa	
07:00	AEA1147	Madrid	
07:35	IB 8720	Madrid	00:25
...			

O segundo contém uma tabela com as siglas e os nomes de cada companhia:

Sigla	Companhia
A5	HOP!
AE	Air Europa
DT	TAAG
EV	Evelop Airlines
EZ	EasyJet Airlines
FR	Ryanair
...	

Construa um programa que leia estes dois ficheiros para estruturas adequadas. Crie a classe *Voo*, por exemplo, use conjuntos para armazenar os voos em memória, bem como outras estruturas/algoritmos que ache necessários para cada uma das alíneas seguintes.

Para ler de, ou escrever para, um ficheiro de texto pode usar como base o seguinte código:

```

// ler
...
try {
    Scanner sc = new Scanner(new File(filename));
    while(sc.hasNextLine()) {
        String line = sc.nextLine();
    }
}

```

```

...
} catch (FileNotFoundException e) {
...

// escrever
...
try {
    FileWriter writer = new FileWriter("cidades.txt");
    writer.write("Origem"+"\t"+"Voos"+"\n");
    ...
    writer.close();
} catch (IOException e) {
...

```

- a) Apresente no ecrã a lista de voos com informação mais completa, tal como consta da tabela seguinte:

Hora	Voo	Companhia	Origem	Atraso	Obs
00:50	TP 1944	TAP Portugal	Lisboa		
07:00	AEA1147	Air Europa	Madrid		
07:35	IB 8720	Iberia	Madrid	00:25	Previsto: 8:00
07:35	T0 3408	Transavia France	Paris, Orly		
07:40	FR 5451	Ryanair	Faro		
07:55	EZY3771	EasyJet Airlines	Paris, Ch. de Gaulle	00:33	Previsto: 8:28
08:05	TP 459	TAP Portugal	Paris, Orly	00:13	Previsto: 8:18
08:15	FR 1386	Ryanair	Brussels, Charleroi		
08:20	FR 4171	Ryanair	Frankfurt, Hahn		
08:30	S4 370	Sata Internacional	Toronto, Pearson	00:35	Previsto: 9:05
...					

- b) Guarde a tabela no ficheiro *Infopublico.txt*.
c) Calcule a média dos atrasos por companhia e apresente no ecrã uma tabela (Companhia, Atraso médio) ordenada por ordem crescente de atraso médio.
d) Guarde no ficheiro *cidades.txt* uma tabela com informação com o total de chegadas de cada cidade origem. Exemplo (ordenação por número de voos):

Origem	Voos
Lisboa	11
Madrid	9
Paris, Orly	8
...	

- e) Repita a alínea b) mas agora armazenando os dados num ficheiro binário (*Infopublico.bin*). Construa também um método para ler o ficheiro e apresentar a tabela no ecrã. Use como base a classe *RandomAccessFile*.

```
RandomAccessFile file = new RandomAccessFile("Infopublico.bin", "rw");
```