

# Aula Prática 6

## Resumo:

- Funções recursivas.

### Exercício 6.1

A função de Fibonacci<sup>1</sup> de um número inteiro (não negativo)  $n$  pode ser definida por:

$$F(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ F(n-2) + F(n-1) & \text{se } n > 1 \end{cases} . \quad (6.1)$$

Complete a função `fibonacci` e teste-a no programa `p61`, que escreve a sequência dos números de Fibonacci até  $F(N)$ .

### Exercício 6.2

Construa uma função recursiva que imprima um array de strings, uma string por linha. O algoritmo é simples: para imprimir  $N$  linhas, imprimimos as  $N - 1$  primeiras e depois a última. Use-a no programa `p62` para imprimir os argumentos da linha de comando.

Repare que, além do array, é conveniente a função ter um parâmetro extra que permita indicar que parte do array deve imprimir. Este “truque” permite usar o mesmo array em todas as invocações, e variar apenas o segundo parâmetro. Assim não é preciso criar um novo array em cada invocação.

### Exercício 6.3

Altere a função recursiva do programa anterior (copiando-o para um novo programa) por forma a que agora escreva as strings por ordem inversa.

### Exercício 6.4

Construa uma função recursiva – `invertString` – que inverta uma qualquer `String` passada como argumento. Para testar a função, implemente um programa que a aplique a cada um dos argumentos.

---

<sup>1</sup>Matemático italiano dos Séculos XII-XIII, responsável, entre outros feitos, pela introdução da chamada numeração árabe na Europa.

### Exercício 6.5

Escreva um programa que mostre o conteúdo de um directório e de todos os seus subdirectórios recursivamente.

Por exemplo, se for executado o comando `java -ea p62 ..`, o resultado deverá assemelhar-se ao seguinte.

```
..
../aula01
../aula01/p11.java
../aula01/p15.java
../aula01/p13.java
../aula01/p16.java
../aula01/p12.java
../aula01/p14.java
../aula02
../aula02/p25.java
../aula02/p22.java
../aula02/Contacto.java
../aula02/pt
../aula02/pt/ua
../aula02/pt/ua/prog2
../aula02/pt/ua/prog2/Contacto.java
(...)
```

**Nota:** Sugere-se a utilização das funções `listFiles` e `getPath` do módulo `File` para obter, respectivamente, a lista de ficheiros existentes num directório e a localização de cada ficheiro.

### Exercício 6.6

Um cliente de um banco pede um empréstimo de  $M$  Euros com uma taxa de juro de  $T\%$  ao mês e uma prestação de  $P$  Euros no fim de cada mês.

- Determine a relação de recorrência que descreve o montante em dívida  $D_n$  ao fim de  $n$  meses.
- Implemente, com o método iterativo, uma função para determinar  $D_n$ .
- Implemente, com o método recursivo, uma função para determinar  $D_n$ .

### Exercício 6.7

- Escreva um programa que encontre numa árvore de directórios todos os ficheiros com um determinado nome.

Por exemplo, se for executado o comando `java -ea p64 .. Contacto.java`, a saída deverá ser:

```
../aula02/Contacto.java  
../aula02/pt/ua/prog2/Contacto.java
```

**Nota:** Sugere-se a utilização da função `getName` do módulo `File` para aceder ao nome do ficheiro.

- b. Generalize o programa anterior por forma a encontrar ficheiros que contenham um determinado texto no seu nome.

Por exemplo, se for executado o comando `java -ea p65 .. Totoloto`, a saída deverá ser a mesma.

**Nota:** Sugere-se a utilização da função `indexOf` do módulo `String` para verificar a ocorrência do texto no nome.

