



## **Programação I**

### **Folha de Exercícios 5**

António J. R. Neves  
João Rodrigues  
Osvaldo Pacheco  
Arnaldo Martins

2018/19

## Folha de Exercícios 5

### Resumo:

- Introdução à programação modular
- Funções
- Tipos primitivos como argumentos
- Visibilidade das variáveis

Quando especificamos um problema obtemos um conjunto de tarefas básicas que este tem de realizar (ex: ler, calcular, imprimir). Com o aumento da complexidade dos problemas que queremos resolver, torna-se vantajoso a implementação e teste de cada uma dessas tarefas em separado.

A linguagem JAVA, como a maioria das linguagens de programação, permite-nos criar funções para implementar as várias tarefas básicas de um programa. Uma função permite realizar um determinado conjunto de operações e, se necessário, devolver um valor lido ou calculado dentro da mesma.

As funções desenvolvidas pelo programador são chamadas no programa da mesma forma que as funções criadas por terceiros (por exemplo as funções de leitura ou escrita de dados ou as funções da classe **Math**).

Para escrever os programas dos exercícios seguintes, deve começar pelas especificações completas e os algoritmos finais em pseudocódigo, dando particular atenção à separação dos programas em tarefas bem definidas que se virão a tornar as funções que deverá implementar. Os problemas apresentados devem então ser resolvidos com funções e na sua solução utilizam todos os conceitos já apresentados: sequenciação de instruções, instruções de decisão e instruções de repetição.

De salientar que alguns dos exercícios já foram resolvidos nas aulas anteriores e aparecem aqui com o objetivo de refinar a sua solução utilizando funções.

### 5.1 Problemas para resolver

#### Exercício 5.1

O programa **Functions.java** define uma nova função, **sqr**, e invoca-a na função principal para a testar. Experimente compilar e executar o programa. Analise o código fonte e experimente alterar as chamadas à função.

**Exercício 5.2**

Acrescente à classe **Functions** as funções abaixo, uma a uma, e teste-as na função principal. (Algumas destas funções poderão ser reutilizadas nos problemas seguintes.)

- Uma função **f** que implemente:

$$f: \mathbb{Z} \rightarrow \mathbb{R}$$
$$n \mapsto \frac{1}{1+n^2}$$

Repare como a definição da função em Java reflete todos os elementos da definição matemática (tipo do parâmetro, tipo do resultado, forma de calcular o resultado).

- Uma função **max** que calcule o maior de dois números reais, passados como argumento à função.
- Uma função **max** que calcule o maior de dois números inteiros, passados como argumento à função. Qual a diferença em relação à função anterior? Repare que estas duas funções podem ter o mesmo nome. Como é que o compilador sabe qual invocar?
- Uma função **poly3** que calcula o valor de um polinómio de 3º grau:

$$p(x)=ax^3+bx^2+cx+d.$$

sendo  $a, b, c, d, x$  os seus parâmetros reais.

- Uma função inteira **fact** que calcule o valor do fatorial de um número inteiro, passado como argumento à função.  $N! = 1 \times 2 \times 3 \times \dots \times N$ ;
- Uma função inteira **getIntPos** que leia com validação do teclado um valor inteiro positivo. A leitura deve ser repetida enquanto o valor não for positivo. A função deve apresentar uma mensagem a pedir o valor (um *prompt*). Essa mensagem deve ser um parâmetro da função.
- Uma função inteira **getIntRange** que leia com validação um valor dentro de um intervalo, sendo os limites desse intervalo passados como argumentos à função. (O *prompt* também deve ser um parâmetro.)
- Uma função (**printNtimes**) que escreva N cópias de uma mensagem. Por exemplo: **printNtimes(3, "Ola")** deve escrever "OlaOlaOla". Note que esta é uma função que não devolve qualquer resultado (é **void**), mas produz efeitos (imprime texto).

**Exercício 5.3**

Escreva um programa que lê do teclado uma data composta pelo mês e o ano, calcula e escreve no monitor o número de dias desse mês, tendo como base o exercício da **folha de exercícios 2**. Comece por implementar uma função booleana que indique se um ano é ou não bissexto e uma função que devolva o número de dias de um certo mês (de certo ano). Faça a leitura com validação do mês (entre 1 e 12) e do ano (positivo) utilizando as funções implementadas no exercício 5.2

**Exercício 5.4**

Escreva um programa que calcule e imprima o fatorial dos números entre  $M$  e 1, sendo o valor de  $M$  lido do teclado e validado (entre 1 e 10). Procure reutilizar as funções desenvolvidas no exercício 5.2 para a sua solução. O resultado do programa deve ser o seguinte para  $M = 4$ :

```
4! = 24
3! = 6
2! = 2
1! = 1
```

**Exercício 5.5**

Escreva um programa que imprima no terminal a linha exterior de um retângulo com o símbolo \*. As dimensões dos lados são pedidas ao utilizador. Para uma largura de 5 e altura de 4 esperamos o seguinte resultado:

```
*****
*      *
*      *
*      *
*****
```

Sugestão: pode usar a função **printNtimes** para a escrita das linhas horizontais e para a escrita dos espaços no interior do retângulo.

**Exercício 5.6**

Escreva um programa que leia do teclado um número inteiro  $N$ , com  $0 < N < 100$ , e que escreva no monitor a sua tabuada, com o formato que a seguir se apresenta. Use a função para a leitura e validação do valor de entrada desenvolvida no exercício 5.2 e implemente uma função para a escrita da tabuada no monitor.

```
-----
|  Tabuada dos ##  |
-----
| ## x 1  | ### |
| ## x 2  | ### |
| ## x 3  | ### |
| ## x 4  | ### |
| ## x 5  | ### |
| ## x 6  | ### |
| ## x 7  | ### |
| ## x 8  | ### |
| ## x 9  | ### |
| ## x 10 | ### |
-----
```

## 5.2 Exercícios complementares

### Exercício 5.7

Implemente uma função que determine o máximo divisor comum (mdc) entre dois números inteiros. Use o algoritmo de Euclides. Este algoritmo consiste em subtrair o menor número ao maior até que os dois números se tornem iguais. (Para maior eficiência, pode usar o resto da divisão em vez de subtração sucessiva.)

### Exercício 5.8

Escreva um programa que produza uma tabela com os valores dos polinómios reais  $5x^2+10x+3$  e  $7x^3+3x^2+5x+2$  calculados para uma sequência regular de valores de  $x$ . Os valores inicial e final de  $x$  e o número de elementos da sequência são introduzidos pelo utilizador. A leitura dos dados deve exigir que o valor final seja superior ao valor inicial, para garantir valores crescentes de  $x$ . Use a função de cálculo de polinómios de 3º grau que fez no exercício 5.2. (Note que um polinómio de 2º grau é um caso particular de polinómio de 3º grau no qual o primeiro coeficiente é nulo.) Apresente a tabela com o formato indicado abaixo.

```

-----
|   x   | 5x2+10x+3 | 7x3+3x2+5x+2 |
-----
| ###.# | #####.## | #####.### |
-----
...
-----
| ###.# | #####.## | #####.### |
-----

```

### Exercício 5.9

Crie uma função booleana **isprime** que indique se um parâmetro inteiro **n** é um número primo. Escreva um programa para listar todos os primos entre 1 e  $M$ . O valor  $M$  (positivo) deve ser lido com validação.