

# **Programação I**

## **Folha de Exercícios 6**

António J. R. Neves  
João Rodrigues  
Osvaldo Pacheco

2017/18

## Folha Exercícios 6

### Resumo:

- Introdução aos **arrays**
- Declaração de variáveis do tipo **array**
- Acesso aos valores de um **array**
- **Arrays** como argumentos de funções

Vimos anteriormente que é possível criar novos tipos de dados referência que permitem declarar variáveis onde é possível guardar mais do que um valor.

No entanto, existem aplicações informáticas que precisam de lidar com grandes volumes de dados, pelo que não é eficiente ter uma variável para cada valor a armazenar.

A linguagem Java disponibiliza outro tipo de dados referência, os arrays, (podemos descrever em português como sequências, vetores ou tabelas). Nesta aula prática pretende-se introduzir este tipo de dados estruturado homogêneo. Um **array** é uma organização de memória que se caracteriza pelo facto de ser um agregado de células contíguas, capaz de armazenar um conjunto de valores do mesmo tipo e aos quais se pode aceder de forma indexada. Nesta aula iremos abordar problemas que nos permitam compreender como definir e utilizar arrays.

### 6.1 Problemas para resolver

#### Exercício 6.1

Escreva um programa que leia uma sequência de  $N$  números inteiros, sendo o valor  $N$  pedido ao utilizador antes do início da introdução dos números. O programa deve depois imprimir esses números pela ordem inversa com que foram inseridos.

#### Exercício 6.2

Escreva um programa que leia uma sequência de números inteiros positivos e conte o número de vezes que um determinado número, pedido ao utilizador, aparece na sequência. A leitura deve terminar após a introdução de 100 números ou com o aparecimento de um número negativo.

### Exercício 6.3

Pretende-se escrever um programa que leia do teclado uma sequência de números inteiros positivos e que permita detetar um conjunto de características acerca da sequência. A cada operação deverá corresponder uma função. A leitura da sequência termina quando aparecer o número zero como indicador de paragem ou quando tiverem sido lidos 50 números. A interação com o programa deverá ser feita através de um menu, tal como apresentado de seguida:

Análise de uma sequência de números inteiros

- 1 - Ler a sequência
  - 2 - Escrever a sequência
  - 3 - Calcular o máximo da sequência
  - 4 - Calcular o mínimo da sequência
  - 5 - Calcular a média da sequência
  - 6 - Detetar se é uma sequência só constituída por números pares
  - 10 - Terminar o programa
- Opção ->

### Exercício 6.4

Tendo como base o problema 6.2, considere agora que, dada uma determinada sequência de notas (valores inteiros de 0 a 20), pretende-se construir e desenhar o respetivo histograma (contagem do número de ocorrências de cada nota). O número de notas a processar deverá ser pedido ao utilizador no início do programa. Implemente esse programa, desenhando o histograma com o seguinte aspeto:

```
Histograma de notas
-----
20 | *****
19 | *
.
.
.
01 | *
00 | **
```

Comece por associar um "\*" a cada nota encontrada, fazendo depois a normalização (linear) do gráfico para que o valor máximo do histograma corresponda a 50 asteriscos.

## 6.2 Exercícios complementares

### Exercício 6.5

Escreva um programa que leia uma sequência de  $N$  números reais, sendo o valor  $N$  pedido ao utilizador antes do início da introdução dos números. O programa deverá calcular a média e o desvio padrão da sequência e imprimir no ecrã os valores superiores à média.

### Exercício 6.6

Escreva um programa que leia uma frase e imprima no monitor quais as letras do alfabeto que apareceram nessa frase. Para a resolução deste problema, sugere-se a utilização de uma sequência de valores booleanos de modo a sinalizar quais os caracteres do alfabeto que apareceram pelo menos uma vez. Usar a função **charAt(i)** para obter o carácter  $i$  de um string.

Exemplo: `"aveiro".charAt(1) → 'v'`

### Exercício 6.7

Considere agora um caso genérico do problema 6.2, e escreva um programa que conte o número de vezes que cada elemento ocorre numa sequência de números inteiros. A leitura deve terminar após a introdução de 100 números ou com o aparecimento de um número negativo.

Considere o seguinte exemplo. Supondo a sequência:

$a = \{4, 2, 5, 4, 3, 5, 2, 2, 4\}$

O resultado do programa devia ser o seguinte:

```
4 ocorre 3 vezes
2 ocorre 3 vezes
5 ocorre 2 vezes
3 ocorre 1 vez
```