

## BLOCO 2 – Implementação de algoritmos em Java e decomposição modular – PL5

**ASSUNTO** - Implementação de algoritmos em Java

### OBJETIVOS GERAIS:

- Implementar programas em Java

### OBJETIVOS ESPECÍFICOS:

- Codificar em Java:
  - Entrada de dados
  - Saída de resultados
  - As estruturas de controlo de fluxo
- Documentar um programa:
  - Utilização dos identificadores adequados
  - Comentários essenciais
  - Indentação
- Mediante um problema analisar, conceber e descrever o algoritmo através de pseudocódigo e implementá-lo em Java.

### TAREFAS DA SEMANA:

#### Exercício 1 (\*)

- a) Descreva a funcionalidade do seguinte algoritmo e complete a instrução de escrita.

```
ED: n, num, s, c, i INTEIRO
    media REAL
INICIO
    s ← 0
    c ← 0
    LER(n)
    PARA ( i ← 1 ATÉ n PASSO 1) FAZER
        LER(num)
        SE (num % 2 = 0) ENTÃO
            c ← c+1
            s ← s + num
        FIMSE
    FIMPARA
    SE (c ≠ 0) ENTÃO
        m ← s/c
        ESCREVER (m, c/n )
    SENÃO
        ESCREVER ("NÃO EXISTE ...")
    FIMSE
FIM
```

- b) Codifique-o em Java corrigindo todos os aspetos que considere relevantes, incluindo nomes de variáveis.

## BLOCO 2 – Implementação de algoritmos em Java e decomposição modular – PL5

- c) Defina um plano de testes adequado (pelo menos 3 testes)
- d) Implemente o programa utilizando o ambiente NetBeans.
  - 1. Crie um projeto de nome PL5\_1
  - 2. Aceite a proposta de criar uma classe de nome PL5\_1 com o método main
  - 3. Verifique a criação do *package* com o mesmo nome do projeto
  - 4. Edite o código e utilize as ajudas na escrita de código (*autocompletion* ou *intellisense*)
  - 5. Compile
  - 6. Analise os ficheiros criados no projeto pelo Netbeans
- e) Teste o programa adequadamente

### Exercício 2 (\*)

Descreva um algoritmo e codifique-o em Java, em que dadas as temperaturas máximas registadas em N dias classifique o dia com a temperatura máxima mais elevada, de acordo com a tabela abaixo. Caso a temperatura máxima mais elevada ocorrida nos N dias seja menor que  $-30^{\circ}\text{C}$  ou maior ou igual a  $50^{\circ}\text{C}$  deverá ser enviada ao utilizador a mensagem “Temperatura extrema”. Para a resolução do exercício considere que todas as temperaturas máximas introduzidas têm valores inteiros.

$-30^{\circ}\text{C} \leq \text{Temp} < 9^{\circ}\text{C}$	Muito Frio
$9^{\circ}\text{C} \leq \text{Temp} < 15^{\circ}\text{C}$	Frio
$15^{\circ}\text{C} \leq \text{Temp} < 20^{\circ}\text{C}$	Ameno
$20^{\circ}\text{C} \leq \text{Temp} < 30^{\circ}\text{C}$	Quente
$30^{\circ}\text{C} \leq \text{Temp} < 50^{\circ}\text{C}$	Muito Quente

### Exercício 3 (\*)

Elabore um programa que leia uma sequência de nomes e de idades, e apresente todos os nomes e a percentagem de pessoas com idade maior ou igual a 18. A leitura termina quando for introduzido o nome “zzz”.

### Exercício 4 (\*\*)

Elabore um programa em Java que determine e visualize os N primeiros números perfeitos. Um número é perfeito se for natural e for igual à soma de todos os seus divisores (excluindo o próprio número). Na codificação do programa utilize a classe Scanner para a entrada de dados e a classe System para a saída de dados.

## BLOCO 2 – Implementação de algoritmos em Java e decomposição modular – PL5

### Exercício 5 (\*)

- a) Descreva a funcionalidade do seguinte algoritmo.

```
ED: num, d, aux, res INTEIRO
INICIO
    res ← 0
    aux ← 1
    LER(num)
    ENQUANTO (num≠0) FAZER
        d ← num%10
        SE (d%2=1) ENTÃO
            res ← res + d * aux
            aux ← aux * 10
        FIMSE
        num ← num / 10
    FIMENQUANTO
    ESCRIVER("O resultado é :", res)
FIM
```

- b) Codifique-o em Java.

### Exercício 6 (\*\*)

Elabore um programa em Java que leia duas sequências de números, a primeira terminada com 0 e a segunda que termina em -1 e determine qual a sequência (a 1ª ou a 2ª) que contém mais números pares.

### Exercício 7 (\*\*)

Codifique em Java o seguinte algoritmo e verifique a sua funcionalidade.

```
ED a, b, aux, num, c, d, e INTEIRO
INICIO
    LER(a, b)
    SE (a>b) ENTÃO
        aux ← a
        a ← b
        b ← aux
    FIMSE
    e ← 0
    LER(d)
    PARA(c ← 1 ATÉ d PASSO 1) FAZER
        REPETIR
            LER(num)
            ENQUANTO (num<0)
                SE (num%a=0 AND b%num=0) ENTÃO
                    e ← e + 1
            FIMSE
        FIMPORA
    ESCRIVER("...", e)
FIM
```

## **BLOCO 2 – Implementação de algoritmos em Java e decomposição modular – PL5**

### **Exercício 8 (\*)**

Elabore um programa em Java que permita receber um número e verificar se é um número binário.

### **Exercício 9 (\*\*)**

Elabore um programa em Java que receba um número binário e o converta para o sistema decimal.

### **Exercício 10 (\*\*)**

Elabore um programa em Java que receba um número decimal e o converta para o sistema binário.

## **Exercícios Complementares**

### **Exercício 1 (\*)**

Corrija os erros do seguinte programa em Java.

```
public class Main {

    public static void main(String[] args) {

        int numero, alg, i, j;

        resp = JOptionPane.showInputDialog("Qual o número?");
        numero = Integer.parseInt(resp);

        do {
            j = j + 1;
            alg = numero % 10;
            if (alg%2 = 0)
                i = i + 1;
            numero = numero/10;
        }while (numero>0);

        perc = (float) i / j + 0.5f;
        JOptionPane.showMessageDialog("Valor=" + (int) perc);
    }
}
```

## BLOCO 2 – Implementação de algoritmos em Java e decomposição modular – PL5

### Exercício 2 (\*\*\*)

Elabore um programa em Java que permita ler N números inteiros positivos no sistema de numeração binário e apresentá-los no sistema de numeração decimal.

Valide adequadamente os valores de entrada.

Elabore um plano adequado de testes para ser executado na validação do programa.

### Exercício 3 (\*\*\*)

Na sequência 6788, 2688, 768, 336, 54, 20, 0, cada termo é o produto dos dígitos do número anterior:

$$6 * 7 * 8 * 8 = 2688$$

$$2 * 6 * 8 * 8 = 768$$

Para um dado número inicial, o número de passos até que se atinja um número com um único dígito (não necessariamente zero) é designado por “persistência” desse número (no exemplo acima é 6).

Escreva um programa em Java para calcular a persistência de um número dado via teclado.

### Exercício 4 (\*\*)

Elabore um programa em Java para mostrar os primeiros N termos da sucessão de Fibonacci, onde N é definido pelo utilizador.

Nesta sucessão, o primeiro termo é zero, o segundo termo é um e qualquer um dos outros termos é igual à soma dos dois anteriores.

### Outros Exercício

Codifique em Java os algoritmos construídos na PL4.

