



ASSUNTO- Codificação de programas em Java

Objetivos Específicos:

Os alunos devem ser capazes de:

- Codificar em Java:
 - Entrada de dados
 - Saída de resultados
 - As estruturas de controlo de fluxo
- Documentar um programa:
 - Utilização dos identificadores adequados
 - Comentários
 - Indentação
- Compreender e utilizar o ambiente de desenvolvimento integrado NetBeans
- Elaborar um plano de testes adequado e testar o programa utilizando esse plano
- Mediante um problema analisar, conceber e descrever o algoritmo através de pseudo código e implementá-lo em Java.

Métodos/Técnicas	Recursos Didáticos	Avaliação
Interrogativo e Ativo	Quadro e Computador	TFormativa com Formulação de Perguntas e Observação

Conteúdo da aula

Exercício Demonstrativo Aula1

- a) Analise o seguinte programa em Java e deduza a sua funcionalidade.

```
import javax.swing.JOptionPane;
public class PL4_1{
    public static void main(String[] args) {
        int i=0;
        double nota, media, soma=0;
        String vAuxiliar;
        vAuxiliar=JOptionPane.showInputDialog("Qual a nota? (para terminar digite 0)");
        nota=Double.parseDouble(vAuxiliar);
        while (nota !=0) {
            soma = soma + nota;
            i++;
            vAuxiliar = JOptionPane.showInputDialog("Qual a nota? (para terminar digite 0)");
            nota=Double.parseDouble(vAuxiliar);
        }
        if (i>0) {
            media=soma/i;
            JOptionPane.showMessageDialog(null, "A média das notas="+media);
        }
        else {
            JOptionPane.showMessageDialog(null, "Não foram introduzidas notas");
        }
    }
}
```





- b) Elabore um plano de testes.
- c) Implemente o programa utilizando o IDE NetBeans
 - a. Crie um projeto de nome PL4_1
 - b. Aceite a proposta de criar uma classe de nome PL4_1 com o método main
 - c. Verifique a criação do package com o mesmo nome do projeto
 - d. Edite o código e utilize as ajudas na escrita de código (autocompletion ou intellisense)
 - e. Compile
 - f. Execute de acordo com o plano de testes definido
- d) Comente o código adequadamente.

Exercício Demonstrativo Aula2

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.

Exercícios para resolver

Exercício 1

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

```
ED: n, num, s, c, i INTEIRO
    m 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 )
    SENÃO
        ESCREVER (“NÃO EXISTE ...”)
    FIMSE
FIM
```

- b) Codifique-o em Java corrigindo todos os aspectos que considere relevantes.
- c) Implemente o programa utilizando o ambiente NetBeans.





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°C ou maior ou igual a 50°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 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

- 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
    ESCREVER(“O resultado é :“,res)
FIM
```

- b) Codifique-o em Java.





Exercício 5

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 E b%num=0) ENTÃO
          e ← e + 1
        FIMSE
      FIMPARA
    ESCREVER("... ", e)
  FIM
```

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 calcule qual a sequência (a 1ª ou a 2ª) que contém mais números pares.

Exercícios Complementares

Exercício 1

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).

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

Exercício 2

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.

Exercício 3

Codifique em Java os algoritmos construídos na PL3.

