



ASSUNTO - Descrição de Algoritmos estruturando-os em módulos e codificação em Java

Objetivos Específicos:

- Mediante apresentação de um problema os alunos deverão ser capazes de o analisar, conceber e descrever o algoritmo através de pseudo código e implementá-lo em Java.
- Mediante a apresentação de um algoritmo em pseudo código o aluno deverá ser capaz de o implementar em Java
- Mediante a apresentação de código Java o aluno deverá ser capaz de compreender a sua funcionalidade e atualizá-lo.

Métodos/Técnicas	Recursos Didáticos	Avaliação
Interrogativo e Activo	Quadro e Computador	Tipo Formativa com Formulação de Perguntas e Observação

CONTEÚDO DA AULA

Conceba e descreva em pseudo código um algoritmo, sempre que adequado estruturado em módulos, para a realização computacional de cada um dos seguintes exercícios. Implemente-os em Java.

Exercício Demonstrativo1

Faça um programa que permita escrever tabuadas de multiplicação de N números inteiros positivos pertencentes a um intervalo fechado definido pelo utilizador.

Implemente o programa usando:

- Um módulo que leia continuamente n.ºs até ser introduzido um número pertencente a um intervalo recebido como parâmetro. O módulo retorna o número validado.
- Um módulo tabuada que escreve a tabuada de um determinado número recebido como parâmetro.

Exercício Demonstrativo2

1 - Analise o seguinte extrato de código cuja funcionalidade é obter o dia de hoje:

```
import java.util.Calendar;
...
Calendar hoje=Calendar.getInstance();
int dia=hoje.get(Calendar.DAY_OF_MONTH);
int mes=hoje.get(Calendar.MONTH)+1;
int ano=hoje.get(Calendar.YEAR);
...
```

2 - Pretende-se uma aplicação que recebe o nome e a data de nascimento (dia, mês e ano) de um conjunto de pessoas e imprime a média de idades, e qual o nome de uma das pessoas que tem a menor idade.

Notas:

- Faça um desenvolvimento modular do programa.
- Sugere-se a implementação das seguintes funcionalidades:
 - Verificação de data válida
 - Determinação da idade





Exercícios para resolver

Exercício 1

- a) Analise os seguintes módulos (em Java designados métodos) e identifique as respectivas funcionalidades.

```
public static boolean metodo1(String pal) {
    boolean resposta=true;
    pal=pal.toLowerCase();
    int tamanho = pal.length();
    for ( int i=0 ; i< tamanho /2 ; i++) {
        if ( pal.charAt( i ) != pal.charAt(tamanho -1- i ) ) {
            resposta=false; break;
        }
    }
    return resposta;
}

public static boolean metodo2(String pal) {
    int i,j;
    pal=pal.toLowerCase();
    i=0; j=pal.length()-1;
    while (i<j && pal.charAt(i)==pal.charAt(j)) {
        i++;
        j--;
    }
    return i>=j;
}
```

- b) Faça um programa que leia uma sequência de palavras até encontrar um palíndromo (palavra cuja leitura da esquerda para a direita é igual à da direita para a esquerda). O programa deve contar o número de palavras lidas que antecedem o palíndromo.

OBS: Utilize o método anterior que achar apropriado.

Exercício 2

Faça um programa que represente sob a forma de gráficos de barras o número de positivas e negativas dos alunos (A) a um conjunto de disciplinas (D). O programa deverá possuir um módulo para imprimir a informação de uma disciplina. O output produzido deverá ter o seguinte aspecto:

```
Disciplina: Português
- Positivas: *****
- Negativas: ****
Disciplina: Matemática
```

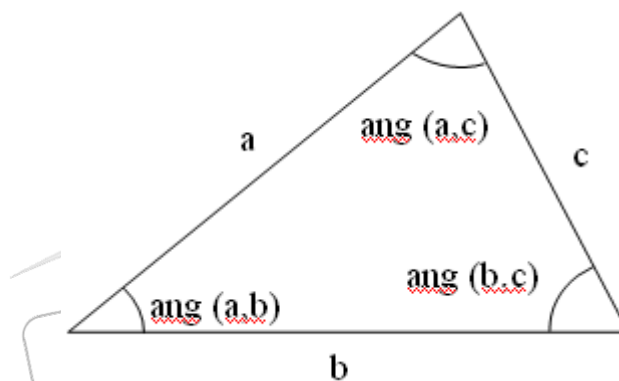




- Positivas: *****
- Negativas: *****

Exercício 3

- a) Faça um método que calcule um ângulo interno de um triângulo sendo dadas as medidas dos três lados desse triângulo. O valor do ângulo deve estar em graus.
- b) Sendas dadas as medidas dos três lados de um triângulo, verificar se as medidas são válidas e se é possível formar triângulo e em caso afirmativo calcule todos os ângulos internos desse triângulo. Para isso chame três vezes o método desenvolvido na alínea anterior.



Ângulo	Fórmula
$ang(a,b)$	$arc\ cos\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$
$ang(a,c)$	$arc\ cos\left(\frac{a^2 + c^2 - b^2}{2ac}\right)$
$ang(b,c)$	$arc\ cos\left(\frac{b^2 + c^2 - a^2}{2bc}\right)$

Exercício 4

Analise a classe CalCombinatorio definida nas aulas teóricas.

Crie um projeto onde vai incluir esta classe CalCombinatorio.

Acrescente à classe CalCombinatorio mais dois métodos de classe, o método arranjos e o método permutações.

Crie uma classe TesteCalcCombinatorio para testar as funcionalidades da classe.

Exercício 5

- a) Faça um módulo que dados dois números inteiros positivos retorna a quantidade de dígitos comuns nas mesmas posições.





- b) Elabore um programa que leia N pares de valores inteiros positivos, sendo N introduzido pelo utilizador e validado. Após a leitura dos N pares de valores o programa deve apresentar o par que tiver mais dígitos comuns.

Exercício 6

Faça um programa que permita determinar volumes de sólidos de revolução (cilindros, cones e esferas). Para cada sólido será introduzido o tipo de sólido e as respectivas dimensões. O programa termina quando o tipo de sólido for a palavra “FIM”. Implemente o programa de forma modular.

OBS:

$$V_{\text{esfera}} = \frac{4}{3} \pi R^3$$

$$V_{\text{cilindro}} = \text{Área Base} \times \text{Altura} = \pi R^2 \text{ Altura}$$

$$V_{\text{cone}} = \frac{1}{3} \pi R^2 \text{ Altura}$$

Exercício Complementar

Exercício 1 (**)

- a) Faça um módulo que verifique se um número é ou não um número capicua.
b) Faça um programa que leia uma sequência de números inteiros e termine quando for introduzido um número capicua ou ter analisado 100 números sem o encontrar. O programa deve escrever uma mensagem adequada.

Exercício 2 (**)

- a) Faça um módulo que calcula e retorna a soma de todos os divisores pares de um número dado como parâmetro. No entanto, não deve considerar o próprio número como divisor.
b) Elabore um programa que, dada uma sequência de números positivos, determina e apresenta a percentagem de números cuja soma dos seus divisores pares é a maior.

Exercício 3 (***)

- a) Faça um método que receba os seguintes parâmetros:
- Um número inteiro, sob a forma de texto, numa determinada base;
 - Base (de 2 a 16) na qual está especificado o número;
 - Base (de 2 a 16) para a qual vai ser convertido o número.

O método deverá verificar a validade de todos os parâmetros (incluindo o primeiro) e se estas forem bem sucedidas, aquele deverá retornar, sob a forma de string, o número convertido para





a base especificada para esse efeito. Se algum dos parâmetros for inválido o método deverá retornar string vazio. Os símbolos que podem surgir na representação de um número são os algarismos (de 0 a 9) ou as letras do alfabeto (de A a F).

- b) Faça um programa que teste o método anterior.

