



# CIÊNCIAS DA COMPUTAÇÃO

FACULDADE DE CIÊNCIAS | UNIVERSIDADE AGOSTINHO NETO



## FUNDAMENTOS DE PROGRAMAÇÃO

Docente:

✓ Lufialuiso Sampaio Velho, MSc.

Monitor

✓ João Pedro



## Cap. III

Estruturas de Selecção (decisão)

# Instrução de selecção (decisão)

- ❑ Em Java As estruturas de selecção comportam-se da mesma forma. Estas diferem-se somente pela sintaxe comparada aos pseudocódigos.

- ❑ Estrutura de Decisão simples.

Pseudocódigo	Em Java
<pre>se(condição) então     // Instruções fimse</pre>	<pre>if (condição) {     // Instruções }</pre>

- ❑ Utilizando esta estrutura, as instruções são somente executadas caso a condição avaliada seja verdadeira. Caso contrário, o programa segue o seu percurso normal; ou seja não faz nada.

# Instrução de selecção (decisão)

## ❑ Estrutura de decisão simples

- Exemplo: Crie um programa em Java que recebe a idade de um indivíduo e imprime “JOVEM” se a idade for maior ou igual a 18 e menor que 45.

### Pseudocódigo

```
idade:inteiro
Inicio

    escreva("Digite a sua idade")
    leia(idade)

    se ((idade>=18) e (idade<45)) entao
        escreva("JOVEM")
    Fimse

finalgoritmo
```

### Em Java

```
import java.util.Scanner;
public class Exercícios {
    public static void main(String[] args){

        Scanner teclado = new Scanner(System.in);

        System.out.println("Digite a sua Idade");
        int idade = teclado.nextInt();

        if((idade>=18) && (idade<45)){
            System.out.print("JOVEM");
        }
    }
}
```

# Instrução de selecção (decisão)

## ❑ Estrutura de decisão composta.

Pseudocódigo	Em Java
<pre>se(condição) então     // Instruções Senão     // Instruções fimse</pre>	<pre>if (condição) {     // Instruções }else{     // Instruções }</pre>

- ❑ Esta estrutura de selecção é implementada quando temos instruções a serem executadas caso o teste (condição) seja verdadeiro e outras instruções caso o teste resulte em falso.

# Instrução de selecção (decisão)

## ❑ Estrutura de decisão composta.

- Exemplo: crie um programa em Java que verifica se um número é par ou ímpar.

Pseudocódigo	Em Java
<pre>num:inteiro inicio   escreva("Digite um número")   leia(num)   se (num mod 2=0) entao     escreva("Número PAR")   senao     escreva("Número IMPAR")   fimse finalgoritmo</pre>	<pre>import java.util.Scanner; public class Exercícios {     public static void main(String[] args){         Scanner teclado=new Scanner(System.in);          System.out.println("Digite um número Inteiro");         int num = teclado.nextInt();          if(num % 2 == 0){             System.out.print("Número PAR");         }else{             System.out.print("Número ÍMPAR");         }     } }</pre>

# Instrução de selecção (decisão)

## ❑ Estrutura de decisão encadeada.

Pseudocódigo	Em Java
<pre>se(condição)então     se(condição)então         //Instruções     Senão         //Instruções fimse; Senão     //Instruções fimse;</pre>	<pre>if (condição) {     if (condição) {         // Instruções     } else{         // Instruções     } } else{     if (condição) {         // Instruções     } }</pre>

❑ **Nota:** Não existe uma regra para as estruturas encadeadas. Elas variam de acordo com a necessidade do problema proposto.

# Instrução de selecção (decisão)

## ❑ Estrutura de decisão switch.

- Java tem um comando interno de selecção múltipla **switch**.
- O **switch** testa sucessivamente o valor de uma expressão contra uma lista de constantes inteiras ou de caratere.
- Quando o valor coincide, os comandos associados àquela constante são executados



# Instrução de selecção (decisão)

## ❑ Estrutura de decisão switch

A forma geral do comando switch é:

```
switch (valor) {  
    case valor1:  
        instruções;  
        break ;  
    case valor2:  
        instruções;  
        break ;  
    case valor3:  
        instruções;  
        break ;  
    ...  
    default:  
        instruções ;  
}
```

# Instrução de selecção (decisão)

## ❑ Estrutura de decisão switch.

- O **default** é opcional e, se não estiver presente, nenhuma acção será realizada se todos os testes falharem.
- Para obter mais eficiência devemos limitar o número de comandos case a uma quantidade menor.
- O comando *break* é um dos comandos de desvio em java.

# Cap. III - Exercício

1. Crie um programa que recebe o salário de um trabalhador e o valor da prestação de um empréstimo, se a prestação for maior que 20% do salário imprimir: Empréstimo não concedido, caso contrário imprimir: Empréstimo concedido.
2. Escrever um programa que solicite um salário ao utilizador e mostre o imposto a pagar.
  - Se o salário for negativo ou zero mostre o erro respectivo.
  - Se o salário for maior que 1000 paga 10% de imposto, senão paga apenas 5%.
3. O sistema de avaliação de determinada disciplina, é composto por três provas. A primeira prova tem peso 2, a segunda tem peso 3 e a terceira tem peso 5. Faça um programa em java para calcular a média final de um aluno desta disciplina.

# Até a próxima Aula

