

R

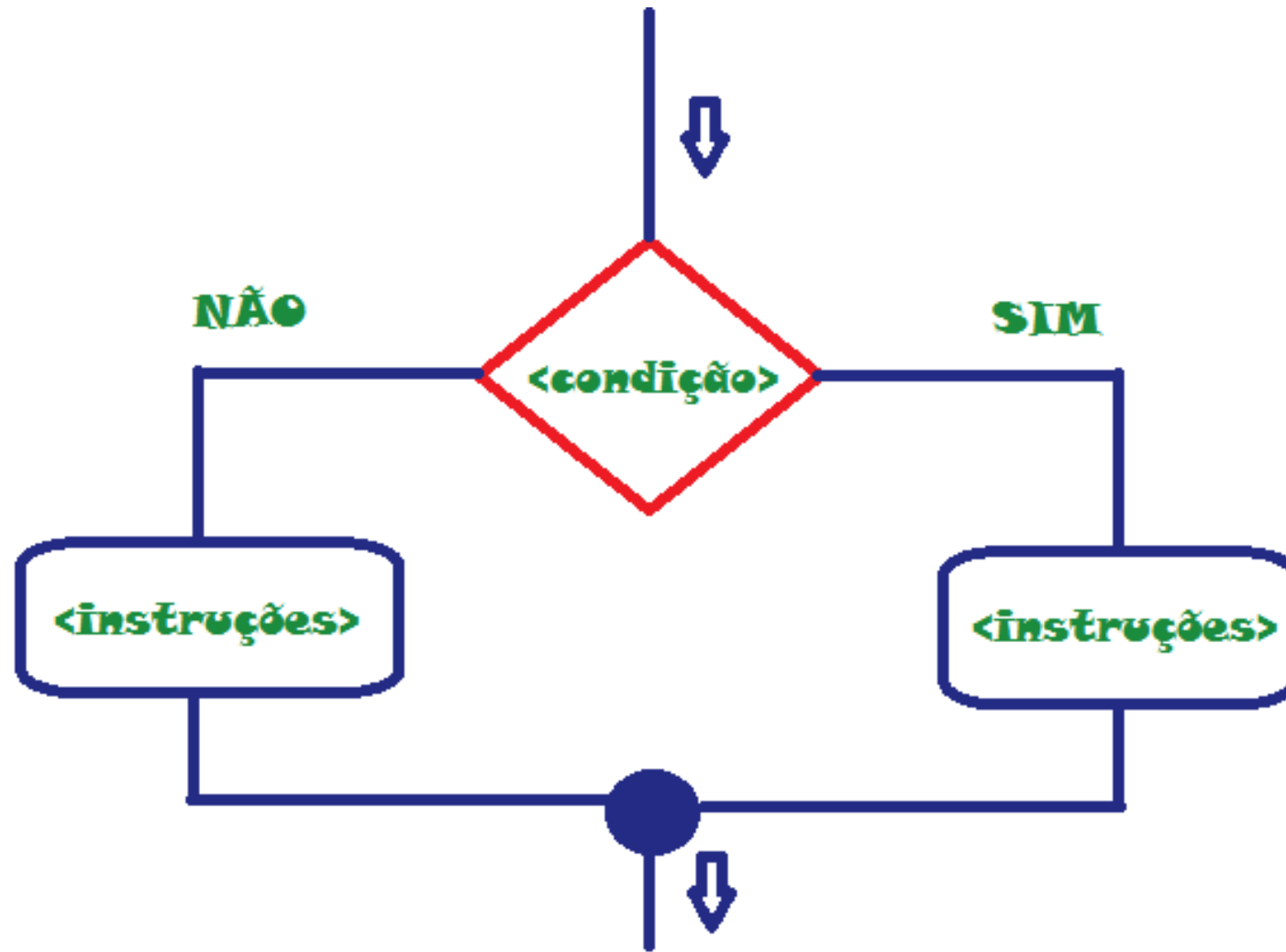
Java
Fundamentos e
Orientação a
Objetos

Java – Aula II

Java
Fundamentos e
Orientação a
Objetos

Estrutura de Decisão

Estrutura de Decisão



Estrutura de Decisão

A maioria dos algoritmos precisam tomar decisões ao longo de sua execução. As principais estruturas de decisão são:

IF...ELSE

IF...ELSE...IF

OPERADOR TERNÁRIO

SWITCH CASE



Operadores Relacionais

Serão utilizados nas estruturas de decisão fazendo a comparação entre valores/informações em uma condição.

Símbolo	Função
<>	Diferente
>	Maior
<	Menor
<=	Menor igual
>=	Maior igual
==	Igual
%	Resto da divisão entre números inteiros
x = (a > b) ? a:b	Ternário

Operadores Lógicos

Serão utilizados nas estruturas de decisão junto com os operadores relacionais para tratar mais de uma condição.

Símbolo	Função
&&	And (e)
	Or (ou)

Estrutura de Decisão

IF

IF <condição lógica>
 <ações>

Significado: Se a <operação lógica> resultar em verdadeiro, então executar as <ações>.

Caso contrário, simplesmente ignorar as <ações> e seguir para a próxima instrução no algoritmo.

Porque usar? Usada para decidir se um conjunto de ações opcionais deve ser executado ou não, dependendo do valor de algum dado ou de algum resultado que já tenha sido calculado no algoritmo. O valor do dado ou do resultado anterior será testado na operação lógica.



Exemplo:

Fazer um programa para verificar se o número é maior que 16.

```
public static void main(String[] args) {  
    int resposta = 19;  
  
    if (resposta > 16)  
        System.out.println("Número maior que 16!");  
}
```



Estrutura de Decisão

IF ELSE

```
IF <condição lógica>  
    <ações_1>  
ELSE  
    <ações_2>
```

Significado: Se a <operação lógica> resultar em verdadeiro, então executar as <ações_1>.

Senão, executar as <ações_2> e seguir para a próxima instrução no algoritmo.

Porque usar? Usada para decidir se um conjunto de ações opcionais deve ser executado ou não, dependendo do valor de algum dado ou de algum resultado que já tenha sido calculado no algoritmo. O valor do dado ou do resultado anterior será testado na operação lógica.



Exemplo:

Fazer um programa para verificar se o número é igual a 10.

```
public static void main(String[] args) {  
    int resposta = 10;  
  
    if (resposta == 10)  
        System.out.println("Você acertou!");  
    else  
        System.out.println("Você errou!");  
}
```



Estrutura de Decisão

IF ELSE IF

```
IF <condição lógica_1>  
    <ações_1>  
ELSE IF <condição lógica_2>  
    <ações_2>
```

Significado: Se a <operação lógica_1> resultar em verdadeiro, então executar as <ações_1>.

Senão, verificar a <operação lógica_2> e executar as <ações_2> e seguir para a próxima instrução no algoritmo.

Porque usar? Usada para decidir se um conjunto de ações opcionais deve ser executado ou não, dependendo do valor de algum dado ou de algum resultado que já tenha sido calculado no algoritmo. O valor do dado ou do resultado anterior será testado na operação lógica.



Exemplo:

Fazer um programa para verificar se o número é igual a 10 ou maior que 10.

```
public static void main(String[] args) {  
    int resposta = 10;  
  
    int resposta = 15;  
    if (resposta == 10)  
        System.out.println("A resposta é exatamente 10!");  
    else if (resposta > 10)  
        System.out.println("A resposta é maior que 10!");  
    else  
        System.out.println("A resposta é menor que 10!");  
}
```



Estrutura de Decisão

OPERADOR TERNÁRIO

O operador ternário é um recurso para tomada de decisões com objetivo similar ao do if/else, mas que é codificado em apenas uma linha.

Sintaxe:

(expressão booleana) ? código 1 : código 2;

Ao avaliar a expressão booleana, caso ela seja verdadeira, o código 1, declarado após o ponto de interrogação (?) será executado; do contrário, o programa irá executar o código 2, declarado após os dois pontos (:).



Exemplo:

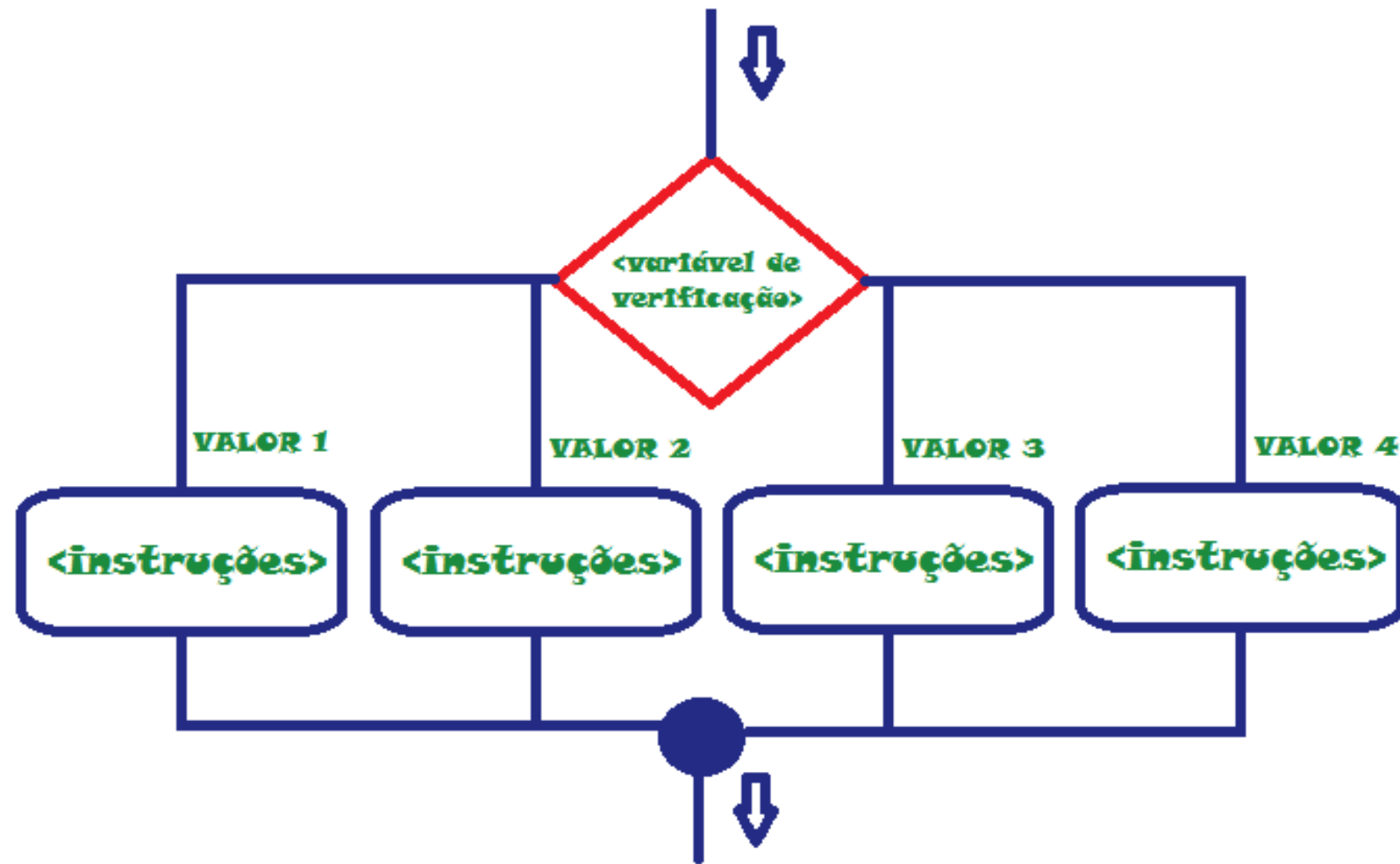
Fazer um programa para verificar se o número de dias informado faz parte da primeira ou segunda quinzena.

```
public static void main(String[] args) {  
    int numeroDias = 15;  
    System.out.println((numeroDias <= 15) ? "Primeira quinzena" :  
"Segunda quinzena"); // Resultado é Primeira Quinzena  
}
```

```
public class OperadorTernario {  
    public static void main (String args []) {  
        int a = 10, b = 8, r;  
        r = (a>b) ? a:b;  
        System.out.println (r); // Resultado é 10  
    }  
}
```



SWITCH CASE



SWITCH CASE

A estrutura condicional switch/case vem como alternativa em momentos em que temos que utilizar múltiplos ifs no código. Múltiplos if/else encadeados tendem a tornar o código muito extenso, pouco legível e com baixo índice de manutenção. O switch/case testa o valor contido em uma variável, realizando uma comparação com cada uma das opções. Cada uma dessas possíveis opções é delimitada pela instrução case.

Podemos ter quantos casos de análise forem necessários e, quando um dos valores corresponder ao da variável, o código do case correspondente será executado. Caso a variável não corresponda a nenhum dos casos testados, o último bloco será executado, chamado de default (padrão).

Sintaxe:

```
SWITCH <variável de verificação>  
CASE <valor1>:  
"instruções a serem executadas <valor1>";  
CASE <valor2>:  
"instruções a serem executadas <valor2>";  
CASE <valor3>:  
"instruções a serem executadas <valor3>";
```



Exemplo:

Fazer um programa para verificar o nome por extenso do valor escolhido.

```
public static void main(String[] args) {  
    int num = 2;  
    switch (num) {  
        case 1:  
            System.out.println("Um");  
            break;  
        case 2:  
            System.out.println("Dois");  
            break;  
        default:  
            System.out.println("Número inválido");  
            break;  
    }  
}
```





f i t @rederecode | y @recoderede

<https://recode.org.br>