

Algoritmos I

Ricardo Ribeiro Assink
ricardo.assink@unisul.br
ricardo@equipedigital.com

<http://www.ricardoassink.com.br/>

Algoritmos I - Ricardo Ribeiro Assink - ricardo.assink@unisul.br

1

Algoritmos I

Estruturas de Seleção

São estruturas que possibilitam a escolha de caminhos de acordo com a satisfação ou não de determinadas condições.

Algoritmos I - Ricardo Ribeiro Assink - ricardo.assink@unisul.br

2

Algoritmos I

Estruturas de Seleção

Há 3 tipos de estruturas de seleção:

- Seleção simples
- Seleção composta
- Seleção encadeada

Algoritmos I

Estruturas de Seleção – Seleção SIMPLES

Nesta estrutura uma única condição (expressão lógica) é avaliada podendo gerar um resultado verdadeiro ou falso.

Sintaxe:

```
se <condição> entao  
<sequência de comandos>  
fimse
```

Algoritmos I

Estruturas de Seleção – Seleção SIMPLES

Nesta estrutura uma única condição (expressão lógica) é avaliada podendo gerar um resultado verdadeiro ou falso.

Sintaxe:

se <condição> entao

<sequência de comandos>

Fimse

Onde: <condição> é uma expressão lógica que quando avaliada pode gerar um resultado falso ou verdadeiro. Se <condição> for verdadeira, a sequência de comandos então será executada; caso contrário, se <condição> for falsa, o comando é encerrado (fimse), sem executar nenhuma ação.

Algoritmos I

Estruturas de Seleção – Seleção SIMPLES

Exemplo: Solicite um número ao usuário, se o número for maior que 10, mostre a sua metade.

```
import javax.swing.JOptionPane;
public class Exemploif {
    public static void main(String[] args) {
        double numero, metade;
        numero = Double.parseDouble(JOptionPane.showInputDialog("Insira um número:
"));
        if ( numero > 10 ) {
            metade = numero / 2;
            JOptionPane.showMessageDialog(null, "A metade é: " + metade);
        }
    }
}
```

Algoritmos I

Estruturas de Seleção – Seleção COMPOSTA

O comando de seleção composta permite que seleções ou escolhas sejam realizadas dependendo da satisfação ou não de uma dada condição. Assim, para uma condição satisfeita (verdadeira), uma dada ação ou grupo de ações é realizado, para a não satisfação da condição (falsa), uma outra ação ou grupo de ações alternativo é realizado;

Algoritmos I

Estruturas de Seleção – Seleção COMPOSTA

Sintaxe:

se <condição> entao

 <sequência de comando>

senao

 <sequência de comandos alternativos>

fimse

Algoritmos I

Estruturas de Seleção – Seleção COMPOSTA

Exemplo: Solicite um número ao usuário, se o número for maior que 10, mostre a sua metade. Se não for maior que 10 mostre seu dobro.

```
import javax.swing.JOptionPane;
public class Exemploif {
    public static void main(String[] args) {
        double numero, metade, dobro;
        numero = Double.parseDouble(JOptionPane.showInputDialog("Insira um número: "));
        if ( numero > 10 ) {
            metade = numero / 2;
            JOptionPane.showMessageDialog(null, "A metade é: " + metade);
        } else {
            dobro = numero * 2;
            JOptionPane.showMessageDialog(null, "O dobro é: " + dobro);
        }
    }
}
```

Algoritmos I

Estruturas de Seleção – Seleção ENCADEADA

Quando, devido à necessidade de processamento, agrupamos várias seleções, formaremos uma seleção encadeada. Normalmente tal formação ocorre quando uma determinada ação ou bloco de ações deve ser executado se um grande conjunto de possibilidades ou combinações de situações for satisfatório.

Algoritmos I

Estruturas de Seleção – Seleção ENCADEADA

Sintaxe:

```
se <condição 1> entao
    se <condição 2> entao
        se <condição 3> entao
            <comandos>
        fimse
    fimse
fimse
```

Algoritmos I

Estruturas de Seleção – Seleção ENCADEADA

Exemplo: Solicitar um número ao usuário e verificar se o mesmo é divisível por 10, por 5, por 2 ou não é divisível por nenhum destes.

Algoritmos I

Estruturas de Seleção – Seleção ENCADEADA

```
import javax.swing.JOptionPane;
public class ExemploIfenc {
    public static void main(String[] args) {
        int numero;
        numero = Integer.parseInt(JOptionPane.showInputDialog("Insira um número: "));
        if( (numero % 10) == 0 ){
            JOptionPane.showMessageDialog(null, "É múltiplo de 10 !");
        }else{
            if( (numero % 2) == 0 ){
                JOptionPane.showMessageDialog(null, "É múltiplo de 2 !");
            }else{
                if( (numero % 5) == 0 ){
                    JOptionPane.showMessageDialog(null, "É múltiplo de 5 !");
                }else{
                    JOptionPane.showMessageDialog(null, "Não é múltiplo destes! ");
                }
            }
        }
    }
}
```

Algoritmos I – Ricardo Ribeiro Assink – ricardo.assink@unisul.br

13

Algoritmos I

Operadores Lógicos

```
> // maior
>= // maior ou igual
< // menor
<= // menor ou igual
== // igual
!= // diferente
|| // um OU outro
&& // um E outro
```

Algoritmos I – Ricardo Ribeiro Assink – ricardo.assink@unisul.br

14

Algoritmos I

Exercício 15

Faça um programa em JAVA que solicite 2 números inteiros e efetue a adição; caso o resultado seja maior que 10, apresentá-lo:

Algoritmos I

Exercício 16

Faça um programa em JAVA que solicite 1 número inteiro, verificar se o mesmo é múltiplo de 2 e mostrar a resposta mesmo que negativa:

Algoritmos I

Exercício 17

Faça um programa em JAVA que solicite 2 números inteiros, verificar qual é o número maior e mostrar na tela, ou se forem iguais, mostrar uma mensagem indicando esta informação:

Algoritmos I

Exercício 18

A prefeitura de Florianópolis abriu uma linha de crédito para os funcionários estatutários. O valor máximo da prestação não poderá ultrapassar 30% do salário bruto. Faça um algoritmo que permita entrar com o salário bruto e o valor da prestação e informar se o empréstimo pode ou não ser concedido.

Algoritmos I

Exercício 19

Solicitar o ano de nascimento de uma pessoa e o ano atual. Verifique se o ano de nascimento é válido, ou seja, se o ano de nascimento é menor que o ano atual, então mostre a idade desta pessoa.

Algoritmos I

Exercício 20

Solicitar nome e idade de uma pessoa.
Se a pessoa possuir idade entre 15 e 25 anos, imprimir o nome e a mensagem: “ACEITA”.
Caso contrário, imprimir o nome e a mensagem: “NÃO ACEITA”.

