

PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

Prof. Richard Henrique de Souza

Prof. Osmar De Oliveira Braz Junior

Material adaptado do Prof. Ricardo Ribeiro Assink





ESTRUTURAS DE SELEÇÃO

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



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

ESTRUTURAS DE SELEÇÃO

Há 3 tipos de estruturas de seleção



01 SELEÇÃO
SIMPLES

03 SELEÇÃO
ENCADEADA

02 SELEÇÃO
COMPOSTA

PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

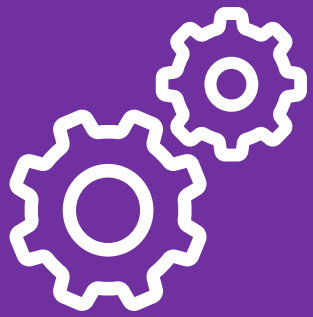
01 SELEÇÃO SIMPLES

EXEMPLO



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





PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS (JAVA) ESTRUTURAS DE SELEÇÃO

01 SELEÇÃO SIMPLES

```
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);
        }
    }
}
```

PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

02 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;

PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

02 SELEÇÃO COMPOSTA

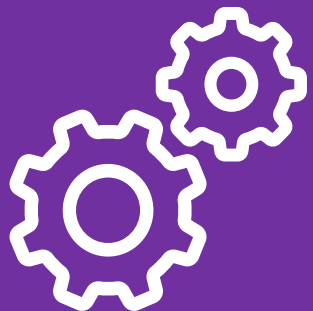


EXEMPLO



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.





PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

02 SELEÇÃO COMPOSTA

```
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);
        }
    }
}
```

PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

03 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.

PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

03 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. Caso encontre ao menos um que é divisível, encerre a aplicação.





PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

03 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! ");  
                }  
            }  
        }  
    }  
}
```

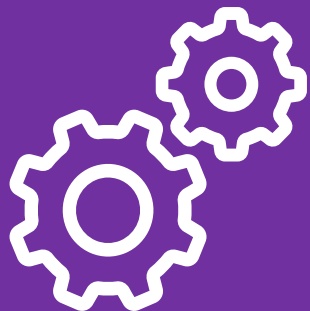
PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

03 SELEÇÃO SWITCH

A estrutura switch verifica uma variável e age de acordo com seus cases. Os cases são as possibilidades de resultados que são obtidos por switch.

Basicamente, o switch serve para controlar várias ações diferentes de acordo com o case definido dentro dele.



PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

03 SELEÇÃO SWITCH

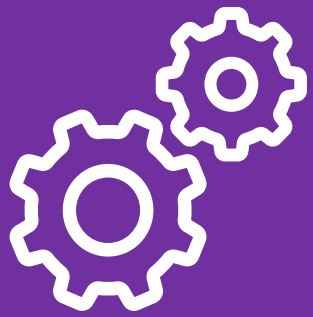
```
import javax.swing.JOptionPane;
public class ExemploSwitch{
    public static void main(String[] args) {
        int controle;
        int resultado, numero1 = 10, numero2 = 5;

        controle = Integer.parseInt(JOptionPane.showInputDialog("Qual operação você quer executar ?"
                                                                + " \n 1 - adição \n 2 - subtração"));

        switch (controle){
            case 1:
                resultado = numero1 + numero2;
                JOptionPane.showMessageDialog(null,"O resultado é: " + resultado);
                break;

            case 2:
                resultado = numero1 - numero2;
                JOptionPane.showMessageDialog(null,"O resultado é: " + resultado);
                break;

            default:
                JOptionPane.showMessageDialog(null,"Opção inválida! ");
                break;
        }
    }
}
```



PROGRAMAÇÃO DE SOLUÇÕES COMPUTACIONAIS

(JAVA) ESTRUTURAS DE SELEÇÃO

03 SELEÇÃO SWITCH

```
import javax.swing.JOptionPane;
public class Exemploswitch2 {
    public static void main(String[] args) {
        String controle;
        controle = JOptionPane.showInputDialog("Você concorda com os termos apresentados ? (s/n)");
        char controleCHAR = controle.charAt(0);

        switch(controleCHAR){
            case 's':
                JOptionPane.showMessageDialog(null,"SIM, você concorda ! ");
                break;

            case 'n':
                JOptionPane.showMessageDialog(null,"NÃO, você não concorda ! ");
                break;

            default:
                JOptionPane.showMessageDialog(null,"Opção inválida! ");
                break;
        }
    }
}
```

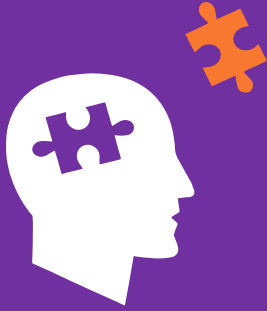


(JAVA) EXERCÍCIO 10



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



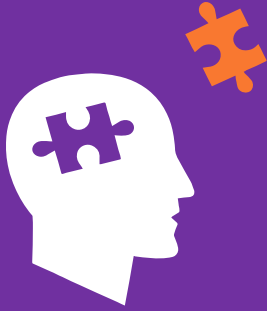


(JAVA) EXERCÍCIO 11



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:





(JAVA) EXERCÍCIO 12



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.



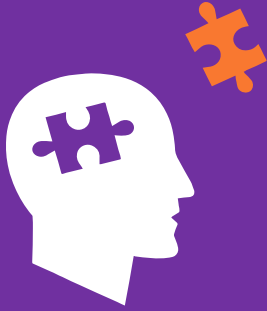


(JAVA) EXERCÍCIO 12

```
import javax.swing.JOptionPane;
public class Exercicio12 {

    public static void main(String[] args) {
        int a,b;
        a = Integer.parseInt(JOptionPane.showInputDialog("Digite o valor de a:"));
        b = Integer.parseInt(JOptionPane.showInputDialog("Digite o valor de b:"));

        if(a > b){
            JOptionPane.showMessageDialog(null,"O número maior é: " + a);
        }else{
            if(a == b){
                JOptionPane.showMessageDialog(null,"Os números são iguais ");
            }else{
                JOptionPane.showMessageDialog(null,"O número maior é: " + b);
            }
        }
    }
}
```

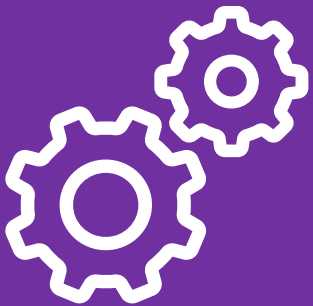


(JAVA) EXERCÍCIO 13



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.



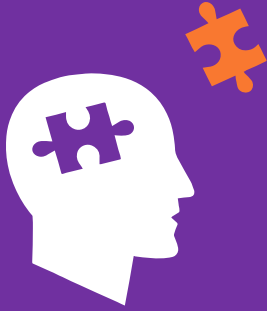


(JAVA) EXERCÍCIO 13

```
import javax.swing.JOptionPane;
public class Exercicio13 {
    public static void main(String[] args) {
        double sb, vp, lp;
        sb = Double.parseDouble(JOptionPane.showInputDialog("Digite o Salário Bruto: "));
        vp = Double.parseDouble(JOptionPane.showInputDialog("Digite o Valor da prestação: "));

        lp = sb * 0.3;

        if(vp <= lp){
            JOptionPane.showMessageDialog(null, "Empréstimo Concedido! ");
        }else{
            JOptionPane.showMessageDialog(null, "Empréstimo NEGADO! ");
        }
    }
}
```

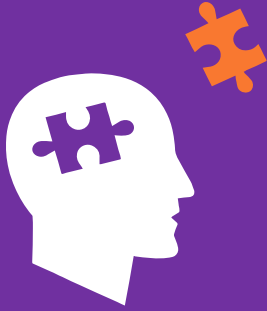


(JAVA) EXERCÍCIO 14



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.





(JAVA) EXERCÍCIO 15



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”.





(JAVA) EXERCÍCIO 16

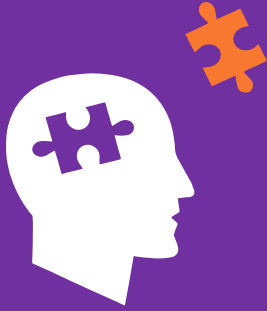


Implemente uma calculadora,
primeiro peça 2 números ao usuário (um de cada vez)
e apresente as seguintes opções:

- adição
- multiplicação
- subtração
- divisão

Mostre o resultado após a operação.





Implemente um algoritmo em JAVA que escreva a seguinte mensagem na tela, substituindo a profissão pela informada pelo usuário:

"Tibúrcio é <profissão>"

Solicite ao usuário que informe a profissão do Tibúrcio, apresente as seguintes opções:

e engenheiro

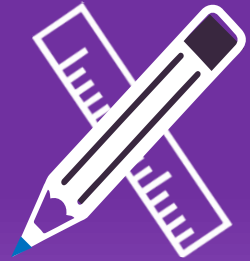
m médico

d designer

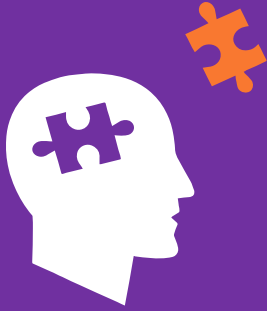
p programador

a advogado

Se informada opção diferente das apresentadas a profissão é **DESEMPREGADO**.



(JAVA) EXERCÍCIO 17

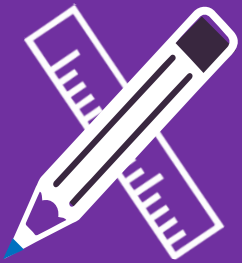


Uma loja dará um desconto em seus produtos de acordo com a categoria de cada produto.

As categorias e os descontos são mostrados abaixo:

A	B	C	D	E
10%	15%	20%	25%	50%

Faça um algoritmo para ler o nome do cliente, o preço e a categoria do produto que o cliente está comprando. Após a leitura mostrar na tela o preço final que o cliente deverá pagar após ter sido dado o desconto no preço da mercadoria.



Busca Ativa!

- 1 Implemente, compile e execute todos os exemplos dados em sala de aula.
- 2 CRIE um exemplo de código utilizando IF. Implemente, compile e execute.



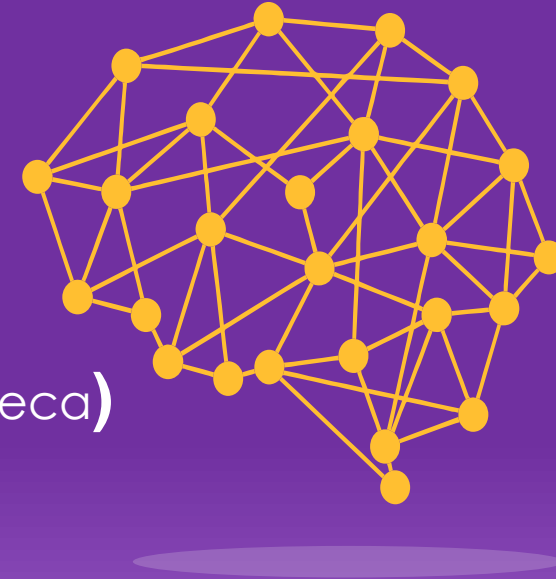
Busca Ativa!

- **Oracle Academy**

JAVA Foundations ==> Seção 5

JAVA Fundamentals ==> Seção 5.1

- **Livro: Java para iniciantes** (Ulife → Biblioteca online → Minha Biblioteca)
Leia os capítulos 1,2 e 3



FIM

