

# Introdução à Orientação a Objetos I

Rafael Silva Guimarães

Instituto Federal do Espírito Santo

*[rafaelg@ifes.edu.br](mailto:rafaelg@ifes.edu.br)*

*<http://rafaelguimaraes.net>*

*<https://github.com/rafaelsilvag/ifesJava/>*

30 de Julho de 2015



INSTITUTO FEDERAL  
ESPÍRITO SANTO

Estruturas de Decisão

Estruturas de Repetição

Referências



- ▶ Uma estrutura de decisão altera o fluxo de execução do programa. Ele faz isso através de uma estrutura aonde você especifica uma condição para que ocorra tal alteração em um determinado programa.
- ▶ No Java, as estruturas de decisão possuem syntaxe idêntica da linguagem C, vista por vocês no semestre anterior. As estruturas que você deve ter conhecimento são: *IF* e *SWITCH*.



- ▶ Vejamos um de utilização da estrutura *IF* no algoritmo Exemplof01.
- ▶ **Obs.: Assim como no C, a linguagem Java também é sensível a letras maiúsculas e minúsculas.**



## Example (Exempl01f01.java)

```
import javax.swing.JOptionPane;  
// Importação da classe responsável por criar uma  
// interface de interação com o usuário  
public class ExemploIf01{  
    public static void main(String args[]){  
        int valor=0;  
        // Retiramos o System.out.println por JOptionPane  
        if (valor == 0){  
            JOptionPane.showMessageDialog(  
                null,  
                "Valor igual a 0");  
        }else{  
            JOptionPane.showMessageDialog(  
                null,  
                "Valor não é igual a 0");  
        }  
    }  
}
```



- ▶ Estamos alterando a forma de como iremos exibir e capturar as informações para o usuário. Estaremos utilizando uma classe gráfica que exibe e coleta informações do usuário chamada de "*JOptionPane*".
- ▶ Esta classe faz parte do pacote *Swing*, responsável por armazenar todas as classes de manipulação gráfica no Java. Iremos aprofundar melhor nela em um outro momento em nossa disciplina.



- ▶ Lembre-se! Para capturar e exibir os dados utilizaremos conforme podemos observar no código abaixo:
- ▶ `JOptionPane.showMessageDialog(null, "Valor");`

```
int i=0;
// Para concatenar valor inteiro com string usamos o +
OptionPane.showMessageDialog(null, "Valor: "+i);
```

- ```
▶ JOptionPane.showInputDialog("Informe um valor: ");
```

```
String str01;  
// Sempre irá retornar um valor do tipo String  
str01 = JOptionPane.showInputDialog("Informe um valor: ");
```



- ▶ Algumas comparações serão alteradas. Por exemplo, em C para compararmos strings era necessário utilizarmos uma coleção de funções na biblioteca "string.h". No caso do Java, uma String na verdade é um objetos que contém métodos e atributos. Veja no exemplo a seguir:
- ▶ Podemos observar que no objeto do tipo String temos um método chamado de "*equal*" ou "*equalsIgnoreCase*", responsável por comparar 2 strings. Portanto, o método irá retornar 2 valores, "true" ou "false".





## Example (ExemplolIf04.java)

```
import javax.swing.JOptionPane;

public class ExemploIf04{
    public static void main(String args[]){
        String nome;

        nome = JOptionPane.showInputDialog("Informe o seu nome");
        // verifica se o nome do usuário é admin
        if(nome.equals("admin")){
            JOptionPane.showMessageDialog(null,
   "Você é um usuário Admin!");
        }else{
            JOptionPane.showMessageDialog(null,
   "Você é um usuário comum!");
        }
    }
}
```



## Example (Exempl0f05.java)

```
import javax.swing.JOptionPane;

public class ExemploIf05{
    public static void main(String args[]){
        String nome;

        nome = JOptionPane.showInputDialog("Informe o seu nome");
        // verifica se o nome do usuário é admin
        if(nome.equalsIgnoreCase("admin")){
            JOptionPane.showMessageDialog(null,
  "Você é um usuário Admin!");
        }else{
            JOptionPane.showMessageDialog(null,
  "Você é um usuário comum!");
        }
    }
}
```



- ▶ Vamos ver um exemplo de utilização da estrutura de decisão Switch em Java, no algoritmo ExemploSwitch01.
- ▶ Nesta etapa executamos uma "Conversão de Tipos", passando um valor do tipo String para um valor do tipo Integer.

| Valor Inicial  | Tipagem Final | Formato para conversão |
|----------------|---------------|------------------------|
| int x = 10     | float         | float y = (float) x    |
| int x = 10     | double        | double y = (double) x  |
| float x = 10.5 | int           | int y = (int) x        |

# Estruturas de Decisão

```
import javax.swing.JOptionPane;

public class ExemploSwitch01{
    public static void main(String args[]){
        String valor;
        int opc;

        valor = JOptionPane.showInputDialog("Informe um valor: ");
        // Convertendo os valor String para Integer.
        opc = Integer.parseInt(valor);

        switch(opc){
            case 1: JOptionPane.showMessageDialog("Valor 1!"); break;
            case 2: JOptionPane.showMessageDialog("Valor 2!"); break;
            case 3: JOptionPane.showMessageDialog("Valor 3!"); break;
            default: JOptionPane.showMessageDialog("Desconhecido!");
        }
    }
}
```



| Valor Inicial      | Tipagem Final  | Formato para conversão           |
|--------------------|----------------|----------------------------------|
| String x = "10"    | int            | int y = Integer.parseInt(x)      |
| String x = "10.50" | float          | float y = Float.parseFloat(x)    |
| String x = "10.50" | double         | double y = Double.parseDouble(x) |
| String x = "IFES"  | Vetor de Bytes | byte b[] = x.getBytes()          |
| int x = 10         | String         | String y = String.valueOf(x)     |
| float x = 10.5     | String         | String y = String.valueOf(x)     |
| double x = 235.222 | String         | String y = String.valueOf(x)     |
| byte x[]           | String         | String y = new String(x)         |



# Estruturas de Repetição

- ▶ Uma estrutura de repetição, cada ciclo de repetição é chamado de iteração.
- ▶ No Java, as estruturas de repetição possuem syntaxe idêntica da linguagem C, vista por vocês no semestre anterior. As estruturas que você deve ter conhecimento são: *FOR*, *FOREACH*, *WHILE* e *DO...WHILE*.



## Example (ExemploFor01.java)

```
import javax.swing.JOptionPane;

public class ExemploFor01{
    public static void main(String args[]){
        int valor=10, soma=0;
        // Exemplo de utilização do FOR
        for(int i = 0; i <= valor; i++){
            soma += i;
        }

        JOptionPane.showMessageDialog(null, "Resultado: "+soma);
    }
}
```



## Example (ExemploFor02.java)

```
import javax.swing.JOptionPane;

public class ExemploFor02{

    public static void main(String args[]){
        int vInteiros[] = {0,1,2,3,4,5,6,7,8,9}, soma=0;
        // Exemplo de utilização do FOREACH
        for(int v: vInteiros){
            soma += v;
        }

        JOptionPane.showMessageDialog(null,"Resultado: "+soma);
    }
}
```





## Example (ExemploFor02.java)

```
import javax.swing.JOptionPane;

public class ExemploWhile01{
    public static void main(String args[]){
        int valor=10, soma=0;
        // Exemplo de utilização do While
        while(valor >= 0){
            soma += valor;
            valor--;
        }

        JOptionPane.showMessageDialog(null,"Resultado: "+soma);
    }
}
```



# Estruturas de Repetição - DO...WHILE

```
import javax.swing.JOptionPane;

public class ExemploWhile02{
    public static void main(String args[]){
        String aux;
        int opc;
        // Exemplo de utilização do Do...While
        do{
            aux = JOptionPane.showInputDialog(
                "##### INFORME UMA OPÇÃO #####\n"+
                "1 - Cadastrar Aluno\n"+
                "2 - Editar Aluno\n"+
                "3 - Visualizar Alunos\n"+
                "0 - Sair\n"+
                "#####\n"
            );
        }while(opc != 0);
    }
}
```





Cornell, G. ; Horstmann, S. C.

Core Java 2: Fundamentos (vol.1.)

*Pearson Makron Books São Paulo*



Cornell, G. ; Horstmann, S. C.

Core Java 2: Recursos Avançados (vol.2.)

*Pearson Makron Books São Paulo*



Sebesta, R.W.

Conceitos de Linguagens de Programação

*Bookman São Paulo*



Deitel, Paul; Deitel, Harvey

Java - Como Programar

*Pearson São Paulo*

