

## ***Tópico 2 – Técnica de Programação I***

- ***Caixas de Mensagens;***
- ***Conversão de Tipos de dados***
- ***Constantes***
- ***Desvio Condicional***

## *Inclusão de Caixa de Mensagem*

A inclusão de mensagem é realizada por meio da classe **JOptionPane**, especializada na geração de caixas de diálogo, isto é, pequenas janelas que fazem uma pergunta, avisam o usuário ou fornecem uma mensagem qualquer.

Essas janelas representam um modo de comunicação com o usuário, possibilitando fornecer uma simples informação ou obter resposta.

A classe **JOptionPane** oferece vários métodos para a criação de caixas de diálogo de acordo com a necessidade.

Uma característica particular da classe **JOptionPane** é a não-necessidade de declaração de objetos para a criação das caixas, que são criadas diretamente pelos métodos da classe.

Existem 4 tipos de padrão de caixas de diálogo:

**MessageDialog** -> Caixa de diálogo que apresenta uma mensagem possibilitando acrescentar ícones de alerta ao usuário.

**ConfirmDialog** -> Caixa de dialogo que além de emitir mensagem, possibilita o usuário a responder uma pergunta.

**InputDialog** -> Caixa de dialogo que além de emitir uma mensagem, permite a entrada de um texto.

**Option Dialog**: Caixa de diálogo que abrange os 3 tipos anteriores

## **MessageDialog**

A caixa de diálogo **MessageDialog** é utilizada para mostrar ao usuário alguma informação e não retorna nenhum tipo de resposta. Essas caixas, podem representar um ícone de acordo com o tipo da mensagem (pergunta, informação, alerta, erro ou definido pelo usuário). Sua sintaxe é:

**JOptionPane.showMessageDialog** (Component, <mensagem>, <titulo da mensagem>, <tipo da mensagem>)

**Onde:**

**Component:** permite definir a posição na tela da caixa de mensagem.

**Tipo da Mensagem:** É o ícone que representa o tipo da mensagem que está sendo informado ao usuário. Esse ícone é definido pelo usuário e pode ser um gif animado dando mais vida a aplicação.

## Tipos de Ícone






Ícone	Comando
 Pergunta	QUESTION_MESSAGE
 Informação	INFORMATION_MESSAGE
 Alerta	WARNING_MESSAGE
 Erro	ERROR_MESSAGE
 Definido pelo usuário	INFORMATION_MESSAGE, <ícone>
Vazio (Somente mensagem)	PLAIN_MESSAGE

Tabela 9.10 - Tipos de mensagens/ícones da caixa de diálogo.

## ConfirmDialog

A caixa de confirmação são pequenas janelas que possibilitam ao usuário responder algum questionamento pelos botões Yes, No e Cancel. Essas janelas são formadas pelo uso do método **showConfirmDialog**.

Uma vez apresentada na tela, o usuário escolhe uma das opções e, dependendo do botão clicado, é gerado um valor inteiro pertencente à classe JOptionPane:

YES\_OPTION=0, NO\_OPTION=1, CANCEL\_OPTION=3.

**Sintaxe:**

`int resposta=JOptionPane.ShowConfirmDialog(component, <message>,<título da mensagem>,< botões presentes> , <tipo da mensagem> .`

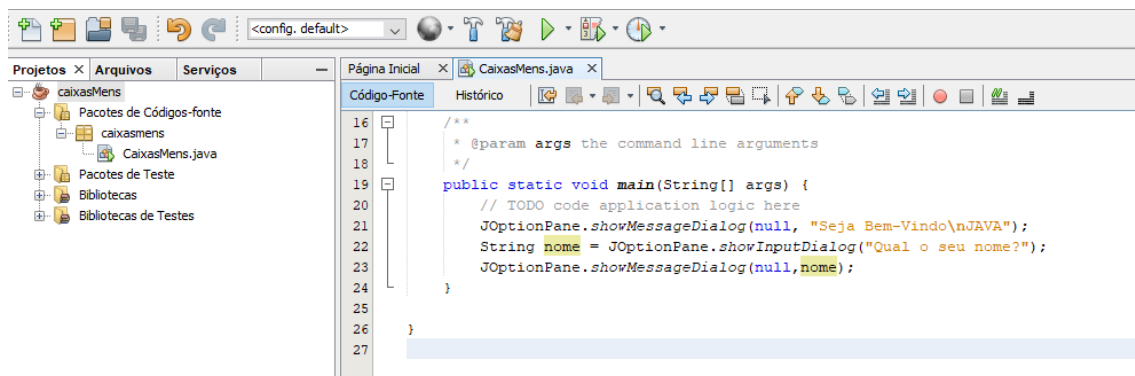
**InputDialog**

As caixas de diálogo de entrada são geradas a partir do método `showInputDialog()` e são utilizadas para fazer perguntas ao usuário e solicitar a entrada de um texto.

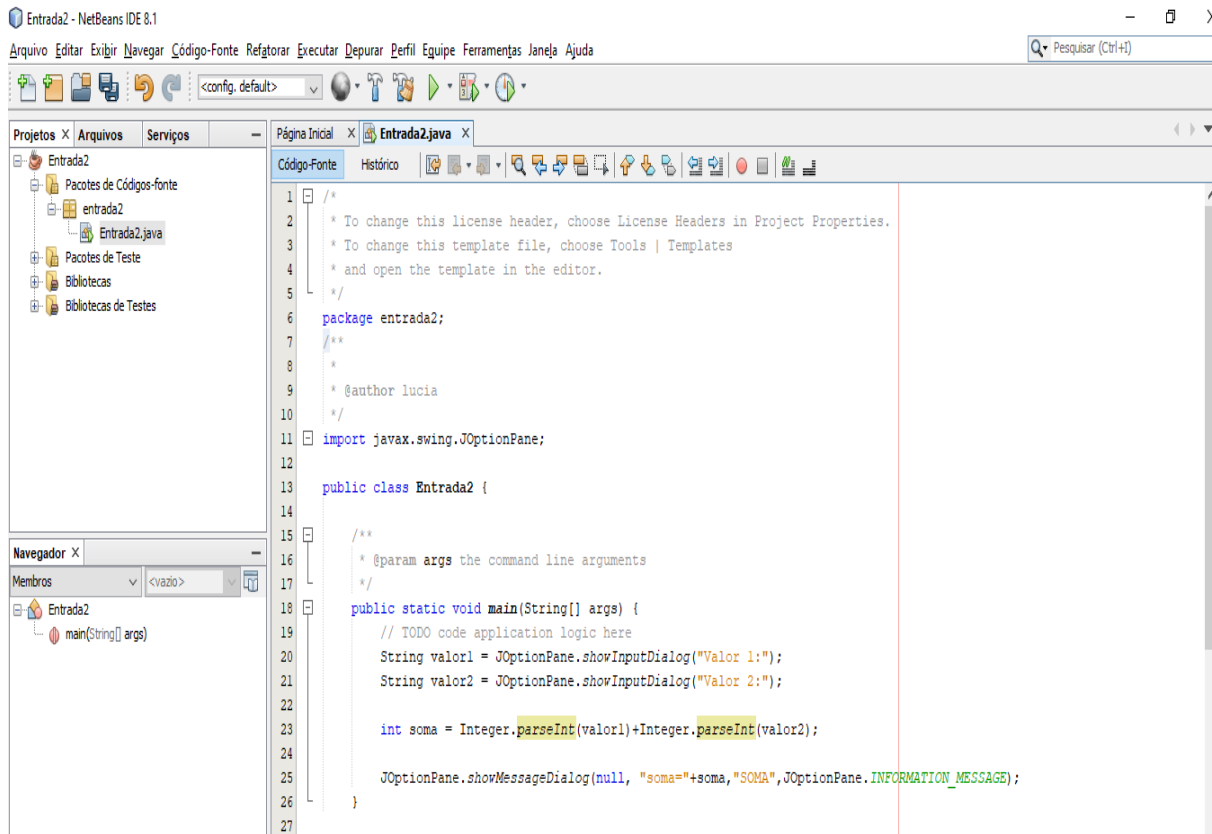
Esse texto será armazenado em uma variável qualquer do tipo `String`. Essas caixas de diálogo são chamadas pelo método `showInputDialog()`. A sintaxe usada na criação de caixas de diálogo de entrada é praticamente a mesma usada nas caixas de mensagem, possuindo o formato seguinte:

`String resp=JOptionPane.showInputDialog(null,<mensagem>,<título da mensagem>,<tipo da mensagem>)`

- Exemplo – Entrada de Dados**



## • Exemplo 2 – Entrada de Dados



## Option Dialog

As caixas de dialogo de opções são geradas a partir do método `OptionDialog` e são bem complexas. Ela e capaz de combinar todos os recursos já vistos nas outras caixas de diálogo. Sua sintaxe é a seguinte:

`int resposta=(null,<mensagem>,<título da mensagem>,<botões presentes>,<tipo de mensagem>,<icone>,<array de objetos>,<seleção padrão>)`

Em que:

**botões presentes:** um tipo de caixa que utiliza a variáveis `YES_NO_OPTION`, `YES_NO_CANCEL_OPTION` ou `OK_CANCEL_OPTION`.

Deve-se utilizar o zero caso outros botões sejam utilizados em lugar destes.

**Tipo de mensagem** -> uma das variáveis de pergunta, informação, alerta, erro ou definida pelo usuário.

**Ícone**->Um objeto ImageIcon a ser apresentada em lugar de um dos ícones do argumento anterior (<tipo de mensagem>). Se nenhum ícone for usado deve-se colocar a palavra null no lugar desse argumento

**array de objetos** -> é um array de objetos que contém os componentes ou outros objetos que representam as escolhas na caixa de diálogo, caso YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION ou OK\_CANCEL\_OPTION não esteja sendo usado.

**Seleção padrão:** o objeto representa a seleção padrão, caso a opção YES\_NO\_OPTION, OK\_CANCEL\_OPTION ou YES\_NO\_CANCEL\_OPTION não estejam sendo usados.

## Conversão de tipos de dados

Para converter tipos de dados o Java dispõe de métodos específicos.

Convertendo uma String em Número:

String teste = "10"

- Int ..... Integer.parseInt(teste)
- Float..... Float.parseFloat(teste)
- Double..... Double.parseDouble(teste)

Caso seja necessário converter Tipos dados diversos em String, também é possível utilizando o método valueOf da classe String.

A conversão entre tipos de dados numéricos se dá pelo operador de cast.

### Exemplo:

Convertendo Inteiro em Double:

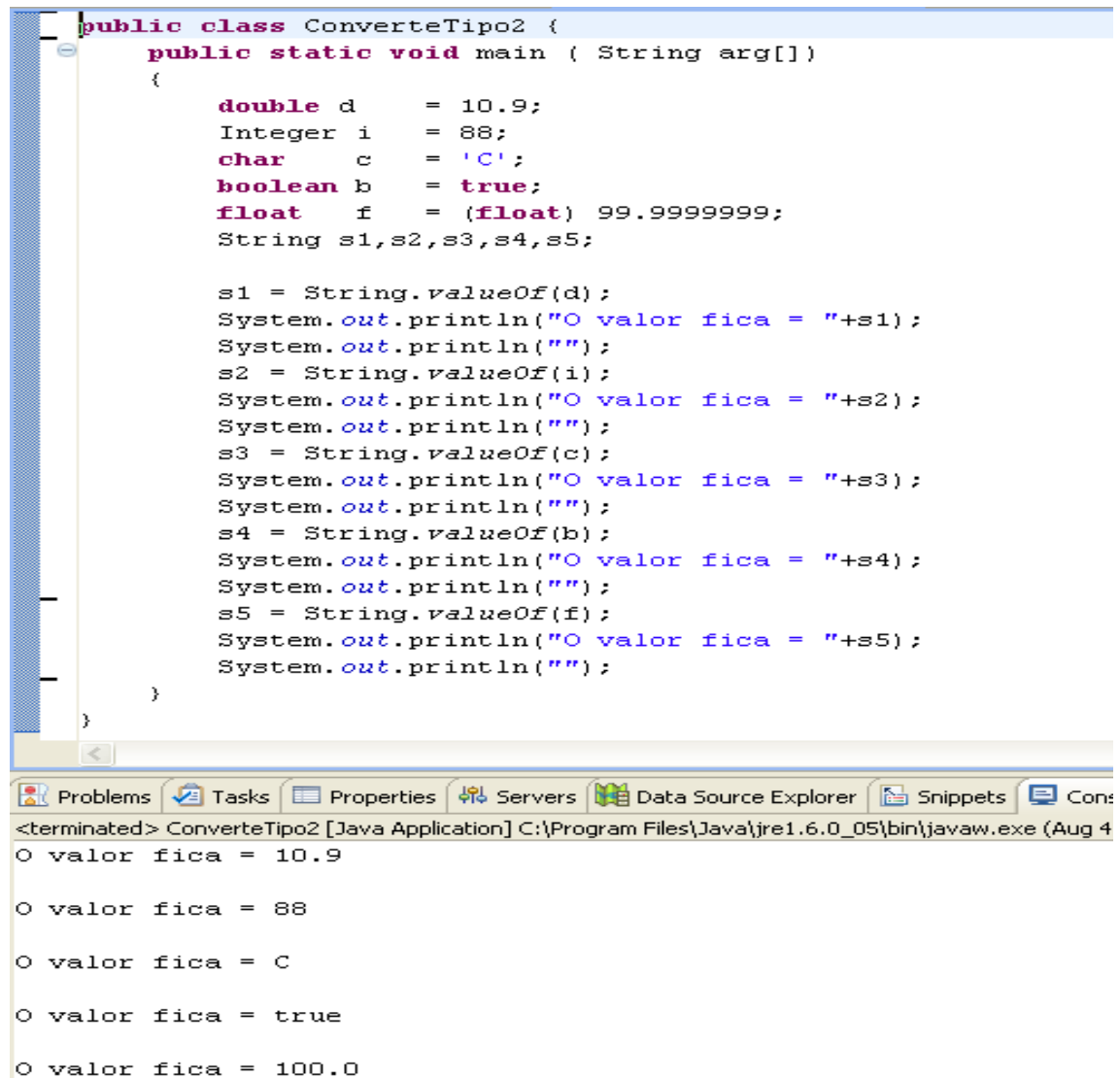
```
int i = 10;
```

```
double d = (double) i;
```

Convertendo double em float:

```
float f = (float) d;
```

### Exemplo:



```

public class ConverteTipo2 {
    public static void main ( String arg[])
    {
        double d      = 10.9;
        Integer i      = 88;
        char   c       = 'C';
        boolean b      = true;
        float   f       = (float) 99.9999999;
        String s1,s2,s3,s4,s5;

        s1 = String.valueOf(d);
        System.out.println("O valor fica = "+s1);
        System.out.println("");
        s2 = String.valueOf(i);
        System.out.println("O valor fica = "+s2);
        System.out.println("");
        s3 = String.valueOf(c);
        System.out.println("O valor fica = "+s3);
        System.out.println("");
        s4 = String.valueOf(b);
        System.out.println("O valor fica = "+s4);
        System.out.println("");
        s5 = String.valueOf(f);
        System.out.println("O valor fica = "+s5);
        System.out.println("");
    }
}

```

Problems Tasks Properties Servers Data Source Explorer Snippets Cons
 <terminated> ConverteTipo2 [Java Application] C:\Program Files\Java\jre1.6.0\_05\bin\javaw.exe (Aug 4
 O valor fica = 10.9
 O valor fica = 88
 O valor fica = C
 O valor fica = true
 O valor fica = 100.0

## Constantes

Constantes são elementos de uma classe cujo valor não pode ser alterado.

Uma constante pode ser definida em qualquer tipo de dados suportado pelo java.

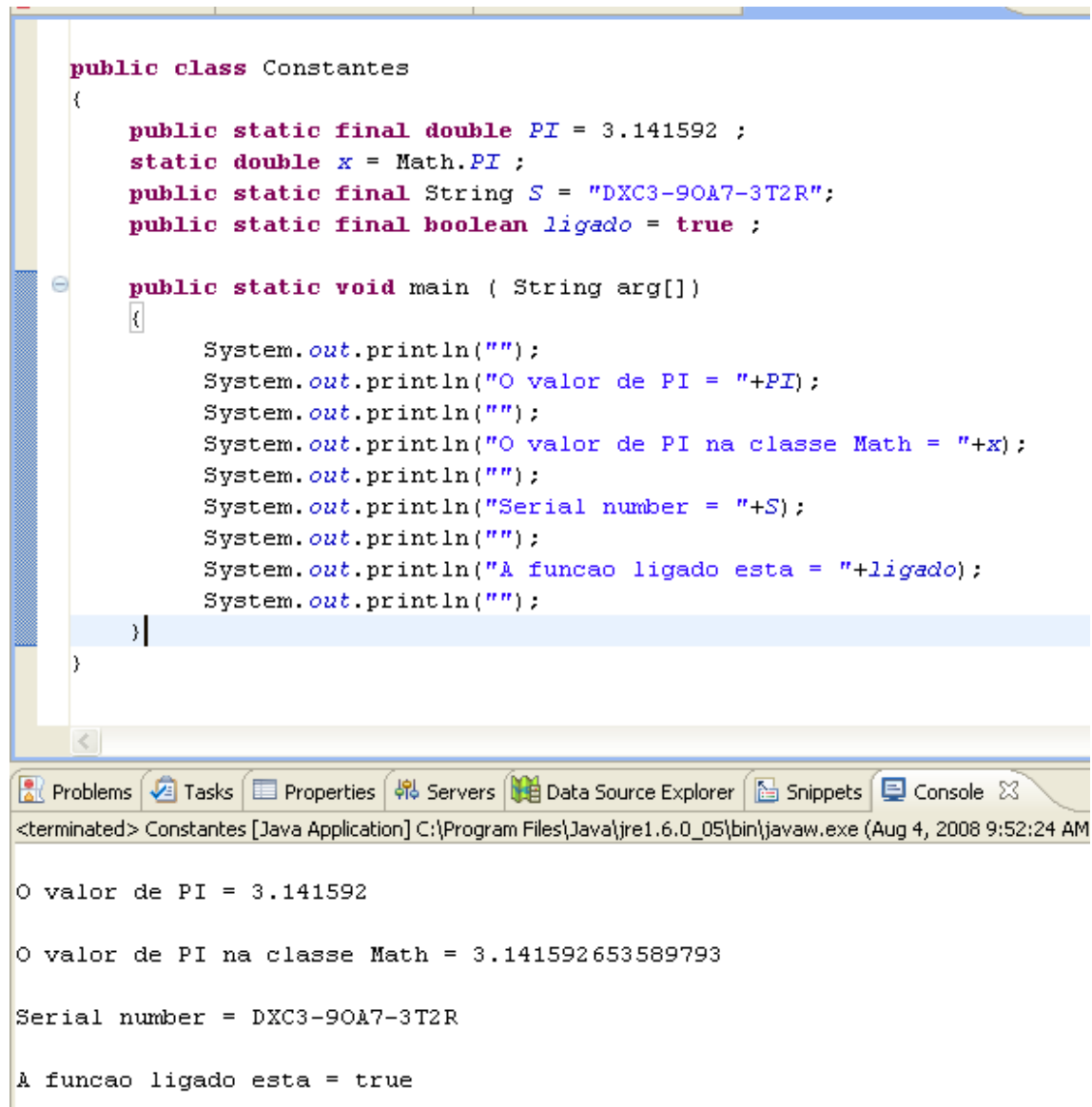
Para definir uma constante declare da seguinte forma:

```
public static final <TipoDado> <NomeConstante> = <valorConstante>
```

Esta declaração deve ficar antes do método Main

Caso a palavra final seja retirada a constante passa a ser uma variável global.

Exemplo: Definindo constantes.



```
public class Constantes
{
    public static final double PI = 3.141592 ;
    static double x = Math.PI ;
    public static final String S = "DXC3-90A7-3T2R";
    public static final boolean ligado = true ;

    public static void main ( String arg[])
    {
        System.out.println("");
        System.out.println("O valor de PI = "+PI);
        System.out.println("");
        System.out.println("O valor de PI na classe Math = "+x);
        System.out.println("");
        System.out.println("Serial number = "+S);
        System.out.println("");
        System.out.println("A funcao ligado esta = "+ligado);
        System.out.println("");
    }
}
```

<terminated> Constantes [Java Application] C:\Program Files\Java\jre1.6.0\_05\bin\javaw.exe (Aug 4, 2008 9:52:24 AM)

O valor de PI = 3.141592

O valor de PI na classe Math = 3.141592653589793

Serial number = DXC3-90A7-3T2R

A funcao ligado esta = true

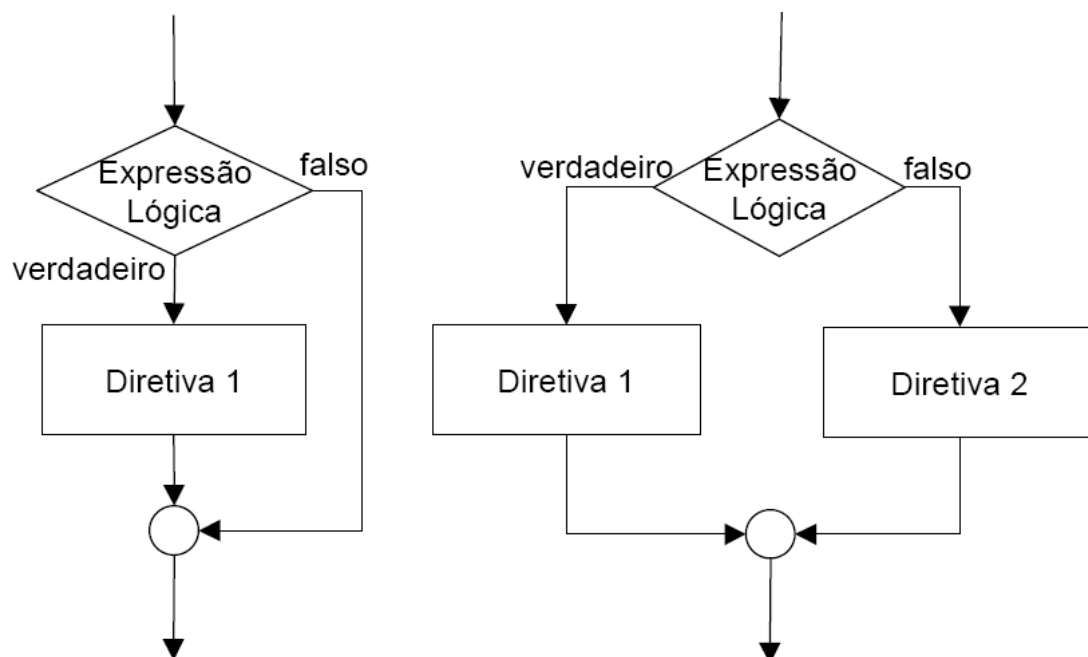


## ***Desvio Condicional - If***

### **Sintaxe:**

```
if ( comparação )  
{  
    <Seqüência de Instruções>  
}  
  
else  
{  
    <Seqüência de Instruções>  
}
```

### ***Comportamento do desvio condicional if***



```

import javax.swing.*;
public class Exemplo_if_2
{
    static double x , y ;
    public static void main (String arg[])
    {

        String X = JOptionPane.showInputDialog(null,"Digite o valor de x = ");
        String Y = JOptionPane.showInputDialog(null,"Digite o valor de y = ");
        x = Double.parseDouble(X);
        y = Double.parseDouble(Y);

        if(x==y)
        {
            String st = "Sao iguais !";
            JOptionPane.showMessageDialog(null,st,"Resultado da Comparação", 1);
        }
        if(x!=y)
        {
            String st = "Sao diferentes !";
            JOptionPane.showMessageDialog(null,st,"Resultado da Comparação", 1);
        }
        System.exit(0);
    }
}

```

### Exemplo 2

```

import javax.swing.*;
public class Exemplo_if_3
{
    static String A , B ;

    public static void main (String arg[])
    {

        String A = JOptionPane.showInputDialog(null,"Digite um nome ");
        String B = JOptionPane.showInputDialog(null,"Digite outro nome ");
        if(A.equals(B) || A.equals("XX")) /*--- comparando duas Strings */
        {
            String st = ("As duas palavras sao "+A+" e "+B) ;
            JOptionPane.showMessageDialog(null,st,"Resultado da Comparação", 1);
        }
        System.exit(0);
    }
}

```

**Exemplo 3**

```

import javax.swing.*;
public class Exemplo_if_4
{
    static String A ;

    public static void main (String arg[])
    {
        String A = JOptionPane.showInputDialog(null,"Digite o valor de X ");

        int x = Integer.parseInt(A);

        if(x<10)
        {
            if(x>4)
            {
                if(x!=7)
                {
                    String st = "X menor do que 10 \n maior do que 4 \n mas diferente de 7 ";
                    JOptionPane.showMessageDialog(null,st,"Resultado da Comparação", 1);
                }
            }
        }

        String ts = ("X = "+"\""+x+"\"");
        JOptionPane.showMessageDialog(null,ts,"Fim da Comparação", 1);
        System.exit(0);
    }
}

```

**Exemplo 4**

```

import javax.swing.*;
public class Exemplo_if_5
{
    static String A ;

    public static void main (String arg[])
    {
        String A = JOptionPane.showInputDialog(null,"Digite o valor de X ");
        double x = Double.parseDouble(A);
        if(x==9)
        {
            x = Math.sqrt(x);
            String st = "A raiz quadrada de X = "+x ;
            JOptionPane.showMessageDialog(null,st,"Resultado da Comparação", 1);
        }
        else
        {
            x = Math.pow(x,2);
            String st = "O valor de X elevado ao quadrado = "+x ;
            JOptionPane.showMessageDialog(null,st,"Resultado da Comparação", 1);
        }
        System.exit(0);
    }
}

```

### Desvio condicional Switch

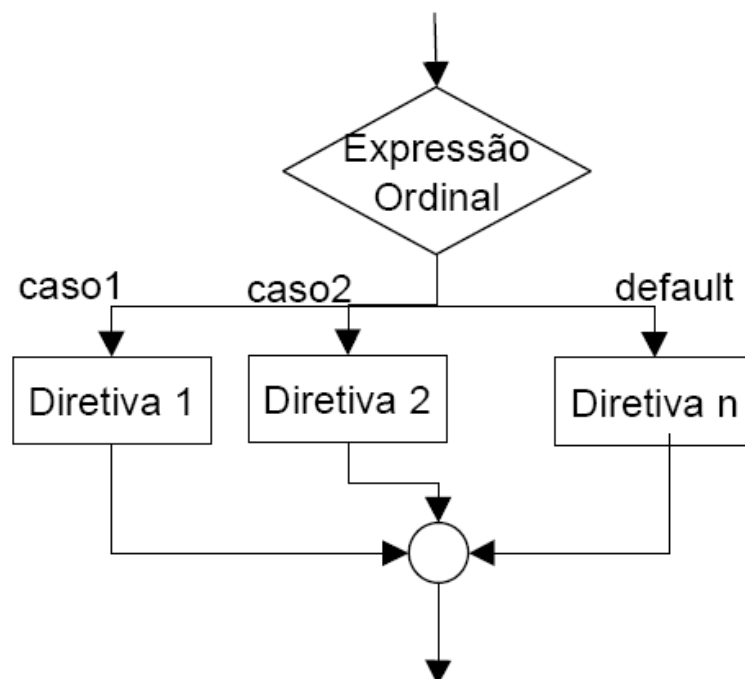
O *switch* equivale logicamente a um conjunto de diretivas *if* encadeadas, embora seja usualmente mais eficiente durante a execução.

A sintaxe desta diretiva é a seguinte:

```
switch (expressão_ordinal) {  
    case ordinal1: diretiva3;  
        break;  
    case ordinal2: diretiva2;  
        break;  
    default: diretiva_default;  
}
```

A expressão utilizada pelo *switch* deve necessariamente retornar um resultado ordinal. Conforme o resultado é selecionado um dos casos indicados pela construção *case* ordinal. As diretivas encontradas a partir do caso escolhido são executadas até o final da diretiva *switch* ou até uma diretiva *break* que encerra o *switch*. Se o valor resultante não possuir um caso específico são executadas as diretivas *default* colocadas, opcionalmente, ao final da diretiva *switch*.

### Comportamento do desvio condicional switch



```

import javax.swing.*;
public class Exemplo_Switch_2
{
    public static void main (String arg[])
    {
        int escolha ;
        // criar a "String desta forma facilita a formatação da mensagem
        String st = "Escolha um sabor :"+
            "\nAbacaxi      = 1"+
            "\nAbacate       = 2"+
            "\nAbobora        = 3"+
            "\nChocolate      = 4"+
            "\nCoco           = 5"+
            "\nLaranja         = 6"+
            "\nMorango          = 7"+
            "\nTangerina        = 8";

        /* em vez de escrever a mensagem dentro da caixa InputDialog ela foi escrita em separado. */
        String a = JOptionPane.showInputDialog(null,st);
        escolha = Integer.parseInt(a);

        switch(escolha)
        {
            case 1 : st ="Abacaxi";break ;
            case 2 : st ="Abacate";break ;
            case 3 : st ="Abóbora";break ;
            case 4 : st ="Chocolate";break ;
            case 5 : st ="Coco";break ;
            case 6 : st ="Laranja";break ;
            case 7 : st ="Morango";break ;
            case 8 : st ="Tangerina";break ;
            default : st ="Inexistente !";
        }
        // duas "Strings" são criadas fora do painel de mensagem
        String ms = "Sorveteria do Zé";
        String mss = "Sabor escolhido = ";

        // as tres "Strings" são chamadas dentro de MessageDialog
        JOptionPane.showMessageDialog(null,mss+st,ms,1);

        System.exit(0); // provoca a saída do sistema
    }
}

```