

# Decisões

Roland Teodorowitsch

Fundamentos de Programação - Escola Politécnica - PUCRS

11 de abril de 2023

# Introdução

# Objetivos

- Implementar decisões (simples e complexas) usando o comando `if`
- Comparar números e cadeias de caracteres
- Escrever comandos usando o tipo de dado `boolean`

# Conteúdos

- O Comando `if`
- Comparando *Strings*
- Múltiplas Alternativas
- Decisões Aninhadas
- Variáveis e operadores Booleanos
- Exercícios
- Tópicos Avançados

## O Comando if

# Decisões

- Um programa de computador frequentemente necessita tomar decisões baseadas em alguma entrada ou em dados sendo processados
- Exemplos:
  - Mostrar a mensagem “Aprovado”, se o aluno tem nota de g1 maior ou igual a 7,0; ou a mensagem “G2 ou Reprovado”, em caso contrário
  - Ao subir incrementar um valor, verificar se esse valor já não extrapolou o valor máximo permitido
  - Calcular o valor de uma multa e o número de pontos de um motorista que passa a uma certa velocidade acima do limite de uma via
  - Calcular as raízes de uma equação do segundo grau ( $ax^2 + bx + c = 0$ ) com delta podendo ser maior do que zero, igual a zero ou menor do que zero
  - etc.
- Para situações como estas, que exigem decisões, usa-se o comando `if`, que muitas vezes vem acompanhado de um `else`

# O Comando if

- `if` é o principal comando de decisão em Java
- Nos fluxogramas, `if` é representado por um losango
- O comando `if` deve especificar uma condição (expressão que gera um `boolean`) entre parênteses e um comando ou bloco para ser executado se a condição for verdadeira (`true`)

```
if ( delta < 0.0 )  
    System.out.println("SEM raizes reais");
```

- `else` pode ser usado para especificar o que fazer se a condição for falsa (`false`)

```
if ( g1 >= 7.0 )  
    System.out.println("Aprovado");  
else  
    System.out.println("G2 ou Reprovado");
```

# Expressões Lógicas

- A condição ou teste do `if` é uma **expressão lógica** ( resultado `true` ou `false`)
- Comparações entre variáveis e expressões numéricas são feitas usando **operadores relacionais**

```
if (tamanhoAtual > LIMITE) { ... } // MAIO
if (tamanhoAtual >= LIMITE) { ... } // MAIOR OU IGUAL
if (tamanhoAtual < LIMITE) { ... } // MENOR
if (tamanhoAtual <= LIMITE) { ... } // MENOR OU IGUAL
if (tamanhoAtual == LIMITE) { ... } // IGUAL
if (tamanhoAtual != LIMITE) { ... } // DIFERENTE
```

- Cálculos são feitos antes das comparações

```
if ( tamanhoAtual + 1 < LIMITE )
    tamanhoAtual = tamanhoAtual + 1;
```

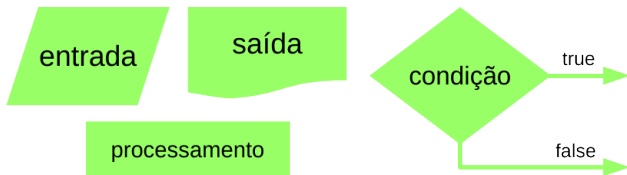


# Exemplos de Expressões Lógicas

- $3 \leq 4$       *Resultado: true*
- $3 > 4$       *Resultado: false*
- $4 < 4$       *Resultado: false*
- $4 \leq 4$       *Resultado: true*
- $3 == 5 - 2$       *Resultado: true*
- $3 != 5 - 1$       *Resultado: true*
- $3 = < 4$       *Resultado: ERRO DE SINTAXE: o operador correto é  $\leq$*
- $3 = 6 / 2$       *Resultado: ERRO DE SINTAXE: o operador correto é  $==$*
- $"10" > 5$       *Resultado: ERRO DE SINTAXE: NÃO se pode comparar strings com números*

# Fluxogramas

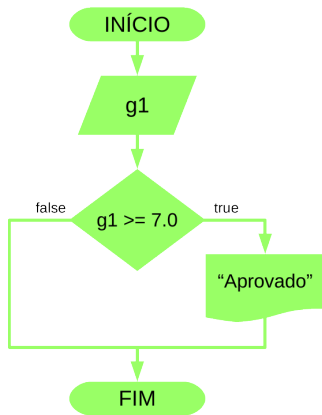
- Já vimos alguns exemplos de fluxogramas
- Um fluxograma mostra a estrutura de tarefas e decisões que tem que ser executadas para resolver um problema
- Os elementos básicos de um fluxograma são:



- Os elementos de um fluxograma são conectados com setas
- Nunca aponte uma seta para dentro de outro ramo (isso dificulta o entendimento e a manutenção)

# Fluxograma do comando `if`

- Um comando `if` pode não necessitar fazer nada se a condição for falsa

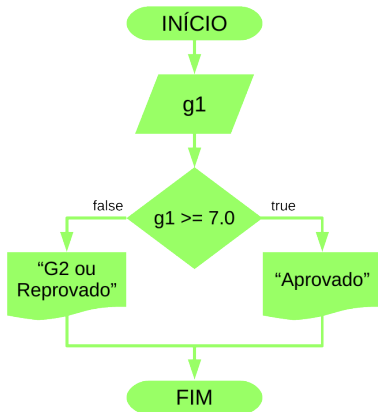


```
import java.util.Scanner;
import java.util.Locale;

public class Aprovado {
    public static void main(String[] args) {
        Locale.setDefault(Locale.ENGLISH);
        Scanner in = new Scanner(System.in);
        System.out.print("g1: ");
        double g1 = in.nextDouble();
        if (g1 >= 7.0) {
            System.out.println("Aprovado");
        }
        in.close();
    }
}
```

# Fluxograma do comando `if` com `else`

- Um comando `if` pode especificar o que fazer quando a condição for verdadeira e o que fazer quando ela for falsa (`else`)

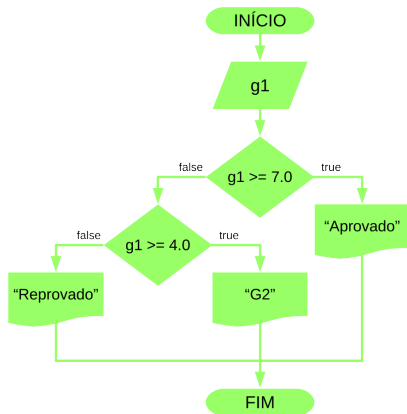


```
import java.util.Scanner;
import java.util.Locale;

public class AprovadoOuNao {
    public static void main(String[] args) {
        Locale.setDefault(Locale.ENGLISH);
        Scanner in = new Scanner(System.in);
        System.out.print("g1: ");
        double g1 = in.nextDouble();
        if (g1 >= 7.0) {
            System.out.println("Aprovado");
        }
        else {
            System.out.println("G2 ou Reprovado");
        }
        in.close();
    }
}
```

# Fluxograma com mais de 2 “ramos”

- Quando há mais de duas opções pode-se usar um `if/else` dentro de um `if` e/ou `else`

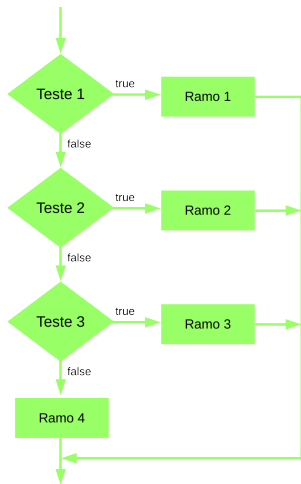


```

import java.util.Scanner;
import java.util.Locale;

public class AprovadoG2OuReprovado {
    public static void main(String[] args) {
        Locale.setDefault(Locale.ENGLISH);
        Scanner in = new Scanner(System.in);
        System.out.print("g1: ");
        double g1 = in.nextDouble();
        if (g1 >= 7.0) {
            System.out.println("Aprovado");
        }
        else {
            if (g1 >= 4.0)
                System.out.println("G2");
            else
                System.out.println("Reprovado");
        }
        in.close();
    }
}
  
```

# Fluxograma com mais ramos



```
if ( teste1 ) {  
    System.out.println("Ramo 1");  
}  
else {  
    if ( teste2 ) {  
        System.out.println("Ramo 2");  
    }  
    else {  
        if ( teste3 ) {  
            System.out.println("Ramo 3");  
        }  
        else {  
            System.out.println("Ramo 4");  
        }  
    }  
}
```

## Dicas sobre o Uso de Chaves

- O código Java é estruturado em blocos
- Uma indentação consistente torna o entendimento do código muito mais fácil
- Dois estilos de indentação comuns são os seguintes:
  - Abre-chaves embaixo do `if` e fecha-chaves na mesma coluna

```
if (tamanhoAtual < LIMITE)
{
    tamanhoAtual = tamanhoAtual + 1;
}
```

- Abre-chaves na linha do `if` e fecha-chaves na mesma coluna do `if`

```
if (tamanhoAtual < LIMITE) {
    tamanhoAtual ++;
}
```

# Dicas sobre o Uso de Chaves

- Mesmo que para seleção de um único comando NÃO seja necessário, para quem está aprendendo, recomenda-se sempre usar chaves
  - Em vez de

```
if (tamanhoAtual < LIMITE)  
    tamanhoAtual = tamanhoAtual + 1;
```

- Prefira usar

```
if (tamanhoAtual < LIMITE) {  
    tamanhoAtual = tamanhoAtual + 1;  
}
```



# Erros Comuns

- Um erro comum é colocar um `;` depois do comando `if`:

```
if (tamanhoAtual > LIMITE) ;  
{  
    tamanhoAtual = tamanhoAtual + 1;  
}
```

- Um `;`, sem comando antes, é um **comando vazio**
- Portanto, no exemplo acima, o comando `if` **NÃO FARÁ NADA** se o teste for verdadeiro (comando vazio) e o bloco (entre chaves) será executado independentemente do teste

# Exemplo

- **Problema:** A livraria da Universidade realiza um Dia Kilobyte de Descontos sempre no dia 24 de outubro, dando um desconto de 8% em todas as compras de acessórios de computador se o preço for menor do que R\$128,00, e um desconto de 16% se o preço é no mínimo R\$128,00. Faça um programa em Java que leia o valor original das compras, calculando e mostrando o valor da compra com o desconto aplicado.

# Exemplo: Passos

## ● Passos:

- 1 Decida qual será a condição para decisão:  
`preço original < 128?`
- 2 Escreva o pseudocódigo para o ramo verdadeiro:  
`preço com desconto = 0,92 x preço original`
- 3 Escreva o pseudocódigo para o ramo falso:  
`preço com desconto = 0,84 x preço original`
- 4 Faça uma verificação dos operadores relacionais, testando-os com valores abaixo (127), igual (128) e acima (129)
- 5 Remova a duplicação:  
`preço com desconto = ____ x preço original`
- 6 Teste ambos os ramos:  
`preço com desconto = 0,92 x 100 = 92`  
`preço com desconto = 0,84 x 200 = 168`
- 7 Escreva o código em Java

# Exemplo: Solução

```
import java.util.Scanner;
import java.util.Locale;

public class Livraria {
    public static void main(String[] args) {
        Locale.setDefault(Locale.ENGLISH);
        Scanner in = new Scanner(System.in);
        System.out.print("Preço original: ");
        double precoOriginal = in.nextDouble();
        double taxaDesconto;
        if (precoOriginal < 128.0 )
            taxaDesconto = 0.92;
        else
            taxaDesconto = 0.84;
        double precoComDesconto = taxaDesconto * precoOriginal;
        System.out.printf("Preço com desconto: %.2f\n", precoComDesconto);
        in.close();
    }
}
```

# Exercícios

- 1 Escreva um programa em Java que verifique se um número inteiro lido do terminal é par ou ímpar.
- 2 Escreva um programa em Java que verifique se um número real lido do terminal é negativo, zero ou positivo.
- 3 Implemente um programa para ler os coeficientes  $a$ ,  $b$  e  $c$  de uma equação do segundo grau, calculando e mostrando as raízes reais dessa equação  $ax^2 + bx + c = 0$ . A solução pode ser obtida com a fórmula de Bhaskara

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

onde:  $\Delta = b^2 - 4ac$ .

## Dica sobre o Exercício 3

- Para que a equação seja do segundo grau, deve-se ter  $a \neq 0$
- O valor de  $\Delta$  determina se:
  - há duas raízes reais ( $\Delta > 0$ )
  - há uma raiz real ( $\Delta = 0$ ) ou
  - NÃO há raízes reais ( $\Delta < 0$ )
- É importante testar essas condições no programa...

# Solução do Exercício 1: ParOuImpar.java

```
import java.util.Scanner;

public class ParOuImpar {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Digite um valor inteiro: ");
        int n = in.nextInt();
        String res;
        if (n % 2 == 1) {
            res = "IMPAR";
        }
        else {
            res = "PAR";
        }
        System.out.printf("%d eh %s!\n",n,res);
        in.close();
    }
}
```

# Solução do Exercício 2: NegativoZeroPositivo.java

```
import java.util.Scanner;

public class NegativoZeroPositivo {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Digite um valor real: ");
        double n = in.nextDouble();
        if ( n < 0.0 ) {
            System.out.printf("%f eh NEGATIVO.\n",n);
        }
        else { // n >= 0.0
            if ( n == 0.0 ) {
                System.out.printf("%f eh ZERO.\n",n);
            }
            else { // n > 0.0
                System.out.printf("%f eh POSITIVO.\n",n);
            }
        }
        in.close();
    }
}
```



# Solução do Exercício 3: Bhaskara.java

```
import java.util.Scanner;
public class Bhaskara {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("a: "); double a = in.nextDouble();
        System.out.print("b: "); double b = in.nextDouble();
        System.out.print("c: "); double c = in.nextDouble();
        if ( a == 0.0 ) { System.out.println("NAO eh uma equacao do segundo grau..."); }
        else {
            double delta = b*b - 4*a*c;
            if ( delta < 0.0 ) { System.out.println("NAO ha raizes reais..."); }
            else {
                if ( delta == 0.0 ) {
                    double r = -b / (2*a);
                    System.out.printf("Raiz: %f\n",r);
                }
                else { // delta > 0.0
                    delta = Math.sqrt(delta);
                    double r1 = (-b+delta)/(2*a);
                    double r2 = (-b-delta)/(2*a);
                    System.out.printf("Raizes: %f, %f\n",r1, r2);
                }
            }
        }
        in.close();
    }
}
```

# Comparando Strings

# Comparando *Strings*

- *Strings* são um pouco “especiais” em Java
- Não use o operador `==` com *strings*
  - O seguinte trecho funciona, mas na verdade compara a localização de duas *strings* e não os seus conteúdos

```
if (string1 == string2) ...
```

- Em vez disto uso o método `equals`:

```
if (string1.equals(string2)) ...
```

# Exemplos

```
if ( "Tomate".substring(0,3).equals("Tom") )  
    System.out.println( ">>> true <<<" );  
else  
    System.out.println( "false" );  
  
if ( "Tomate".substring(0,3) == ("Tom") )  
    System.out.println( "true" );  
else  
    System.out.println( ">>> false <<<" );
```

## Mais um exemplo

- Java cria uma nova variável `String` cada vez que é usado um texto entre aspas
- Se há uma *string* que coincida exatamente com ela, Java a reusa

```
String nickname = "Rob";  
if (nickname == "Rob")  
    System.out.println( ">>> true <<<" );  
else  
    System.out.println( "false" );
```

```
String name = "Robert";  
String nickname = name.substring(0,3);  
if (nickname == "Rob")  
    System.out.println( "true" );  
else  
    System.out.println( ">>> false <<<" );
```

# Ordem Lexicográfica

- Para comparar *strings* pela ordem de dicionário, pode-se usar `compareTo()`
- Considerando duas *strings* `string1` e `string2`:
  - `string1.compareTo(string2)` será **menor do que zero** se `string1` vier antes de `string2`
  - `string1.compareTo(string2)` será **igual a zero** se elas forem iguais
  - `string1.compareTo(string2)` será **maior do que zero** se `string1` vier depois de `string2`
- Observações:
  - Todas as letras maiúsculas vem antes das minúsculas
  - “Espaço” vem antes de todos os caracteres imprimíveis
  - Dígitos (0-9) vem antes das letras
- Lembre-se: para comparar desconsiderando a diferença entre minúsculas e maiúsculas, pode-se usar `equalsIgnoreCase` e `compareToIgnoreCase`

# Exercícios

- 1 Escreva um programa em Java que pergunte ao usuário se ele deseja sair do programa ou não (“Sair do programa (SIM ou NÃO)? ”). Se o usuário desejar sair, execute `System.exit(0)`; para abandonar o programa. Em caso contrário, leia um nome completo e imprima uma saudação juntamente com o nome lido (por exemplo, se nome lido for “Han Solo”, imprima “Bom dia, Han Solo!”). Para testar a resposta da pergunta, seu programa deverá aceitar todas as combinações de maiúsculas e minúsculas para “sim”, com ou sem espaços antes ou depois (por exemplo, “Sim”, “ slm ”, “siM ”, etc.).
- 2 Escreva um programa em Java que leia duas palavras e verifique se elas são iguais entre si ou NÃO. Seu programa deverá imprimir uma das seguintes mensagens: “IGUAIS” ou “DIFERENTES”.
- 3 Escreva um programa em Java que leia dois nomes de pessoas e mostre esses em ordem alfabética crescente, ignorando o uso de minúsculas ou maiúsculas. Por exemplo, se forem lidos, respectivamente, “LUKE SKYWALKER” e “Leia Organa”, seu programa deverá imprimir: “Leia Organa” e “LUKE SKYWALKER”.

# Solução do Exercício 1: SairDoPrograma.java

```
import java.util.Scanner;

public class SairDoPrograma {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Sair do programa (SIM ou NÃO)? ");
        String resposta = in.nextLine().trim().toUpperCase();
        if (resposta.equals("SIM")) {
            System.out.println("Saindo do programa...");
            System.exit(0);
        }
        System.out.print("Digite seu nome: ");
        String nome = in.nextLine().trim();
        System.out.printf("Bom dia, %s!\n", nome);
        in.close();
    }
}
```



## Solução do Exercício 2: IguaisOuDiferentes.java

```
import java.util.Scanner;

public class IguaisOuDiferentes {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Palavra 1? ");
        String palavra1 = in.next();
        System.out.print("Palavra 2? ");
        String palavra2 = in.next();
        if (palavra1.equals(palavra2) ) {
            System.out.println("IGUAIS");
        }
        else {
            System.out.println("DIFERENTES");
        }
        in.close();
    }
}
```

## Solução do Exercício 3: OrdenaDoisNomes.java

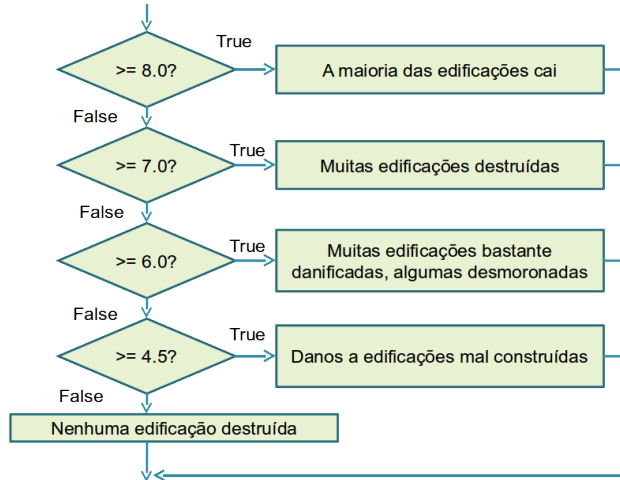
```
import java.util.Scanner;
public class OrdenaDoisNomes {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Nome 1: ");    String nome1 = in.nextLine().trim();
        System.out.print("Nome 2: ");    String nome2 = in.nextLine().trim();
        if ( nome1.compareToIgnoreCase(nome2) < 0 ) {
            System.out.println(nome1);
            System.out.println(nome2);
        }
        else {
            System.out.println(nome2);
            System.out.println(nome1);
        }
        in.close();
    }
}
```

# Múltiplas Alternativas

# Múltiplas Alternativas

- Um `if` tem 2 ramos, mas o que acontece se forem necessários mais do que dois ramos?
- Por exemplo, uma escala para o efeito de um terremoto:
  - 8 (ou mais): a maioria das edificações cai
  - 7 até 7.99: muitas edificações destruídas
  - 6 até 6.99: muitas edificações bastante danificadas, algumas desmoronadas
  - 4.5 até 5.99: danos a edificações mal construídas
  - menos do que 4.5: nenhuma edificação destruída

# Fluxograma para Múltiplos Ramos



# O que há de errado com este código?

```
if (richter >= 8.0) {  
    System.out.println("A maioria das edificacoes cai");  
}  
if (richter >= 7.0) {  
    System.out.println("Muitas edificacoes destruidas");  
}  
if (richter >= 6.0) {  
    System.out.println("Muitas edificacoes bastante danificadas, "+  
                        "algumas desmoronadas");  
}  
if (richter >= 4.5) {  
    System.out.println("Danos a edificacoes mal construidas");  
}  
if (richter < 4.5) {  
    System.out.println("Nenhuma edificacao destruida");  
}
```

# E com este código? O que há de errado?

```
if (richter >= 8.0) {  
    System.out.println("A maioria das edificacoes cai");  
}  
if (richter < 8.0 && richter >= 7.0) {  
    System.out.println("Muitas edificacoes destruidas");  
}  
if (richter < 7.0 && richter >= 6.0) {  
    System.out.println("Muitas edificacoes bastante danificadas, "+  
        "algumas desmoronadas");  
}  
if (richter < 6.0 && richter >= 4.5) {  
    System.out.println("Danos a edificacoes mal construidas");  
}  
if (richter < 4.5) {  
    System.out.println("Nenhuma edificacao destruida");  
}
```

# Construção if-else-if

```
if (richter >= 8.0) {  
    System.out.println("A maioria das edificacoes cai");  
}  
else if (richter >= 7.0) {  
    System.out.println("Muitas edificacoes destruidas");  
}  
else if (richter >= 6.0) {  
    System.out.println("Muitas edificacoes bastante danificadas, "+  
                        "algumas desmoronadas");  
}  
else if (richter >= 4.5) {  
    System.out.println("Danos a edificacoes mal construidas");  
}  
else {  
    System.out.println("Nenhuma edificacao destruida");  
}
```



## Outra forma de Implementar Múltiplos Ramos

- O comando `switch` escolhe uma opção de um conjunto de opções
- Cada opção deve ser especificada por um `case`
- `default` trata todas as opções não especificadas em `cases`
- NÃO serve para intervalos, apenas para valores discretos e para os seguintes tipos:
  - tipos primitivos `byte`, `short`, `int`, `long` e `char`
  - variáveis/objetos de classes (`Byte`, `Short`, `Integer`, `Long` e `Character`) que podem ser convertidas para os tipos primitivos suportados
  - variáveis/objetos da classe `String`
  - enumerações
- Para encerrar um `case` deve-se usar o comando `break`

# Exemplo de switch/case

```
Scanner in = new Scanner(System.in);  
int unidade = in.nextInt() % 10;  
switch (unidade) {  
    case 1: System.out.println("um");      break;  
    case 2: System.out.println("dois");     break;  
    case 3: System.out.println("três");     break;  
    case 4: System.out.println("quatro");   break;  
    case 5: System.out.println("cinco");     break;  
    case 6: System.out.println("seis");      break;  
    case 7: System.out.println("sete");      break;  
    case 8: System.out.println("oito");      break;  
    case 9: System.out.println("nove");      break;  
    default: System.out.println("zero");  
}
```

# Enumerações

- Definem uma lista finita de valores que uma variável pode armazenar
- Funcionam como uma declaração de novos tipos, com uma lista de possíveis valores

```
public enum EstadoCivil {  
    SOLTEIRO, CASADO, SEPARADO, DIVORCIADO, VIUVO  
}
```

- Você pode ter qualquer número de valores, mas tem que incluir eles todos na declaração `enum`
- É possível declarar variáveis do tipo de uma “enumeração”

```
EstadoCivil estado = EstadoCivil.SOLTEIRO;
```

- Você também pode usar o operador de comparação entre eles:

```
if (estado == EstadoCivil.SOLTEIRO) ...
```

# Enumerações: Exemplos

```
public enum DiasDaSemana {  
    DOMINGO, SEGUNDA_FEIRA, TERCA_FEIRA, QUARTA_FEIRA,  
    QUINTA_FEIRA, SEXTA_FEIRA, SABADO  
}  
  
public enum MesesDoAno {  
    JANEIRO, FEVEREIRO, MARCO, ABRIL, MAIO, JUNHO,  
    JULHO, AGOSTO, SETEMBRO, OUTUBRO, NOVEMBRO, DEZEMBRO  
}  
  
public enum EstadoSemaforo {  
    VERMELHO, AMARELO, VERDE  
}
```

# Exercícios

- 1 Faça um programa em Java para ler o peso em Kg de um ou uma atleta adulto de Judô, mostrando a categoria à qual o ou a atleta pertence.
  - Categorias Masculinas: Superligeiro (até 55 Kg), Ligeiro (até 60 Kg); Meio-leve (até 66 Kg), Leve (até 73 Kg), Meio-médio (até 81 Kg), Médio (até 90 Kg), Meio-pesado (até 100 Kg), Pesado (mais de 100 Kg).
  - Categorias Femininas: Superligeiro (até 44 Kg), Ligeiro (até 48 Kg), Meio-leve (até 52 Kg), Leve 1 (até 57 Kg), Meio-médio (até 63 Kg), Médio (até 70 Kg), Meio-pesado (até 78 Kg), Pesado (mais de 78 Kg).
- 2 Escreva um programa em Java que leia o nome do mês, aceitando qualquer combinação de maiúsculas e minúsculas (eliminando também eventuais espaços no início e no final da leitura), e converta este nome no número inteiro de mês correspondente. Por exemplo, “janeiro” corresponde a 1, “fevereiro” corresponde a 2, etc. Nomes inválidos de mês correspondem a -1. Inicialmente implemente usando `if/else`, depois faça uma versão com `switch/case`.
- 3 Escreva um programa em Java que leia um valor inteiro correspondendo ao dia da semana (1 correspondendo a domingo, 2 correspondendo a segunda-feira, 3 correspondendo a terça-feira, 4 correspondendo a quarta-feira, 5 correspondendo a quinta-feira, 6 correspondendo a sexta-feira e 7 correspondendo a sábado). E imprima “DIA ÚTIL” para os dias de segunda-feira até sexta-feira, “DESCANSO” para sábado e domingo e “INVÁLIDO” para qualquer outro valor. Inicialmente implemente usando `if/else`, depois faça uma versão com `switch/case`.

# Solução do Exercício 1: JudoMasculino.java

```
import java.util.Scanner;
public class JudoMasculino {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("JUDO MASCULINO\nQual o peso do atleta em Kg? ");
        double peso = in.nextDouble();
        System.out.print("Categoria: ");
        if ( peso <= 55.0 ) { System.out.println("Superligeiro"); }
        else if ( peso <= 60.0 ) { System.out.println("Ligeiro"); }
        else if ( peso <= 66.0 ) { System.out.println("Meio-leve"); }
        else if ( peso <= 73.0 ) { System.out.println("Leve"); }
        else if ( peso <= 81.0 ) { System.out.println("Meio-médio"); }
        else if ( peso <= 90.0 ) { System.out.println("Médio"); }
        else if ( peso <= 100.0 ) { System.out.println("Meio-pesado"); }
        else { System.out.println("Pesado"); }
        in.close();
    }
}
```

# Solução do Exercício 1: JudoFeminino.java

```
import java.util.Scanner;
public class JudoFeminino {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("JUDO FEMININO\nQual o peso do atleta em Kg? ");
        double peso = in.nextDouble();
        System.out.print("Categoria: ");
        if ( peso <= 44.0 ) { System.out.println("Superligeiro"); }
        else if ( peso <= 48.0 ) { System.out.println("Ligeiro"); }
        else if ( peso <= 52.0 ) { System.out.println("Meio-leve"); }
        else if ( peso <= 57.0 ) { System.out.println("Leve"); }
        else if ( peso <= 63.0 ) { System.out.println("Meio-médio"); }
        else if ( peso <= 70.0 ) { System.out.println("Médio"); }
        else if ( peso <= 78.0 ) { System.out.println("Meio-pesado"); }
        else { System.out.println("Pesado"); }
        in.close();
    }
}
```

# Solução do Exercício 2: Meses.java

```
import java.util.Scanner;
public class Meses {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Digite um nome de mês: ");
        String strMes = in.next().toLowerCase();
        int mes;
        if ( strMes.equals("janeiro") ) { mes = 1; }
        else if ( strMes.equals("fevereiro") ) { mes = 2; }
        else if ( strMes.equals("março") ) { mes = 3; }
        else if ( strMes.equals("abril") ) { mes = 4; }
        else if ( strMes.equals("maio") ) { mes = 5; }
        else if ( strMes.equals("junho") ) { mes = 6; }
        else if ( strMes.equals("julho") ) { mes = 7; }
        else if ( strMes.equals("agosto") ) { mes = 8; }
        else if ( strMes.equals("setembro") ) { mes = 9; }
        else if ( strMes.equals("outubro") ) { mes = 10; }
        else if ( strMes.equals("novembro") ) { mes = 11; }
        else if ( strMes.equals("dezembro") ) { mes = 12; }
        else { mes = -1; }
        System.out.printf("Número do mês = %d\n", mes);
        in.close();
    }
}
```



# Solução do Exercício 2: MesesComSwitch.java

```
import java.util.Scanner;
public class MesesComSwitch {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Digite um nome de mês: ");
        String strMes = in.next().toLowerCase();
        int mes;
        switch ( strMes ) {
            case "janeiro":    mes = 1; break;
            case "fevereiro": mes = 2; break;
            case "março":     mes = 3; break;
            case "abril":     mes = 4; break;
            case "maio":      mes = 5; break;
            case "junho":     mes = 6; break;
            case "julho":     mes = 7; break;
            case "agosto":    mes = 8; break;
            case "setembro":  mes = 9; break;
            case "outubro":   mes = 10; break;
            case "novembro":  mes = 11; break;
            case "dezembro":  mes = 12; break;
            default:          mes = -1;
        }
        System.out.printf("Número do mês = %d\n", mes);
        in.close();
    }
}
```

# Solução do Exercício 3: DiasDaSemana.java

```
import java.util.Scanner;
public class DiasDaSemana {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Digite o valor do dia da semana: ");
        int diaDaSemana = in.nextInt();
        String resultado;
        if ( diaDaSemana == 1 ) { resultado = "DESCANSO"; }
        else if ( diaDaSemana == 2 ) { resultado = "DIA ÚTIL"; }
        else if ( diaDaSemana == 3 ) { resultado = "DIA ÚTIL"; }
        else if ( diaDaSemana == 4 ) { resultado = "DIA ÚTIL"; }
        else if ( diaDaSemana == 5 ) { resultado = "DIA ÚTIL"; }
        else if ( diaDaSemana == 6 ) { resultado = "DIA ÚTIL"; }
        else if ( diaDaSemana == 7 ) { resultado = "DESCANSO"; }
        else { resultado = "INVÁLIDO"; }
        System.out.println(resultado);
        in.close();
    }
}
```

# Solução do Exercício 3: DiasDaSemanaComSwitch.java

```
import java.util.Scanner;

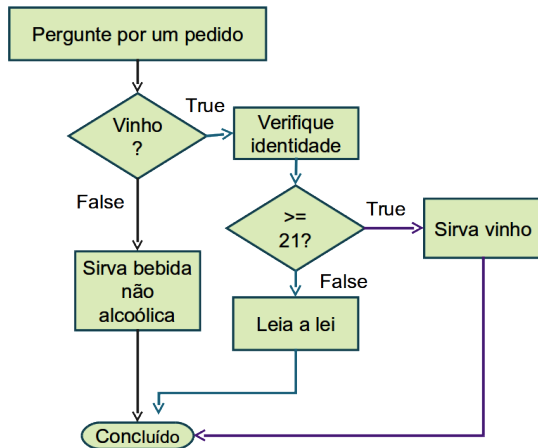
public class DiasDaSemanaComSwitch {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Digite o valor do dia da semana: ");
        int diaDaSemana = in.nextInt();
        String resultado;
        switch (diaDaSemana) {
            case 1:
            case 7: resultado = "DESCANSO"; break;
            case 2:
            case 3:
            case 4:
            case 5:
            case 6: resultado = "DIA ÚTIL"; break;
            default: resultado = "INVÁLIDO";
        }
        System.out.println(resultado);
        in.close();
    }
}
```

# Decisões Aninhadas

# Decisões Aninhadas

- Você pode aninhar um `if` dentro de um ramo de um comando `if`
- Um exemplo simples: pedidos de bebidas em um bar
  - Pergunte ao cliente o que ele quer beber
  - Se o cliente pedir vinho:
    - Peça a identidade do cliente
    - Se a idade do cliente for maior ou igual a 21, então  
Sirva vinho
    - Senão  
Polidamente explique como funciona a lei ao cliente
  - Senão
    - Sirva uma bebida não alcoólica ao cliente

# Fluxograma de um `if` aninhado



`if-else` aninhado dentro do ramo verdadeiro de um comando `if`: 3 partes

## Erro Comum: `else` “desamparado”

- Quando um `if` é aninhado dentro de outro `if`, o seguinte pode ocorrer:

```
double custoDeEnvio = 5.00;
if (pais.equals("EUA"))
    if (alaskaOuHawaii.equals("SIM"))
        custoDeEnvio = 10.00;
else // Cuidado!
    custoDeEnvio = 20.00;
```

- O nível de indentação sugere que o `else` esteja relacionado ao `if` de `pais.equals("EUA")`
- Porém a cláusula `else` sempre se associa ao `if` mais próximo

# Variáveis e operadores Booleanos



# Variáveis e operadores Booleanos

## ● Variáveis Booleanas

- Uma variável booleana é frequentemente chamada de *flag* porque pode assumir ou o valor verdadeiro (*true*) ou falso (*false*)
- Java dispõe do tipo `boolean` para variáveis booleanas, que podem assumir ou `true` ou `false`

```
boolean acertou = true;  
boolean sair = false;
```

## ● Operadores Booleanos: `&&` e `||`

- São usados para combinar múltiplas condições
- `&&` é o operador AND (E)
- `||` é o operador OR (OU)

## Condições Combinadas: & &

- A combinação de dois testes ou condições é usada frequentemente para verificar se um valor está dentro de um intervalo
- Ambos os lados do AND devem ser verdadeiros para que o resultado também seja:

```
if (temperatura > 0 && temperatura < 100) {  
    System.out.println("Líquido");  
}
```

A	B	A && B
true	true	true
true	false	false
false	true	false
false	false	false

# Condições Combinadas: ||

- Se apenas um dos testes precisa ser verdadeiro, pode-se combinar os testes com OR
- Se um dos lados do OR for verdadeiro, o resultado também será:

```
if (saldo > 100.0 || credito > 100.0) {  
    System.out.println("Aceito");  
}
```

A	B	A    B
true	true	true
true	false	true
false	true	true
false	false	false

# O Operador NOT: !

- Se for necessário inverter o valor de uma variável booleana ou de uma condição, basta precedê-la com !:

A	!A
true	false
false	true

- O que é melhor?

```
if (!temPresencas || nota < 7.0)
    System.out.println("Desistiu?");
```

```
if (temPresencas && !(nota < 7.0))
    System.out.println("Aprovado");
```

- Ao usar !, procure simplificar a lógica:

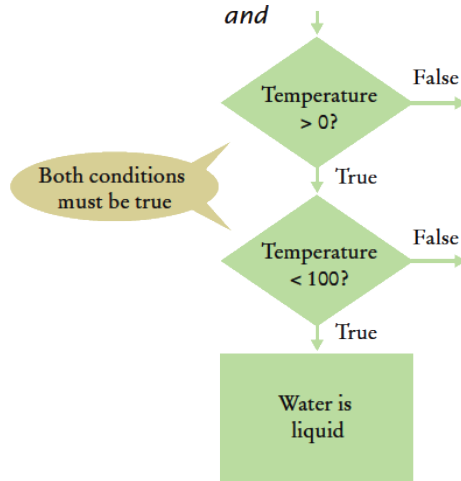
```
if (temPresencas && nota >= 7.0)
    System.out.println("Aprovado");
```

# Fluxograma para AND

- Isto é frequentemente chamado de “verificação de limite”, sendo usado para validar se uma entrada está entre 2 valores

```
if (temperatura > 0 && temperatura < 100) {  
    System.out.println("Líquido");  
}
```

# Fluxograma para AND

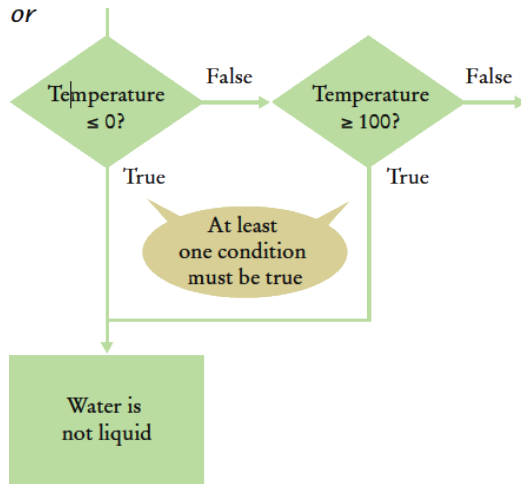


# Fluxograma para OR

- Outra forma de “verificação de limite”: verifica se o valor está fora do limite

```
if (temperatura <= 0 || temperatura >= 100) {  
    System.out.println("Gelo ou Vapor");  
}
```

# Fluxograma para OR





# Exemplos de Uso de Operadores Booleanos

Expressão	Valor	Comentário
<code>0 &lt; 200 &amp;&amp; 200 &lt; 100</code>	false	Apenas a primeira condição é verdadeira
<code>0 &lt; 200    200 &lt; 100</code>	true	A primeira condição é verdadeira
<code>0 &lt; 200    100 &lt; 200</code>	true	Ambas as condições são verdadeiras
<code>!(0 &lt; 200)</code>	false	<code>0 &lt; 200</code> é verdadeiro, e a sua negação é falsa
<code>true    true &amp;&amp; false</code>	true	<code>&amp;&amp;</code> tem maior precedência que <code>  </code>
<code>true    (true &amp;&amp; false)</code>	true	Os parênteses confirmam a precedência
<code>(true    true) &amp;&amp; false</code>	false	Os parênteses alteram a precedência
<code>frozen == true</code>	frozen	NÃO é necessário comparar uma variável booleana com <code>true</code>
<code>frozen == false</code>	!frozen	É mais claro usar <code>!</code> do que comparar com <code>false</code>

# Erros Comuns

- O seguinte formato é usado na matemática, mas NÃO em Java:

```
if (0 <= temperatura <= 100) // Erro de sintaxe!
```

São necessárias duas comparações:

```
if (0 <= temperatura && temperatura <= 100)
```

- Isto também não é permitido em Java:

```
if (input == 1 || 2) // Erro de sintaxe!
```

É preciso usar 2 comparações:

```
if (input == 1 || input == 2)
```

- Cuidado para NÃO confundir && (dentro do intervalo) e || (fora do intervalo)

# Exercícios

# Exercícios (1)

- 1 Tendo como dados de entrada a altura (em metros) e o gênero de uma pessoa, construa um programa em Java que calcule o seu peso ideal, utilizando as seguintes fórmulas:
  - Para mulheres:  $(62.1 * \text{altura}) - 44.7$
  - Para homens:  $(72.7 * \text{altura}) - 58$
- 2 Escreva um programa em Java que leia dois valores reais,  $x$  e  $y$ , que correspondem às coordenadas de um ponto no plano cartesiano. E imprima em que quadrante este ponto se encontra (“Quadrante 1”, “Quadrante 2”, “Quadrante 3”, “Quadrante 4”) ou se ele se encontra na origem (“Origem”) ou sobre um dos eixos (“Eixo X”, “Eixo Y”).
- 3 Escreva um programa em Java que leia o ano de nascimento de uma pessoa, calcule e mostre sua idade, e também verifique e mostre:
  - se ela já deve votar (obrigatório para pessoas com idade entre 18 e 70 anos), pode votar (opcional para pessoas com idade entre 16 e 18 anos ou maior que 70 anos) ou não pode votar (impedido para pessoas com idade menor que 16 anos); e
  - se ela tem idade para conseguir Carteira de Habilitação (18 anos ou mais).

## Exercícios (2)

- 4 Escreva um programa em Java que leia o código (inteiro) de um determinado produto e mostre a sua classificação. Utilize as seguintes categorias para classificar os produtos: 1 para “Alimento não perecível”; 2, 3 ou 4 para “Alimento perecível”; 5 ou 6 para “Vestuário”; 7 para “Higiene pessoal”; e qualquer outro código para “Inválido”.

## Exercícios (3)

- 5 Escreva um programa em Java que leia as 4 notas de um aluno de Fundamentos de Programação:  $P_1$  (nota da prova 1),  $P_2$  (nota da prova 2),  $M_E$  (média de exercícios) e  $T_F$  (nota do trabalho final). Calcule o seu grau  $G_1$ , mostrando-o juntamente com uma mensagem indicando se o aluno passou por média ( $G_1 \geq 7$ ), ficou em  $G_2$  ( $G_1 \geq 4$  e  $G_1 < 7$ ) ou reprovou ( $G_1 < 4$ ). Caso o aluno tenha ficado em  $G_2$ , calcule qual a nota mínima que o aluno deve tirar nessa prova para obter aprovação. Considere que a média de  $G_1$  é calculada com a seguinte fórmula:

$$G_1 = \frac{P_1 + 2 \times P_2 + M_E + 2 \times T_F}{6}$$

Lembre-se que, para obter aprovação depois da realização do  $G_2$ , a média aritmética entre  $G_1$  e  $G_2$  deve ser maior ou igual a 5.

## Exercícios (4)

- 6 Escreva um trecho de programa em Java para calcular o número de pontos e o valor de multas de trânsito por excesso de velocidade. Inicialmente seu programa deverá ler o limite de velocidade da via e a velocidade do veículo (medida com um radar), ambos em quilômetros por hora e sem casas decimais. Para determinar a velocidade do veículo que será considerada, aplica-se uma tolerância de 7 km/h, se a velocidade medida for menor ou igual a 100 km/h, ou de 7%, em caso contrário. Se a velocidade considerada for menor ou igual ao limite da via, não há multa. Se a velocidade considerada exceder 50% do limite da via, a multa será de R\$880,41 (infração gravíssima, 7 pontos). Se a velocidade considerada exceder 20% do limite da via, a multa será de R\$195,23 (infração grave, 5 pontos). Senão a multa será de R\$130,16 (infração média, 4 pontos).

# Solução do Exercício 1: PesoIdeal.java

```
import java.util.Scanner;

public class PesoIdeal {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Altura (em m)? ");
        double altura = in.nextDouble();
        System.out.print("Sexo (HOMEM ou MULHER)? ");
        String sexo = in.next().toUpperCase();
        double pesoIdeal;
        if (sexo.equals("HOMEM") )
            pesoIdeal = (72.7 * altura) - 58.0;
        else
            pesoIdeal = (62.1 * altura) - 44.7;
        System.out.printf("Peso ideal: %.3f Kg\n", pesoIdeal);
        in.close();
    }
}
```



# Solução do Exercício 2: Quadrante.java

```
import java.util.Scanner;

public class Quadrante {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        double x = in.nextDouble();
        double y = in.nextDouble();
        if ( x == 0.0 ) {
            if ( y == 0.0 ) System.out.println("Origem");
            else
                System.out.println("Eixo y");
        }
        else { // x != 0
            if ( x > 0 ) { // 1, Eixo X, 4
                if ( y > 0.0 ) System.out.println("Quadrante 1");
                else if ( y == 0.0 ) System.out.println("Eixo X");
                else
                    System.out.println("Quadrante 4");
            }
            else { // x < 0 ==> 2, Eixo X, 3
                if ( y > 0.0 ) System.out.println("Quadrante 2");
                else if ( y == 0.0 ) System.out.println("Eixo X");
                else
                    System.out.println("Quadrante 3");
            }
        }
        in.close();
    }
}
```

# Solução do Exercício 3: Idade.java

```
import java.util.Scanner;

public class Idade {
    public static void main(String[] args) {
        final int ANO_ATUAL = 2023;
        Scanner in = new Scanner(System.in);
        System.out.print("Ano de nascimento: ");
        int nascimento = in.nextInt();
        int idade = ANO_ATUAL - nascimento;
        System.out.printf("Idade: %d anos em %d\n", idade, ANO_ATUAL);
        if (idade < 16)
            System.out.println("- NÃO pode votar");
        else if (idade < 18 || idade > 70)
            System.out.println("- pode votar");
        else
            System.out.println("- deve votar");
        if (idade >= 18)
            System.out.println("- tem idade para ter Carteira de Motorista");
        else
            System.out.println("- NÃO tem idade para ter Carteira de Motorista");
        in.close();
    }
}
```

# Solução do Exercício 4: Classificacao.java

```
import java.util.Scanner;

public class Classificacao {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Código do produto? ");
        int codigo = in.nextInt();
        String classificacao = "";
        if (codigo==1) { classificacao = "Alimento não perecível"; }
        else if (codigo==2 || codigo==3 || codigo==4) { classificacao = "Alimento perecível"; }
        else if (codigo==5 || codigo==6) { classificacao = "Vestuário"; }
        else if (codigo==7) { classificacao = "Higiene pessoal"; }
        else { classificacao = "Inválido"; }
        System.out.printf("Classificacao: %s\n", classificacao);
    }
}
```

# Solução do Exercício 4: ClassificacaoComSwitch.java

```
import java.util.Scanner;

public class ClassificacaoComSwitch {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Código do produto? ");
        int codigo = in.nextInt();
        String classificacao = "";
        switch(codigo) {
            case 1: classificacao = "Alimento não perecível"; break;
            case 2:
            case 3:
            case 4: classificacao = "Alimento perecível"; break;
            case 5:
            case 6: classificacao = "Vestuário"; break;
            case 7: classificacao = "Higiene pessoal"; break;
            default: classificacao = "Inválido";
        }
        System.out.printf("Classificacao: %s\n", classificacao);
    }
}
```

# Solução do Exercício 5: Notas.java

```
import java.util.Scanner;

public class Notas {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("P1? "); double p1 = in.nextDouble();
        System.out.print("P2? "); double p2 = in.nextDouble();
        System.out.print("ME? "); double me = in.nextDouble();
        System.out.print("TF? "); double tf = in.nextDouble();
        double g1 = (p1 + 2*p2 + me + 2*tf) / 6.0;
        if (g1 >= 7.0) { System.out.printf("G1=%.1f APROVADO\n", g1); }
        else if (g1 >= 4.0) {
            double g2Minimo = 10.0 - g1;
            System.out.printf("G1=%.1f ALUNO EM G2 (G2>=%.1f)\n", g1, g2Minimo);
        }
        else { System.out.printf("G1=%.1f REPROVADO\n", g1); }
        in.close();
    }
}
```

# Solução do Exercício 6: MultaDeTransito.java

```
import java.util.Scanner;

public class MultaDeTransito {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Limite de velocidade da via (km/h): "); int limVel = in.nextInt();
        System.out.print("Velocidade do veículo (km/h): ");      int velMed = in.nextInt();
        double velConsiderada;
        if (velMed <= 100)
            velConsiderada = velMed - 7;
        else
            velConsiderada = velMed * 0.93;
        if (velConsiderada <= limVel) {
            System.out.println("SEM multa.");
        }
        else if (velConsiderada > limVel * 1.5) {
            System.out.println("Multa de R$880,41, infração gravíssima, 7 pontos.");
        }
        else if (velConsiderada > limVel * 1.2) {
            System.out.println("Multa de R$195,23, infração grave, 5 pontos.");
        }
        else {
            System.out.println("Multa de R$130,16, infração média, 4 pontos.");
        }
    }
}
```

# Tópicos Avançados

# Tópicos Avançados

- Comparação de Números de Ponto-flutuante
- Expressões Condicionais
- Casos de Teste e Acompanhamento Manual
- Avaliação *short-circuit*
- Lei de De Morgan
- Métodos para Teste de Caracteres
- Validação da Entrada
- Exemplos



# Comparação de Números de Ponto-flutuante

- Alguns números de ponto-flutuante NÃO podem ser representados em binário de forma exata, pois, na sua conversão para binário, surge uma dízima periódica que acabará sendo interrompida
- Com valores reais, há risco de erros de representação e de arredondamento que podem levar a resultados inesperados

```
double umTerco = 1.0/3.0;
if ( umTerco != 0.333333333333 )
    System.out.println("Um terço NÃO é 0,333333333333");
// RESULTADO: Um terço NÃO é 0,333333333333

double valor = 4.35, total = 4.35 * 100;
int reais = (int) total;
System.out.println("reais="+reais);
// RESULTADO: reais=434

double r = Math.sqrt(2.0);
if (r*r != 2.0)
    System.out.printf("Math.sqrt(2.0)^2 == %.20f\n", r*r);
// RESULTADO: Math.sqrt(2.0)^2 == 2,00000000000000040000
```

# Comparação de Números de Ponto-flutuante: Solução

- Define-se um erro mínimo aceitável ( $\text{EPSILON}$ ), abaixo do qual se considera que dois valores de ponto-flutuante “sejam iguais”
  - O valor absoluto da diferença entre os dois valores deve ser menor do que determinado limite
  - Matematicamente, diz-se que  $x$  e  $y$  estão suficientemente próximos se  $|x - y| < \varepsilon$

```
final double EPSILON = 1E-14;  
double r = Math.sqrt(2.0);  
if (Math.abs(r * r - 2.0) < EPSILON)  
    System.out.printf("Math.sqrt(2.0)^2 == 2.0");
```

# Expressões condicionais

- O **operador ternário** (?:) permite inserir uma decisão em uma expressão
- Por exemplo, em vez de fazer:

```
if ( andar < MAIOR_ANDAR )  
    novoAndar = andar + 1;  
else  
    novoAndar = andar;
```

pode-se fazer:

```
novoAndar = andar < MAIOR_ANDAR ? andar + 1 : andar;
```

- A sintaxe das expressões condicionais é a seguinte:  
condição ? expressãoParaV : expressãoParaF

# Expressões condicionais: Exemplos

*// Exemplo 1*

```
int inteiro = in.nextInt();  
System.out.println( (inteiro%2 == 0) ? "Par" : "Impar" );
```

*// Exemplo 2*

```
taxaDesconto = (preco < 128) ? 0.92 : 0.84;  
precoComDesconto = taxaDesconto * preco;
```

*// OU*

```
precoComDesconto = ( (preco < 128) ? 0.92 : 0.84 ) * preco;
```

# Casos de Teste e Acompanhamento Manual

- É importante prever os ramos de decisão que podem ocorrer na execução de um programa
- Por exemplo:
  - Estado do aluno (aprovado, em G2, reprovado)
  - No cálculo da *bhaskara*: se é uma equação do segundo grau e quais os valores possíveis para delta (negativo, zero, positivo)
- Deve-se criar casos de teste para:
  - Cada ramo de decisão
  - Todos os limites dos ramos de decisão (a que ramo pertence cada valor que corresponde ao limite?...)
  - Entradas inválidas

# Acompanhamento Manual (ou Teste de Mesa)

- Acompanhar a execução manualmente ajuda a entender se o programa está funcionando corretamente ou não
- Crie uma tabela com as variáveis mais importantes
  - Use lápis e papel para acompanhar os valores destas variáveis
- Pode ser feito com pseudocódigo ou até mesmo código Java
  - Você pode marcar a execução no código com um *clips*
- Use valores de entrada:
  - Para os quais você já sabe os resultados que serão produzidos
  - Que testem todos os ramos possíveis do seu código

## Avaliação *short-circuit*: &&

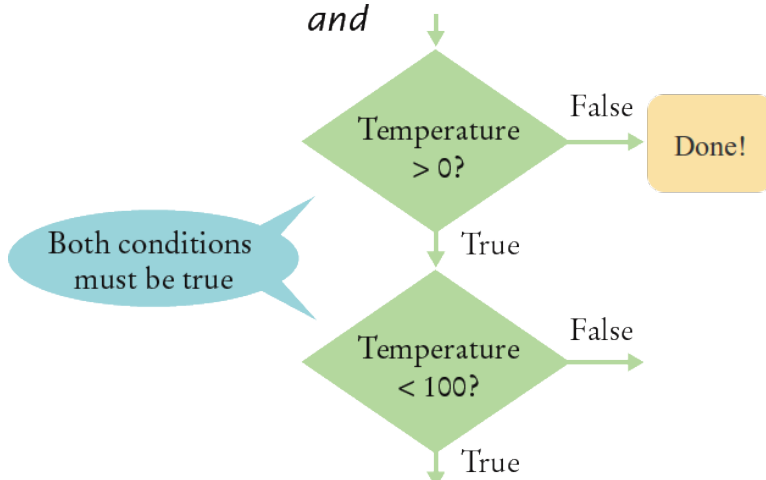
- Condições combinadas são avaliadas da esquerda para a direita
- No caso do E-lógico (*and*), se uma das condições avaliadas for falsa, por que continuar avaliando as demais?

```
if ( temperatura > 0 && temperatura < 100 ) {  
    System.out.println("A água está líquida.");  
}
```

- Um exemplo útil:

```
if ( quantidade > 0 && preco / quantidade < 10.0)
```

# Avaliação *short-circuit*: & &



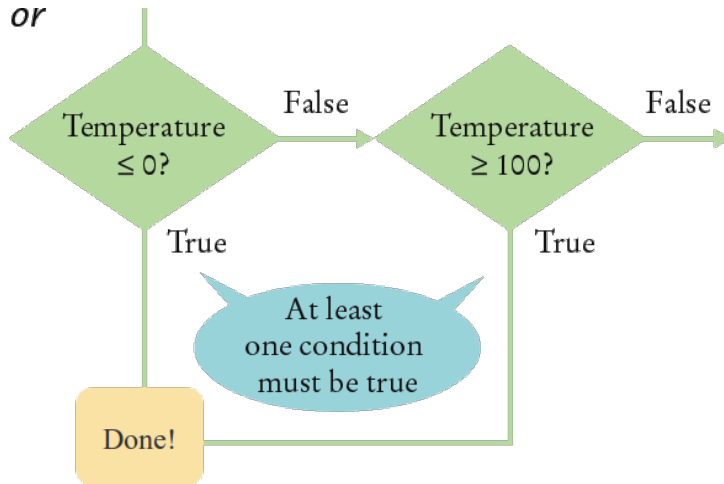


## Avaliação *short-circuit*: ||

- No caso do OU-lógico (*or*), se alguma condição à esquerda for verdadeira, por que continuar avaliando as demais?

```
if ( temperatura <= 0 || temperatura >= 100 ) {  
    System.out.println("A água NÃO está líquida.");  
}
```

# Avaliação *short-circuit*: | |



# Lei de De Morgan

- A Lei de De Morgan diz como negar condições `&&` e `||`
  - `!(A && B)` é o mesmo que `!A || !B`
  - `!(A || B)` é o mesmo que `!A && !B`
- Exemplo: Envio para AK e HI é mais caro

```
if ( !(pais.equals("USA") && !estado.equals("AK") && !estado.equals("HI")) )  
    custoDeEnvio = 20.00;
```

```
if ( !pais.equals("USA") || estado.equals("AK") || estado.equals("HI") )  
    custoDeEnvio = 20.00;
```

- Para simplificar condições com negações de condições AND ou OR, geralmente é uma boa ideia aplicar a Lei de De Morgan para mover as negações para o nível mais interno.

# Métodos para Teste de Caracteres

- A classe `Character` tem muitos métodos para teste de caracteres

Método	Exemplos de Caracteres Aceitos
<code>isDigit</code>	0, 1, 2
<code>isLetter</code>	A, B, C, a, b, c
<code>isLetterOrDigit</code>	A, a, B, b, 0, 1
<code>isUpperCase</code>	A, B, C
<code>isLowerCase</code>	a, b, c
<code>isWhiteSpace</code>	espaço, nova linha, tabulação
etc.	...

- Pode-se usar, por exemplo:

```
if (Character.isDigit(ch)) {
    System.out.printf("%c é um dígito.\n", ch);
}
```

# Validação da Entrada

- Aceitar entrada do usuário é perigoso
- O que acontece quando se usa `nextInt()` para ler valores inteiros e o usuário fornece uma palavra ou valor inválido?
- O método `hasNextInt()` da classe `Scanner` pode ajudar nesse caso
  - `hasNextInt()` retorna `true` se o valor for um inteiro válido, ou `false` em caso contrário

```
if (in.hasNextInt()) {  
    int inteiro = in.nextInt();  
    // ... processamento do valor inteiro lido  
}  
else {  
    System.out.println("Nenhum valor inteiro válido foi fornecido...");  
}
```

- Há outros métodos semelhantes: `hasNextDouble()`, `hasNext()`, `hasNextLine()`, etc.

# Exemplo: SimuladorDeElevador.java [Adaptado de Horstmann (2013, p. 84-85)]

```
import java.util.Scanner;

/**
 * Este programa simula um painel de elevador que pula o andar 13.
 */
public class SimuladorDeElevador {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Andar: ");
        int andar = in.nextInt();
        // Ajusta o andar se for necessário
        int novoAndar;
        if (andar > 13) {
            novoAndar = andar - 1;
        }
        else {
            novoAndar = andar;
        }
        System.out.println("O elevador vai se deslocar até o andar " + novoAndar);
        in.close();
    }
}
```

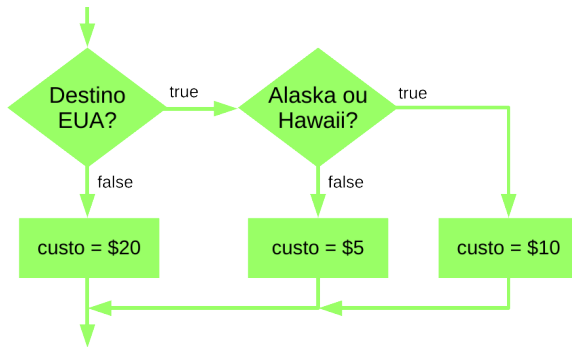
# Exemplo: SimuladorDeElevador2.java [Adaptado de Horstmann (2013, p. 117)]

```
import java.util.Scanner;

/**
 * Este programa simula um painel de elevador que pula o andar 13, verificando erros de digitação.
 */
public class SimuladorDeElevador2 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Andar: ");
        if ( in.hasNextInt() ) { // hasNextInt() testa se o usuário digitou um valor inteiro
            int andar = in.nextInt(); // nextInt() será executado com a certeza de que há um inteiro válido na entrada
            if (andar == 13) {
                System.out.println("Erro: NÃO há décimo terceiro andar.");
            }
            else if (andar <= 0 || andar > 20) {
                System.out.println("Erro: O número do andar deve variar de 1 a 20.");
            }
            else { // Agora se tem certeza de que o número de andar é válido
                int novoAndar = andar;
                if (andar > 13) { novoAndar = andar - 1; }
                System.out.printf("O elevador se deslocará para o andar real %d.\n", novoAndar);
            }
        }
        else
            System.out.println("Erro: Nenhum valor inteiro encontrado.");
        in.close();
    }
}
```

## Exemplo: Custo de Postagem nos EUA

- O custo de envio interno nos EUA é de \$5, exceto para Hawaii e Alaska para onde o custo é de \$10. O custo internacional é de \$20.





# Exemplo: Custo de Postagem nos EUA

```
import java.util.Scanner;

public class PostagemEUA {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Envio para EUA (S/N)? ");
        String resp = in.next().toUpperCase();
        double custo;
        if (resp.equals("S")) {
            System.out.println("Alaska ou Hawaii (S/N)? ");
            resp = in.next().toUpperCase();
            if (resp.equals("S"))
                custo = 10.0;
            else
                custo = 5.0;
        }
        else
            custo = 20.0;
        System.out.printf("Custo de postagem: $%.2f\n", custo);
        in.close();
    }
}
```

## Exemplo: Imposto de Renda nos EUA

O imposto de renda anual nos EUA considera basicamente 2 alternativas iniciais: contribuição para solteiros e contribuição para casados. Para cada uma destas possibilidades o imposto é calculado de forma diferente, conforme o ganho individual ou do casal.

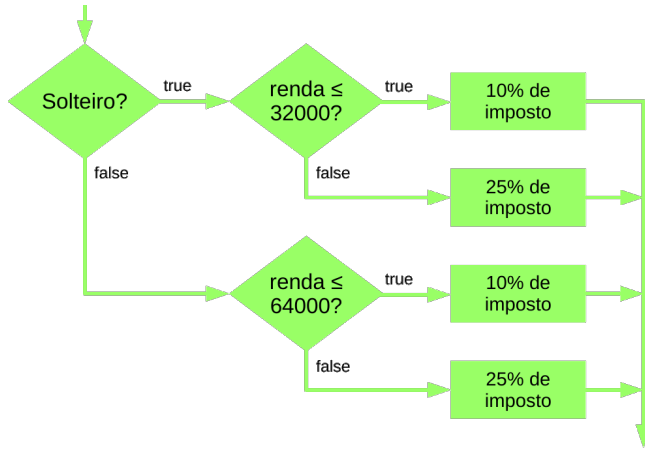
- Solteiro

- Renda menor ou igual a \$32.000,00  
taxa = 10%
- Renda maior do que \$32.000,00  
taxa = \$3.200,00 + 25% sobre o que exceder \$32.000,00

- Casado

- Renda conjunta menor ou igual a \$64.000,00  
taxa = 10%
- Renda conjunta maior do que \$64.000,00  
taxa = \$6.400,00 + 25% sobre o que exceder \$64.000,00

# Exemplo: Imposto de Renda nos EUA



# Exemplo: Imposto de Renda nos EUA

```
import java.util.Scanner;

public class ImpostoEUA {
    public static void main(String[] args) {
        final double PERCENTUAL1 = 0.10, PERCENTUAL2 = 0.25;
        final double LIMITE_SOLTEIRO = 32000, LIMITE_CASAL = 64000;
        Scanner in = new Scanner(System.in);
        System.out.print("Informe a sua renda: ");
        double imposto, renda = in.nextDouble();
        System.out.print("Digite S para solteiro ou C para casado: ");
        String estadoCivil = in.next().toUpperCase();
        if (estadoCivil.equals("S")) {
            if (renda <= LIMITE_SOLTEIRO)
                imposto = PERCENTUAL1 * renda;
            else
                imposto = PERCENTUAL1 * LIMITE_SOLTEIRO + PERCENTUAL2 * (renda - LIMITE_SOLTEIRO);
        }
        else {
            if (renda <= LIMITE_CASAL)
                imposto = PERCENTUAL1 * renda;
            else
                imposto = PERCENTUAL1 * LIMITE_CASAL + PERCENTUAL2 * (renda - LIMITE_CASAL);
        }
        System.out.printf("O imposto é de $%.2f\n", imposto);
        in.close();
    }
}
```

## Referências

# Referências

HORSTMANN, C. **Java for Everyone – Late Objectt.** 2. ed. Hoboken: Wiley, 2013. xxxiv, 589 p.