

# Estruturas de Controle

Mário Meireles Teixeira  
UFMA – DEINF



## Tópicos

- Estruturas de Controle
  - condicionais
  - repetição
- Expressões Lógicas
  - operadores relacionais
  - operadores lógicos

# Introdução



- O **fluxo de controle** de um programa determina o próximo comando a ser executado
- Fluxo de controle puramente **seqüencial**
  - Próximo comando na seqüência
- Fluxo de controle **condicional**
  - Comandos de seleção
- Fluxo de controle **iterativo**
  - Comandos de repetição
  - Comandos de desvio
- Bohm e Jacopini (1966) demonstraram que qualquer programa pode ser escrito baseado **somente** nos três tipos de estruturas de controle acima (seqüência, seleção e repetição) → Fim do "goto"

# true e false



- Todas as expressões condicionais usadas nas estruturas de controle são expressões booleanas
  - O resultado das expressões deve ser sempre **true** ou **false**
  - Não há conversões automáticas envolvendo booleanos em Java (evita erros de programação comuns em C/C++)

Código errado.  
Não compila em Java

```
int x = 10;  
if (x = 5) {  
    ...  
}
```

código aceito em C/C++  
(mas provavelmente errado)  
x, com valor 5, converte-se em 'true'.

Código correto.  
x == 5 é expressão com resultado true ou false

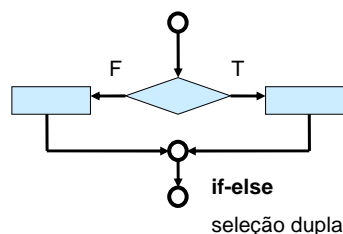
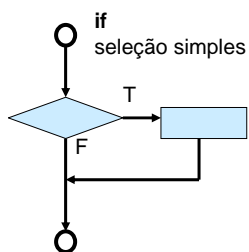
```
int x = 10;  
if (x == 5) {  
    ...  
}
```

## Estruturas de Controle



- Estruturas de **seleção**
  - **if, if-else, switch**
- Estruturas de **repetição**
  - **for, while, do-while**
- Desvios (*somente em estruturas de repetição*)
  - **continue, break**, rótulos
- Java não possui um comando **goto**, porém “goto” é uma palavra reservada

## Controle Condicional: Seleção simples e dupla



if-else  
seleção dupla

## Expressões Booleanas



- Uma condição usa **operadores relacionais** e resulta em um valor booleano (**true** ou **false**):

|    |                          |
|----|--------------------------|
| == | equal to                 |
| != | not equal to             |
| <  | less than                |
| >  | greater than             |
| <= | less than or equal to    |
| >= | greater than or equal to |

7

## Operadores Lógicos



- Operadores **&&** (and), **||** (or) e **!** (not)
- Exemplo de tabela verdade:

| a     | b     | a && b | a    b |
|-------|-------|--------|--------|
| true  | true  | true   | true   |
| true  | false | false  | true   |
| false | true  | false  | true   |
| false | false | false  | false  |

Estruturas de Controle

8

## Exemplo if-else



```
if ( ano < 0 ) {
    System.out.println("Não é um ano!");
} else if ( (ano%4==0 && ano%100!=0) || (ano%400==0) ) {
    System.out.println("É bissexto!");
} else {
    System.out.println("Não é bissexto!");
}
```

```
if(nota >= 9.0)
    System.out.println('A');
else if (nota >= 8.0)
    System.out.println('B');
else if (nota >= 7.0)
    System.out.println('C');
else
    System.out.println('D');
```

```
if (preco < 400)
    if (preco < 300)
        System.out.println("Menor que 300");
    else // OK ??
        System.out.println("Maior que 400");
```

9

## O operador condicional ? :

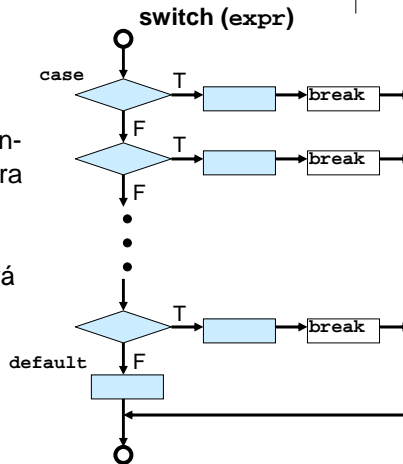


- Mesma semântica da estrutura **if-else**
- Único operador ternário da linguagem Java
  - o primeiro operando é uma expressão booleana
  - retorna o segundo operando se a expressão condicional for **true** ou o terceiro operando se for **false**
- Exemplos:

```
System.out.println( (media >= 7) ? "Aprovado" : "Reprovado" );
int x = (y != 0) ? 50 : 500;
String titulo = (sex == 'f') ? "Sra." : "Sr." ;
```

## Seleção múltipla: switch

- Avalia uma expressão de números inteiros e tenta selecionar um rótulo (*case*):
  - Caso haja um **case** correspondente, transfere o controle para a primeira instrução após o rótulo
  - Caso contrário, o controle será transferido para a primeira instrução após o rótulo **default**
  - **break** provoca a saída do comando **switch**



## Sintaxe da instrução switch

```
switch(expr1) {  
    case expr2:  
        instruções;  
        break;  
    case expr3:  
        instruções;  
        break;  
    default:  
        instruções;  
}
```

Na instrução **switch (expr1)**, **expr1** deve ser compatível com atribuições do tipo **int**, sendo permitidos os tipos de dados **short**, **int**, **byte** ou **char**, pois pode ser feita a coerção (*cast*) para o tipo **int**, o que não é possível com os tipos **long** ou de **ponto flutuante**.

## Exemplo switch



```
String s = "xyz";
switch (s.length()) {
  case 1: System.out.println("tamanho = 1");
          break;
  case 2: System.out.println("tamanho = 2");
          break;
  case 3: System.out.println("tamanho = 3");
          break;
  default:
}

```

```
switch(letra) {
  case 'A' : System.out.println("A");
             break;
  case 'B' : System.out.println("B");
             break;
  ...
  default  : System.out.println("?");
}

```

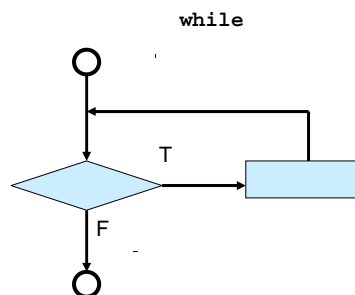
Estruturas de Controle

13

## Estrutura de repetição: while



- Especifica uma ação que será repetida enquanto uma condição for verdadeira



```
while (expressão booleana)
{
  instruções;
}

```

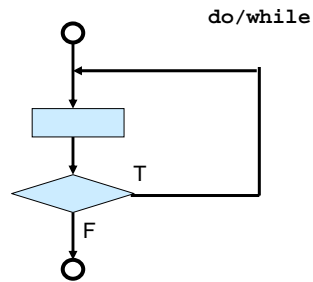
Estruturas de Controle

14



## Estrutura de repetição: do-while

- O loop será executado pelo menos uma vez, pois a condição só é testada no final



Estruturas de Controle

```
do
{
    instruções;
} while (expressão booleana);
```

15



## Exemplos while/do-while

```
int x = 0;
while (x < 10) {
    System.out.println ("item " + x);
    x++;
}
```

```
int x = 0;
do {
    System.out.println ("item " + x);
    x++;
} while (x < 10);
```

```
// loop infinito
while ( true ) {
    ...
}
```

Estruturas de Controle

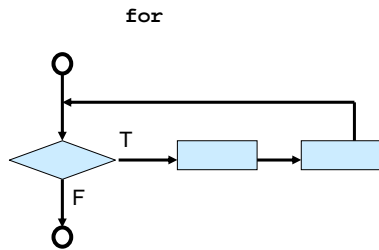
16



## Estrutura de repetição: for



- Repete um bloco de comandos um número pré-determinado de vezes. Normalmente está associada a algum contador



```
for ( inicialização ;  
      expressões booleanas ;  
      passo da repetição )  
{  
  instruções ;  
}
```

Estruturas de Controle

17

## Exemplos for



```
for ( int x = 0; x < 10; x++ ) {  
  System.out.println ("item " + x);  
}
```

```
for ( int x = 0, int y = 25;  
      x < 10 && (y % 2 == 0);  
      x++, y = y - 1 ) {  
  System.out.println (x + y);  
}
```

```
// loop infinito  
for ( ; ; ) {  
  ...  
}
```

Estruturas de Controle

18

## Comandos de desvio: break e continue



- **break**: interrompe a execução do bloco de repetição
  - Continua com a próxima instrução após o bloco
- **continue**: interrompe a iteração atual
  - Ignora as instruções restantes no bloco, testa a condição e reinicia com a próxima iteração

```
while (!terminado) {
    passePagina();
    if (alguemChamou == true) {
        break; // caia fora deste loop
    }
    if (paginaDePropaganda == true) {
        continue; // pule esta iteração
    }
    leia();
}
restoDoPrograma();
```

Estruturas de Controle

19

## Exemplo break



```
public static void main( String args[] ) {
    int count;
    for ( count = 1; count <= 10; count++ ) {
        if ( count == 5 )
            break;
        System.out.print(count + " ");
    } // fim for
    System.out.println("\nValor de count: " + count);
} // fim main
```

1 2 3 4

Valor de count: 5

Estruturas de Controle

20

## Exemplo continue



```
public static void main( String args[] ) {
    final int SKIP = 5; // constante
    for (int count = 1; count <= 10; count++ ) {
        if ( count == SKIP )
            continue;
        System.out.print(count + " ");
    } // fim for
    System.out.println("\nNumero " + SKIP +
        " nao foi impresso");
} // fim main
```

1 2 3 4 6 7 8 9 10

Numero 5 nao foi impresso

Estruturas de Controle

21

## Usando rótulos



- **break** e **continue** sempre atuam sobre o bloco de repetição onde são chamados
- Em blocos de repetição contidos em outros blocos, pode-se usar **rótulos** para fazer **break** e **continue** atuarem em blocos externos
- Os rótulos só podem ser usados antes de **do**, **while** e **for**
- As chamadas só podem ocorrer dentro de blocos de repetição

```
revista: while (!terminado) {
    for (int i = 10; i < 100; i += 10) {
        passePagina();
        if (textoChato) {
            break revista;
        }
    }
    maisInstrucoes();
}
restoDoPrograma();
```

break sem rótulo quebraria aqui!

```
ident: do {...}
      ou
ident: while () {...}
      ou
ident: for () { ...}
```

Estruturas de Controle

22