



## Aula 3

# Instruções Condicionais

Programação em Java 2006-2007



## Instruções de Selecção

Instruções que permitem o controlo da execução, de modo a que o programa possa fazer escolhas

*O Java, tal como a maior parte das linguagens de alto nível apresenta três variantes de estruturas de selecção:*

- Selecção simples (a instrução **if**)
- Selecção em alternativa (a instrução **if-else**)
- Selecção múltipla (a instrução **switch-case**)

Programação em Java 2006-2007

3



## Seleccção Simples

Permite decidir entre executar ou não um conjunto de instruções em função do resultado de uma condição

```
if (condição) {  
    instruções;  
}
```

condição := expressão lógica



## Seleccção em alternativa

Escolhe entre dois conjuntos de instruções qual deve ser executado em função do resultado de uma condição

```
if (condição) {  
    instruções se condição verdadeira;  
}  
else {  
    instruções se condição falsa;  
}
```

## Expressão Lógica

O resultado da avaliação de uma expressão lógica é um valor lógico (booleano):  
**true** (verdadeiro) ou **false** (falso)

*As expressões lógicas constroem-se mediante:*

- Operadores relacionais
- Operadores lógicos

## Operadores Relacionais

Operam sobre operandos  
(*constantes, variáveis, expressões*)  
de diferentes tipos  
(**numéricos, caracteres**)  
e como resultado retornam um  
valor de tipo lógico

Operador	Operação
==	igual a
!=	diferente de
<	menor que
<=	menor ou igual
>	maior que
>=	maior ou igual

*Exemplos*

a	b	a == b	a != b	a < b	a <= b	a > b	a >= b
2	3	false	true	true	true	false	false
'a'	'd'	false	true	true	true	false	false
33	33	true	false	false	true	false	true

## Precedência dos Operadores Relacionais

- **( )** sub-expressões entre parêntesis avaliam-se primeiro  
(maior nível de precedência)
- **<, <=, >, >=, ==, !=** depois  
(menor nível de precedência)

Exemplo:

Para  $a = 31$ ,  $b = -1$ ,  $x = 3$ ,  $y = 2$

$$(a + b - 1) < (x * y)$$

$$R1 = a + b$$

$$R2 = R1 - 1$$

$$R3 = x * y$$

$$R4 = R2 < R3 \rightarrow \text{resultado}$$

$$(a + b - 1) < (x * y)$$

$$R1 = 31 + (-1) = 30$$

$$R2 = 30 - 1 = 29$$

$$R3 = 3 * 2 = 6$$

$$R4 = 29 < 6 = \text{falso} \rightarrow \text{resultado}$$

## Exercício (I)

Determine o resultado das seguintes expressões lógicas:

1.  $(a + b / c) \geq (x - y \% c)$   
para  $a=3$ ,  $b=6$ ,  $c=2$ ,  $x=1$  e  $y=4$
2.  $a - b != x + y$   
para  $a=10$ ,  $b=1$ ,  $x=5$  e  $y=4$
3.  $(a + b \% c) \leq (x * y) + (y \% 2)$   
para  $a=2$ ,  $b=7$ ,  $c=3$ ,  $x=2$ ,  $y=1$

## Operadores Lógicos

Operam sobre operandos  
(*constantes, variáveis, expressões*)

**de tipo lógico**

e retornam um valor de tipo lógico

Operador	Operação
<b>&amp;&amp;</b>	conjunção (AND)
<b>  </b>	disjunção (OR)
<b>!</b>	Negação (NOT)

a	b	a && b	a    b	! a	! b
true	true	true	true	false	false
true	false	false	true	false	true
false	true	false	true	true	false
false	false	false	false	true	true

Programação em Java 2006-2007

9

## Exemplos de Operadores Lógicos

### Exemplo do operador AND

```
if ((x <= 5) && (x >= 1))  
    println("x está entre 1 e 5");  
else  
    println("x e maior que 5 ou menor que 1");
```

### Exemplo do operador OR

```
if ((x > 5) || (x < 1))  
    println("x e maior que 5 ou menor que 1");  
else  
    println("x está entre 1 e 5");
```

Programação em Java 2006-2007

10

## Precedência dos Operadores Lógicos

- **( )** sub-expressões entre parêntesis avaliam-se primeiro  
(maior nível de precedência)
- **&&, ||, !** depois  
(menor nível de precedência)

Exemplo:

`!(x < 5) && !(y >= 7)`

R1 = `x < 5`

R2 = `y >= 7`

R3 = `!R2`

R4 = `!R1`

R5 = `R4 && R3` → Resultado

Para `x = 3` e `y = 6`

`!(x < 5) && !(y >= 7)`

R1 = `3 < 5` → verdadeiro

R2 = `6 >= 7` → falso

R3 = `!falso` → verdadeiro

R4 = `!verdadeiro` → falso

R5 = `verdadeiro && falso` = **falso**

## Exercício (II)

Determine o resultado das seguintes expressões lógicas:

1. `!((i > 4) || (j <= 6))`  
para `i=10` e `j=3`
2. `!(a + b == c) || ((c != 0) && (b - c >= 19))`  
para `a=34`, `b=12` e `c=8`

## Exercício (III)

Qual a mensagem que aparece no ecrã do monitor?

```
int x = 0, y = 1, z = 2;
if (((x != 0) && (y == 1)) || (z != 0))
    print("Ganhou");
else
    print("Perdeu");
println(" o prémio");
```

- a. Ganhou   b. Perdeu   c. Ganhou o prémio   d. Perdeu o prémio.

## Encadeamento de vários if-else

Escolhe entre mais de duas alternativas qual conjunto de instruções deve ser executado

```
if (condição1) {
    instruções se condição1 verdadeira;
} else {
    if (condição2) {
        instruções se condição2 verdadeira;
    } else {
        if (condição3) {
            instruções se condição3 verdadeira;
            ...
        } else {
            instruções se condição falsa;
        }
    }
}
```

## Exemplo 1

Escrever um programa que determine o tipo de bilhete que cada visitante de um parque de diversões deve comprar segundo a idade do visitante, de acordo com a tabela seguinte:

Idade	Bilhete
Inferior a 6	Isento de pagamento
Entre 6 e 12	Bilhete de criança
Entre 13 e 65	Bilhete normal
Mais de 65	Bilhete de 3ª idade

## Exemplo 1. Bilhete (1)

Quando usamos apenas uma instrução por bloco podemos tirar as chavetas, e assim o programa fica menos confuso

```
if (idade <= 5){           // terá menos de 6 anos?
    println("Isento de pago");
} else {                  // terá entre 6 e 12 anos?
    if (idade <= 12){
        println("Bilhete de criança");
    } else {              // terá entre 12 e 65 anos?
        if (idade <= 65){
            println("Bilhete normal");
        } else {          // terá mais do que 65 anos?
            println("Bilhete 3ª idade");
        } // fecha bloco se idade > 65
    } // fecha bloco se idade > 12
} // fecha bloco se idade > 5
```

## Exemplo 1. Bilhete (2)

```
import p1.*;
public class Ex_3_1_Bilhete extends P1App {
    public static void main(String args[]) {
        int idade;
        idade = readInt("Introduza a idade do visitante: ");
        if (idade <= 5) // terá menos de 6 anos?
            println("Isento de pago");
        else if (idade <= 12) // terá entre 6 e 12 anos?
            println("Bilhete de criança");
        else if (idade <= 65) // terá entre 12 e 65 anos?
            println("Bilhete normal");
        else // terá mais do que 65 anos?
            println("Bilhete 3ª idade");
    }
}
```

Programação em Java 2006-2007

17

## Exemplo 2

Dado um número do mês retorna qual a estação do ano

```
import p1.*;
public class Ex_3_2_EstAno extends P1App {
    public static void main(String args[]) {
        int mes;
        mes = readInt("Introduza o número do mês do ano: ", 0, 12);
        if (mes == 12 || mes == 1 || mes == 2)
            println("A estação do ano é o Inverno");
        else if (mes == 3 || mes == 4 || mes == 5)
            println("A estação do ano é a Primavera");
        else if (mes == 6 || mes == 7 || mes == 8)
            println("A estação do ano é o Verão");
        else if (mes == 9 || mes == 10 || mes == 11)
            println("A estação do ano é o Outono");
    }
}
```

Em vez de usar **if-then** podemos usar a estrutura de selecção múltipla **switch-case**

Programação em Java 2006-2007

18

## Seleccção Múltipla: Switch-Case

Quando a escolha for feita em função do valor de uma expressão **inteira** ou **carácter**, é mais eficaz utilizar uma estrutura de switch-case

**break** provoca o fim da instrução, evitando que as instruções das outras opções sejam executadas

```
switch (expressão) {  
  case valor_1:  
    instruções;  
    break;  
  ...  
  case valor_n:  
    instruções;  
    break;  
  default:  
    instruções;  
    break;  
}
```

**default** é opcional. Caso exista, será executada se nenhum dos valores for igual ao resultado da expressão

Programação em Java 2006-2007

19

## Exemplo 2

Podemos re-escrever o programa usando **switch-case**

```
switch (mes) {  
  case 12:  
  case 1:  
  case 2:  
    println("A estação do ano é o Inverno");  
    break;  
  case 3:  
  case 4:  
  case 5:  
    println("A estação do ano é a Primavera");  
    break;  
  case 6:  
  case 7:  
  case 8:  
    println("A estação do ano é o Verão");  
    break;  
  case 9:  
  case 10:  
  case 11:  
    println("A estação do ano é o Outono");  
    break;  
}
```

Programação em Java 2006-2007

20

## Implementação de Menus

A instrução **switch-case** pode ser usada para implementar um menu com várias opções a escolher pelo utilizador. As opções podem ser números ou caracteres.

```
int op
println("Menu");
println(" 1 - primeira opção");
println(" 2 - segunda opção");
....
println(" n - n-esima opção");
op = readInt("Selecione uma opção ", 1, n);
switch (op) {
    case 1:
        println("Selecionou a opção 1");
        break;
    ....
    case n:
        println("Selecionou a opção n");
        break;
}
```

## Problema 1

Escrever um programa que lê do teclado uma data composta pelo nº de mês e ano, calcula e imprime no ecrã o nº de dias desse mês.

Implementar duas versões:

uma usando **if-then** e a outra usando **switch-case**

Ano bissexto: de 4 em 4 anos, com excepção dos finais de século, que são bissextos de 4 em 4 séculos. Por exemplo, os anos 1600, 1996, 2000, 2004 são bissextos, mas os anos 1700, 1800 ou 1900 não o são.



## Problema 2

---

Escrever um programa que permita ao utilizador seleccionar de um menu uma operação aritmética (*soma, resta, multiplicação ou divisão*), introduzir dois valores para os operandos, e que calcule e mostre o resultado da operação seleccionada.