



Estruturas de Controle da Linguagem Java

Alberto Costa Neto
DComp - UFS



Roteiro

- Comandos Condicionais
 - Condicionais Simples e Compostas
 - Estrutura Case
 - Estrutura Ternária
 - Comandos de Repetição
 - While
 - Do While
 - For
 - Controle de Fluxo
 - Escopo de Variável
-



Comandos Condicionais

- O que você entende deste código???

```
int idade = 15;
boolean amigoDoDono = true;
if ((idade < 18) && !amigoDoDono)
    System.out.println("Não pode entrar");
else
    System.out.println("Pode entrar");
```



Comandos Condicionais

- Permitem **selecionar** um ou mais **comandos** a executar, a **depende**r de uma ou mais **condições**



Comandos Condicionais

■ Condicional Simples

```
if (expr)
    comando; // Executado quando expr retorna true
```

```
if (expr) {
    // Quando a expr retorna true, os comandos que
    // estão entre { e } são executados
    comando1;
    comando2;
}
```

Comandos Condicionais

■ Condicional Composta

```
if (expr)
    comando1; // Executado quando expr retorna true
else
    comando2; // Executado quando expr retorna false
```

```
if (expr) {
    comando1; // Executados quando expr
    comando2; // retorna true
} else {
    comando3; // Executados quando expr
    comando4; // retorna false
}
```



Comandos Condicionais

- Problema de *dangling-else*
 - O comando do "else" será executado em qual situação?

```
if (X > 5)
    if (Y > 5)
        System.out.println ("X e Y são maiores que 5");
else
    System.out.println ("X é menor ou igual a 5");
```



Comandos Condicionais

- Problema de *dangling-else*
 - Solução

```
if (X > 5)
{
    if (Y > 5)
        System.out.println ("X e Y são maiores que 5");
}
else
    System.out.println ("X é menor ou igual a 5");
```




Dever de Sala

- 1) Faça um programa em Java que leia 3 notas de um aluno, calcule e mostre a média aritmética e a mensagem constante na tabela a seguir.

$0,0 \leq \text{media} < 3,0$	Reprovado
$3,0 \leq \text{media} < 7,0$	Prova Final
$7,0 \leq \text{media} \leq 10,0$	Aprovado

Estrutura Case

■ Sintaxe

```
switch (expr) {  
    case valor1:  
        lista comandos1; break;  
    case valor2:  
        lista comandos2; break;  
    default:  
        lista comandos3;  
}
```

- expr deve ser do tipo **byte**, **short**, **int**, **long** ou **char**
- O comando **break** impede a execução dos comandos subsequentes.
- Quando não há correspondências nos valores dos cases o comando **default** é executado.



Dever de Sala

- 2) Faça um programa em Java que recebe a idade de um nadador e mostre sua categoria, usando as regras a seguir:

0 a 10	Infantil
11 a 15	Adolescente
16 a 30	Jovem
30 a 45	Adulto
Acima de 45	Sênior



Estrutura Ternária

- Operador Ternário ou Condicional

Operador	Sintaxe	Descrição	Exemplo	Resultado
?	<code>X ? Y : Z</code>	Quando a expressão antes do ? É true, retorna Y. Caso contrário, retorna Z	<pre>X = (exp1 ? 'S' : 'N') Y = (exp2 ? 'S' : 'N')</pre>	<pre>X = 'S' Y = 'N'</pre>

* Considere exp1 uma expressão que retorna true e exp2 uma que retorna false



Dever de Sala

- 3) Crie um programa em Java usando a estrutura ternária condicional que recebe dois números e mostra o maior.



Roteiro

- Comandos Condicionais
 - Condicionais Simples e Compostas
 - Estrutura Case
 - Estrutura Ternária
 - **Comandos de Repetição**
 - **While**
 - **Do While**
 - **For**
 - Controle de Fluxo
 - Escopo de Variável
-

Comandos de Repetição

■ Comando *While*

- Os comandos serão repetidos enquanto a condição assumir o valor **verdadeiro**.
- Utilizada normalmente quando o número de **repetições não é fixo**.
- Teste condicional ocorre no **início**.
- **Sintaxe:**

```
while (expr)
    comando;
```

```
while (expr) {
    comando1;
    comando2;
    ...
}
```



Comandos de Repetição

- O que será impresso?

```
int idade = 15;
while(idade < 18) {
    System.out.println(idade);
    idade = idade + 1;
}
```

```
int i = 0;
while(i < 10) {
    System.out.println(i);
    i = i + 1;
}
```


Comandos de Repetição

- Comando *Do While*

- Também utilizada quando o número de repetições **não é fixo**.
- Teste condicional **no fim**, assim, o bloco de **comando é executado** pelo menos **uma vez**.
- Sintaxe:

```
do {  
    comandos;  
} while (expr);
```



Comandos de Repetição

- O que será impresso?

```
x = 0;  
do {  
    System.out.println(x) ;  
    x = x + 1;  
} while (x != 5) ;
```



Comandos de Repetição

- O que será impresso?

```
for (int i = 0; i < 10; i = i + 1) {  
    System.out.println("olá!");  
}
```

Comandos de Repetição

■ Comando *For*

- Utilizado quando o número de vezes que o laço irá executar é conhecido
- Sintaxe:

```
for (inicialização; testeParaTérmino;  
      incremento) {  
    comando;  
}
```

- Os comandos contidos em **inicialização** são executados uma única vez e no início do laço.
- A expressão em **testeDeTérmino** é executada antes de cada iteração e ao retornar **false** o laço encerra.
- O **incremento** é executado após o final de cada iteração.



Comandos de Repetição

- Laços infinitos

```
while (true)  
    comando;
```

```
do  
    comando;  
while (true);
```

```
for (;;)   
    comando;
```



Roteiro

- Comandos Condicionais
 - Condicionais Simples e Compostas
 - Estrutura Case
 - Estrutura Ternária
 - Comandos de Repetição
 - While
 - Do While
 - For
 - Controle de Fluxo
 - Escopo de Variável
-

Controle de Fluxo

■ Comando return

```
// Retornar o primeiro múltiplo de 3 maior
// que i

public static int retornaMultiplo(int i) {
    while (true) {
        if (i % 3 == 0)
            return i;
        i++;
    }
}
```



Controle de Fluxo

- Comando return
 - Provoca a **saída** de um método
 - O fluxo de controle **retorna** ao **comando seguinte** à **chamada** do método
 - O comando return tem duas formas:
 - Com valor de retorno
 - Sem retornar valor (métodos = void)



Controle de Fluxo

- Comando break
 - Exemplo: para $X < Y$

```
for (int i = x; i < y; i++) {  
    if (i % 19 == 0) {  
        System.out.println("Achei um número  
            divisível por 19 entre X e Y");  
        break;  
    }  
}
```



Controle de Fluxo

- Comando break
 - Pode ser usado em **switch, for, while** e **do while**
 - **Provoca a saída** do laço ou switch mais interno no qual foi chamado



Controle de Fluxo

- Comando Continue
 - Quais números não serão impressos?

```
for (int i = 0; i < 100; i++) {  
    if ((i > 50) && (i < 60)) {  
        continue;  
    }  
    System.out.println(i);  
}
```



Controle de Fluxo

- Comando Continue
 - Faz com que um **laço continue** na próxima iteração, **ignorando o restante dos comandos** dentro do laço.
 - Ao contrário do break, **não encerra a execução** do laço.



Roteiro

- Comandos Condicionais
 - Condicionais Simples e Compostas
 - Estrutura Case
 - Estrutura Ternária
- Comandos de Repetição
 - While
 - Do While
 - For
- Controle de Fluxo
- **Escopo de Variável**



Escopo de Variável

- Trecho de código onde a variável existe e onde é possível acessá-la

```
//aqui a variável i não existe
```

```
int i = 18;
```

```
// a partir daqui ela existe
```



Escopo de Variável

- O escopo depende do bloco onde foi declarada

```
//aqui a variável i não existe  
int i = 5;  
// a partir daqui ela existe  
while (condicao) {  
    // o i ainda vale aqui  
    int j = 7;  
    // o j passa a existir  
}  
// aqui o j não existe mais, mas o i continua a valer
```



Escopo de Variável

- Algo de errado neste código???

```
if (algumBooleano) {  
    int i = 5;  
    System.out.println(i);  
} else {  
    int i = 10;  
    System.out.println(i);  
}
```


Escopo de Variável

- E agora???

```
if (algumBooleano) {  
    int i = 5;  
} else {  
    int i = 10;  
}  
System.out.println(i);
```



Solução?

Escopo de Variável

- Qual o resultado da execução?

```
for (int i = 0; i < 10; i++) {  
    System.out.println("olá!");  
}  
System.out.println(i);
```



Solução?



Dever de Sala

- 4) Leia dois números e imprima a soma de todos os números entre eles.
- 5) Leia um valor N , inteiro e positivo e mostre o valor de E conforme a expressão:

$$E = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$



Dever de Sala

- 6) Faça um programa que leia a idade e peso dos jogadores de um time, calcule e mostre:
 - a) a quantidade de jogadores com idade < 18 .
 - b) A média das idades dos jogadores
 - c) A percentagem de jogadores com mais de 80 quilos



Referências

- Slides “Estruturas Condicionais e Subrotinas” e “Estruturas de Repetição”. Prof. Marcos Dósea. UFS. 2010.
- Slides “Estruturas de Controle da Linguagem Java”, Prof^a. Débora. UFS. 2010
- Java How to Program
 - Capítulos – 4 e 5