



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

AULA:

Estruturas de decisão e repetição em Java

Programação Orientada a Objetos

Alba Lopes, Profa.

<http://docentes.ifrn.edu.br/albalopes>
alba.lopes@ifrn.edu.br

Estruturas de desvio condicional

- ▶ Desviam o código para um outro trecho, ao invés de prosseguir para a linha seguinte;
- ▶ Em Portugol, são as estruturas **se** e **escolha**

Portugol	Java
se	if
escolha	switch



Desvio condicional com if

```
if ([expressão])  
    [diretiva 1]  
else  
    [diretiva 2]
```

- ▶ **[expressão]**: expressão lógica avaliada (*deve retornar valor do tipo boolean*);
- ▶ **[diretiva 1]**: diretiva ou bloco de diretivas executadas se a condição retornar true;
- ▶ **[diretiva 2]**: diretiva ou bloco de diretivas executadas se a condição retornar false.



Portugol vs Java

Portugol

```
algoritmo "SituacaoAluno"  
var  
  nota: inteiro  
inicio  
  nota <- 70  
  se (nota >= 60) entao  
    escreval("Aprovado")  
  fimse  
  se (nota < 60) entao  
    escreval("Reprovado")  
  fimse  
fimalgoritmo
```

Java

```
public class SituacaoAluno {  
    public static void main(String[] args) {  
        int nota = 70;  
        if(nota >= 60) {  
            System.out.println("Aprovado!");  
        }  
        if(nota < 60) {  
            System.out.println("Reprovado");  
        }  
    }  
}
```



Encadeamento de ifs

- ▶ **if** é uma diretiva como qualquer outra;
- ▶ Podemos colocá-lo como [diretiva 2], logo após o **else** (executada quando expressão é false):

```
if ([expressão 1])
    [diretiva 1]
else if ([expressão 2])
    [diretiva 2]
else if ([expressão 3])
    [diretiva 3]
...
else
    [diretiva N]
```



Portugol vs Java

Portugol

```
algoritmo "SituacaoAluno"  
var  
  nota: inteiro  
inicio  
  nota <- 70  
  se (nota >= 60) entao  
    escreval("Aprovado")  
  senao  
    se (nota >= 20) entao  
      escreva("Recuperação")  
    senao  
      escreval("Reprovado")  
  fimse  
fimse  
fimalgoritmo
```

Java

```
public class SituacaoAluno {  
    public static void main(String[] args) {  
        int nota = 70;  
        if(nota >= 60) {  
            System.out.println("Aprovado!");  
        } else {  
            if(nota >= 20) {  
                System.out.println("Recuperação");  
            }else{  
                System.out.println("Reprovado");  
            }  
        }  
    }  
}
```



Exemplos

```
int x = 10, y = 15, z = 20;
boolean imprimir = true;

if ((x == 10) || (z > y)) {
    if (imprimir) System.out.println(x);
}
else if (x == 20) {
    z += x + y;
    if (imprimir) System.out.println(z);
}
else System.out.println("Não sei!");
```



Operador Ternário

- ▶ Forma simplificada de uma estrutura se- então;
- ▶ ***Produz um valor de acordo com uma expressão:***
 - ▶ `<expressão> ? <valor 1> : <valor 2>`
 - ▶ Se `<expressão>` for `true`, o resultado é `<valor 1>`, do contrário o resultado é `<valor 2>`.

```
int x = 7;  
int y = (x < 10) ? x * 2 : x / 2;  
System.out.println("y = " + y); // y = 14
```



Exercícios

- ▶ Crie um algoritmo que leia dois valores (x e y) e escreva na tela qual o maior valor.
- ▶ Crie um algoritmo que leia um valor inteiro representando a temperatura e informe se o clima está quente ou frio. Considere que o clima está quente se a temperatura for superior a 30. Caso contrário, informe que o clima está frio.
- ▶ Crie um algoritmo que leia um outro valor referente à temperatura e exiba uma mensagem informando o tipo do clima de forma mais detalhada, considerando as seguintes condições: se a temperatura estiver até 18 graus, o clima é frio; se a temperatura estiver entre 19 e 23 graus, o clima é agradável; se a temperatura estiver entre 24 e 35 graus, o clima é quente; se a temperatura estiver acima de 35 graus, o clima é muito quente.



Desvio condicional com switch

```
switch ([expressão ordinal]) {  
    case [valor ordinal 1]:  
        [diretiva 1]  
    break;  
    case [valor ordinal 2]:  
        [diretiva 2]  
    break;  
    ...  
    default:  
        [diretiva N]  
}
```



Desvio condicional com switch

- ▶ **[expressão ordinal]**: expressão que retorna um valor de *algum tipo discreto* (inteiro, char, etc.);
- ▶ **[valor ordinal X]**: um dos possíveis valores que a expressão ordinal pode assumir (deve ser do mesmo tipo);
- ▶ **[diretiva X]**: diretiva ou conjunto de diretivas (não é necessário abrir um bloco) executado se a expressão ordinal for igual ao [valor ordinal X].



Portugol vs Java

Portugol

```
algoritmo "dia_da_semana"
var
    dia: inteiro
inicio
    escreva("Digite um número de 1 a 7")
    leia(dia)
    escolha (dia)
    caso 1
        escreva("Domingo")
    caso 2
        escreva("Domingo")
    caso 3
        escreva("Domingo")
    caso 4
        escreva("Domingo")
    caso 5
        escreva("Domingo")
    caso 6
        escreva("Domingo")
    caso 7
        escreva("Domingo")
    outrocaso
        escreva("Dia inválido")
    fimescolha
fimalgoritmo
```

Java

```
import java.util.Scanner;

public class DiaDaSemana{
    public static void main(String [] args){
        Scanner teclado = new Scanner(System.in);
        int dia = teclado.nextInt();
        switch(dia){
            case 1:
                System.out.println("Domingo");
                break;
            case 2:
                System.out.println("Domingo");
                break;
            case 3:
                System.out.println("Domingo");
                break;
            case 4:
                System.out.println("Domingo");
                break;
            case 5:
                System.out.println("Domingo");
                break;
            case 6:
                System.out.println("Domingo");
                break;
            case 7:
                System.out.println("Domingo");
                break;
            default:
                System.out.println("Dia inválido");
        }
    }
}
```



Estruturas de repetição simples

- ▶ Repetição de um trecho de código;
- ▶ Número fixo de repetições (sabe-se de antemão quantas vezes o trecho será repetido);
- ▶ Java dispõe da diretiva **for** (sintaxe também herdada de C):

```
// Contar de 1 a 10:  
for (int i = 1; i <= 10; i++)  
    System.out.println(i);
```



Comando de repetição While

- ▶ Repetição de um trecho de código;
- ▶ Número indeterminado de repetições, depende de uma condição (expressão lógica);
- ▶ Equivalente ao comando de repetição enquanto do Portugol

```
while ( [expressão] ){  
    [diretiva]  
}
```



Comando de repetição While

Portugol

```
algoritmo "exemplo_enquanto"  
var  
    contador: inteiro  
inicio  
    contador <- 1  
    enquanto (contador <= 1000) faça  
        escreval("Algoritmos é muito bom!")  
        contador <- contador + 1  
    fimenquanto  
fimalgoritmo
```

Java

```
public class ExemploEnquanto{  
    public static void main(String [] args){  
        int contador = 1;  
        while(contador <= 1000){  
            System.out.println("Algoritmos é muito bom!");  
            contador <- contador + 1  
        }  
    }  
}
```



Comando de repetição do...while

- ▶ Repetição de um trecho de código;
- ▶ Número indeterminado de repetições, depende de uma condição (expressão lógica);
- ▶ Equivalente ao comando de repetição repetido Portugal

```
do {  
    [diretiva]  
}while( [expressão] );
```



Comando de repetição do...while

Portugol

```
algoritmo "ExemploRepita"  
var  
    contador: inteiro  
inicio  
    contador<- 1  
    repita  
        escreva("Suba um degrau")  
        contador <- contador + 1  
    ate (contador > 10)  
fimalgoritmo
```

Java

```
public class ExemploRepita{  
    public static void main(String [] args){  
        int contador = 1;  
        do{  
            System.out.println("Suba um degrau!");  
            contador <- contador + 1  
        } while(contador <= 10);  
    }  
}
```



Comando de repetição For

- ▶ Equivalente ao **para** do Portugol

```
for ([início]; [condição]; [inc/dec])  
    [diretiva]
```

- ▶ **[início]**: diretiva executada antes do laço começar (geralmente, atribuir o valor inicial do contador);
- ▶ **[condição]**: expressão de condição de parada do laço (geralmente, comparação com o valor final);
- ▶ **[inc/dec]**: *diretiva executada no final de cada iteração do laço* (geralmente usada para incrementar ou decrementar o contador);
- ▶ **[diretiva]**: diretiva ou bloco de diretivas executadas em cada iteração do laço.



Comando de repetição For

Portugol

```
algoritmo "ExemploPara"  
var  
    degraus: inteiro  
inicio  
    para degraus de 1 ate 10 faca  
        escreval("Suba um degrau")  
    fimpara  
fimalgoritmo
```

Java

```
public class ExemploPara{  
    public static void main(String [] args){  
        int i = 1;  
        for(i = 1; i<= 10; i = i+ 1){  
            System.out.println("Suba um degrau!");  
        }  
    }  
}
```



Os campos do for são opcionais

```
// Conta até 10.  
int i = 1;  
for (; i < 10;) {  
    System.out.println(i++);  
}  
  
// Preenche um vetor.  
int[] v = new int[5];  
for (int i = 0; i < 5; v[i] = i++);  
  
// Loop infinito.  
for (;;) ;
```



Inicialização / incremento múltiplo

- ▶ Podemos efetuar múltiplas diretivas na inicialização e no incremento, se necessário, separando com vírgulas:

```
for (int i = 1, j = i + 10; i < 5; i++, j = i * 2) {  
    System.out.println("i= " + i + " j= " + j);  
}
```

- ▶ Claro que também podemos ter condicionais grandes (usando operadores lógicos):

```
for (int i = 0; (i < 5) && (! achou); i++) {  
    /* ... */  
}
```



Exercícios

- ▶ Escreva um algoritmo que exiba 20 vezes a mensagem “Eu gosto de estudar Algoritmos!”. Utilize o comando de repetição enquanto (while).
- ▶ Escreva um algoritmo que exiba 20 vezes a mensagem “Eu gosto de estudar Algoritmos!”. Utilize o comando de repetição Para (For).
- ▶ Escreva um algoritmo que mostre na tela os números de 1 a 50. Utilize o comando de repetição Enquanto (While).
- ▶ Escreva um algoritmo que mostre na tela os números de 1 a 50. Utilize o comando de repetição Para (For).
- ▶ Escreva um algoritmo que leia 10 números do usuário e calcule a soma desses números. Utilize o comando de repetição Enquanto (While)
- ▶ Escreva um algoritmo que leia 10 números do usuário e calcule a soma desses números. Utilize o comando de repetição Enquanto (For)



Referências

- ▶ SIERRA, Katy; BATES, Bert. **Use a cabeça JAVA**. Ed 2, Editora Altabooks;
- ▶ SIERRA, Katy; BATES, Bert. **SCJP - Certificação Sun para Programador Java**. Editora Editora Altabooks;
- ▶ Material do **ESJUG - Grupo de Usuários Java do Espírito Santo** - Programação Orientada Objetos. Março de 2008.
- ▶ LIGUORI, Robert; LIGUORI, Patricia. **Java Guia de bolso**. Editora Alta Books.
- ▶ Material produzido pela empresa **Argonavis** - Helder da Rocha. Programação Orientada Objetos.

