



# Características das Linguagens de Programação I

**Conteúdo:** Comandos de repetição - for

Prof. Dsc. Giomar Sequeiros  
[giomar@eng.uerj.br](mailto:giomar@eng.uerj.br)



**Comando for**

# Comando for

- O comando "for" é utilizado para **executar** um bloco de **código repetidamente** um número **específico** de vezes.
- Sintaxe básica:

```
for (inicialização; condição; incremento/decremento) {  
    // Código a ser repetido  
}
```

Onde:

- **inicialização**: é a expressão que é **executada** uma **única vez** no início do loop, geralmente para inicializar uma variável de controle.
- **condição**: é a expressão que é **avaliada a cada iteração** do loop. Se a condição for **verdadeira**, o bloco de código dentro do loop é **executado**. Se for **falsa**, o loop é **encerrado**.
- **incremento/decremento**: é a expressão que é executada no **final** de cada **iteração** do loop, geralmente para atualizar a **variável de controle**.

# Comando for: Exemplo 1

- Imprimir os números de 1 a 5:

```
public static void main(String[] args) {  
    for (int i = 1; i <= 5; i++) {  
        System.out.println("valor: " + i);  
    }  
}
```

Diagram illustrating the components of the `for` loop:

- Inicialização** (Initialization): `int i = 1`
- Condição** (Condition): `i <= 5`
- Incremento** (Increment): `i++`

# Comando for: Exemplo 2

---

- Imprimir os números pares de 0 a 10

```
public static void main(String[] args) {  
    for (int i = 0; i <= 10; i += 2) {  
        System.out.println(i);  
    }  
}
```

# Comando for: Exemplo 3

---

- Contagem regressiva de 10 a 1

```
public static void main(String[] args) {  
    for (int i = 10; i >= 1; i--) {  
        System.out.println(i);  
    }  
}
```

# Comando for: outras formas de uso

## Loop infinito

```
for (;;) {  
    // código aqui  
}
```

## Loop com múltiplas variáveis de controle:

```
for (int i = 0, j = 10; i < j; i++, j--) {  
    // código aqui  
}
```

## Loop sem inicialização

```
int i = 1;  
for (; i <= 5; i++) {  
    System.out.println(i);  
}
```

## Loop "for" aninhado

```
for (int i = 0; i < 10; i++) {  
    for (int j = 0; j < 10; j++) {  
        System.out.println("i: " + i + ", j: " + j);  
    }  
}
```

## Loop for sem a parte de incremento

```
for (int i = 1; i <= 5;) {  
    System.out.println(i);  
    i++; // incremento dentro do bloco  
}
```



# Comando for-each

- O for-each é uma **variação** do **loop** for que simplifica a iteração sobre **arrays** e **coleções**. Ele é muito útil para percorrer todos os elementos de uma estrutura de dados sem precisar de um índice.
- Sintaxe:

```
for (TipoElemento elemento : arrayOuColecao) {  
    // Código a ser executado para cada elemento  
}
```

- Onde:
  - TipoElemento: O tipo de dado dos elementos que estão no array ou na coleção.
  - elemento: Nome da variável que irá armazenar cada item durante a iteração.
  - arrayOuColecao: O array ou a coleção sobre o qual o for-each vai iterar.



# Comando for-each: Exemplos

---

## Usando for-each para percorrer um array

```
String[] frutas = {"Maçã", "Banana", "Laranja", "Uva"};

for (String fruta : frutas) {
    System.out.println("Fruta: " + fruta);
}
```

```
int[] numeros = {1, 2, 3, 4, 5};

for (int num : numeros) {
    System.out.println(num);
}
```

# Comando break e continue

# Comando break

- O comando **break** serve para **interromper** a execução de um **loop** ou **sair** de uma estrutura **switch**.
- Quando o **break** é chamado dentro de um loop, o controle sai imediatamente do loop, interrompendo a execução das iterações seguintes.
- Exemplo:

```
public static void main(String[] args) {  
    for (int i = 1; i <= 10; i++) {  
        if (i == 5) {  
            break; // Interrompe o loop quando i for igual a 5  
        }  
        System.out.println("i: " + i);  
    }  
    System.out.println("Loop finalizado.");  
}
```

# Comando continue

- O comando **continue** é usado para pular a iteração atual do loop e avançar diretamente para a próxima. Ele não interrompe o loop, apenas pula o restante do código da iteração atual.
- Exemplo:

```
public static void main(String[] args) {  
    for (int i = 1; i <= 10; i++) {  
        if (i % 2 == 0) {  
            continue; // Pula a iteração se o número for par  
        }  
        System.out.println("i: " + i); // Imprime apenas números ímpares  
    }  
}
```

# Comando break e continue: Exemplo 1

- Programa que valida a entrada de usuário em um programa que solicita um número inteiro.

```
public static void main(String[] args) {  
    for (int i = 1; i <= 10; i++) {  
        if (i == 3) {  
            continue; // Pula a iteração quando i for 3  
        }  
        if (i == 7) {  
            break; // Interrompe o loop quando i for 7  
        }  
        System.out.println("i: " + i);  
    }  
}
```


# Comando break e continue: Exemplo 2

- Programa que determina se um número inteiro positivo é primo.

```
Scanner scanner = new Scanner(System.in);  
System.out.print("Digite um número inteiro: ");  
int numero = scanner.nextInt();
```

```
if (numero <= 1) {  
    System.out.println(numero + " não é primo.");  
} else {  
    boolean primo = true;  
  
    for (int i = 2; i <= Math.sqrt(numero); i++) {  
        if (numero % i == 0) {  
            primo = false;  
            break;  
        }  
    }  
    if (primo)  
        System.out.println(numero + " é primo.");  
    else  
        System.out.println(numero + " não é primo.");  
}
```

// Verifica se o número  
é divisível por  
qualquer número entre 2  
e a raiz quadrada do  
número



```
scanner.close();
```

# Comando break e continue: Exemplo 3

- Programa que valida a entrada de usuário em um programa que solicita um número inteiro entre 1 e 10.

```
Scanner scanner = new Scanner(System.in);
int num;

while (true) { // Loop infinito
    System.out.print("Digite um número inteiro entre 1 e 10: ");
    num = scanner.nextInt();

    if (num < 1 || num > 10) {
        System.out.println("Número inválido. Tente novamente.");
        continue;
    } else {
        System.out.println("O número digitado foi " + num);
        break;
    }
}
scanner.close();
```

## Exemplo de Saída:

```
Digite um número inteiro entre 1 e 10: 21
Número inválido. Tente novamente.
Digite um número inteiro entre 1 e 10: 34
Número inválido. Tente novamente.
Digite um número inteiro entre 1 e 10: 13
Número inválido. Tente novamente.
Digite um número inteiro entre 1 e 10: 12
Número inválido. Tente novamente.
Digite um número inteiro entre 1 e 10: 1
O número digitado foi 1
```



# Exercício 1

- O que será impresso no código a seguir?

```
public static void main(String[] args) {  
    for (int i = 1; i <= 10; i++) {  
        if (i == 3) {  
            continue; // Pula a iteração quando i for 3  
        }  
        if (i == 7) {  
            break; // Interrompe o loop quando i for 7  
        }  
        System.out.println("i: " + i);  
    }  
}
```

## Exercício 2

---

- Escreva um programa que leia um número do teclado e ache todos os seus divisores.
- Escreva um programa que exibe a tabela ascii.

## Exercício 3

- Uma pessoa **investe** uma **quantidade** (em reais) em uma conta de **poupança** que rende uma taxa de juros ao ano. Admitindo que todos os juros são deixados em depósito na conta, calcule e imprima a quantia na conta ao final de cada ano, ao longo de **n** anos. Use a seguinte fórmula para determinar estas quantias:

$$a = p(1 + r)^n$$

- Onde:
  - p** é a quantia investida originalmente (i.e., o valor principal)
  - r** é a taxa anual de **juros**.
  - n** é o número de **anos**
  - a** é a quantia existente em depósito no final do n-ésimo ano.

Exemplo de saída para R\$1000,00 em 10 anos e 5% (0.05) de juros ao ano

Ano	Saldo na conta
1	1050.00
2	1102.50
3	1157.62
4	1215.51
5	1276.28
6	1340.10
7	1407.10
8	1477.46
9	1551.33
10	1628.89

# Referências

## BIBLIOGRAFIA BÁSICA:

- DEITEL, Harvery M.. Java : como programar. 10ª ed. São Paulo: Pearson - Prentice Hall, 2017.
- BORATTI, Isaías Camilo. Programação Orientada a Objetos em Java : Conceitos Fundamentais de Programação Orientada a Objetos. 1ª ed. Florianópolis: VisualBooks, 2007.
- SIERRA, Kathy; BATES, Bert. Use a Cabeça! Java. 2ª ed. Rio de Janeiro: Alta Books, 2007.

