



Características das Linguagens de Programação I

Conteúdo: Comandos de repetição

Prof. Dsc. Giomar Sequeiros
giomar@eng.uerj.br



Comando while

Comando While

- O comando while é uma estrutura de repetição que executa um bloco de código enquanto uma condição for verdadeira.
- A condição é verificada antes da execução do bloco, e, se for falsa desde o início, o bloco de código não é executado.

- Sintaxe:

```
while (condicao) {  
    // Código a ser repetido  
    // enquanto a condição for verdadeira  
}
```

- A condição é uma **expressão** que é **avaliada a cada iteração** do loop
- O bloco de **código** dentro das chaves é executado se a **condição** for **verdadeira**.
- O loop **continuará** a ser executado até que a condição seja **falsa**.

Comando While: Exemplo 1

- Crie uma classe e no método main insira o código mostrado abaixo.

Variável contador

O valor de "i" é incrementado a cada iteração do loop usando o operador "++".

```
int i = 1;

while (i <= 5) {
    System.out.println(i);
    i++;
}
```

O bloco de código é executado enquanto a condição é "i <= 5", for verdadeira.

Saída:

```
1
2
3
4
5
```

O loop continuará a ser executado até que "i" seja maior que 5.

Comando While: Exemplo 2

- Programa que calcula a **soma** dos **números** de **1** a **10** usando um loop "while":

Variável
contador

Variável
acumulador

```
int i = 1;  
int soma = 0;
```

```
while (i <= 10) {  
    soma += i;  
    i++;  
}
```

O loop "while" é executado enquanto a condição "i <= 10" for verdadeira.

```
System.out.println("A soma de 1 a 10 é: " + soma);
```

Saída:

A soma de de 1 a 10 e: 55

Comando While: Exemplo 3

- Programa que cria uma **tabuada**

```
Scanner scanner = new Scanner(System.in);  
int num;  
int i = 1;  
  
System.out.print("Digite um número inteiro: ");  
num = scanner.nextInt();  
  
while (i <= 10) {  
    System.out.println(num + " x " + i + " = " + (num * i));  
    i++;  
}  
  
scanner.close();
```

Saída:

Digite um número inteiro:

8

8 x 1 = 8

8 x 2 = 16

8 x 3 = 24

8 x 4 = 32

8 x 5 = 40

8 x 6 = 48

8 x 7 = 56

8 x 8 = 64

8 x 9 = 72

8 x 10 = 80

Também podemos usar a função printf:

```
System.out.printf("\n %d x %d = %d", num, i, num*i);
```

Comando While: Exemplo 4

Programa solicita ao usuário o número de linhas e colunas e imprime uma matriz

```
Scanner scanner = new Scanner(System.in);
int linhas, colunas;
int i = 1, j = 1;
System.out.print("Digite o número de linhas: ");
linhas = scanner.nextInt();
System.out.print("Digite o número de colunas: ");
colunas = scanner.nextInt();

while (i <= linhas) {
    while (j <= colunas) {
        System.out.printf("%2d ", i * j);
        j++;
    }
    System.out.println();
    i++;
    j = 1;
}
scanner.close();
```

Saída:

```
Digite o numero de linhas: 5
Digite o numero de colunas: 7
1 2 3 4 5 6 7
2 4 6 8 10 12 14
3 6 9 12 15 18 21
4 8 12 16 20 24 28
5 10 15 20 25 30 35
```

// Imprime o produto de i e j
formatado com 2 dígitos

// Quebra de linha após
completar uma linha da matriz

// Reseta o contador j para 1 para
começar uma nova linha

Exercício 1

- Crie um programa que solicite ao usuário um número **inteiro positivo** e, em seguida, imprima os primeiros "n" números da **série de Fibonacci**.
- A série de **Fibonacci** é uma **sequência** de números em que cada **número subsequente** é a **soma dos dois números anteriores**.
 - Os primeiros dois números da série são 0 e 1.
 - Se por **exemplo** o usuário digitar **6**, o programa deve imprimir os primeiros 6 números da série:
0, 1, 1, 2, 3, 5.
 - Se o usuário digitar 10, o programa deve imprimir os primeiros **10** números da série:
- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.

Exercício 1: solução

```
Scanner scanner = new Scanner(System.in);
int n, i = 0, t1 = 0, t2 = 1, proximo;

System.out.print("Digite o número de termos da série que deseja imprimir: ");
n = scanner.nextInt();

System.out.println("Os " + n + " primeiros termos da série são:");

while (i < n) {
    System.out.print(t1 + ", ");
    proximo = t1 + t2;
    t1 = t2;
    t2 = proximo;
    i++;
}

scanner.close();
```

Exercício 1: solução

Exemplo saída 1:

Digite o número de termos da série que deseja imprimir: 5
Os 5 primeiros termos série são:
0, 1, 1, 2, 3,

Exemplo saída 2:

Digite o número de termos da série que deseja imprimir: 10
Os 10 primeiros termos série são:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

Exercício 2

- O jogo da adivinhação é um jogo em que o programa escolhe um **número aleatório** e o jogador deve tentar **adivinhar** esse número.
 - O jogador deve **digitar** um **palpite** e o programa deve dizer se o palpite é **maior** ou **menor** do que o número secreto.
 - O jogo deve continuar até que o jogador adivinhe corretamente o número.

Dicas:

- Estabelecer um intervalo para o número a ser adivinhado (por exemplo, **1** a **100**)
- Permitir que o jogador faça várias tentativas usando leitura por teclado
- Use a função `nextInt(valor)` do objeto `Random`

Exercício 2: solução

```
Random random = new Random();
Scanner scanner = new Scanner(System.in);
```

```
int numSecreto = random.nextInt(100) + 1;
int palpite = 0;
int tentativas = 0;
```

```
System.out.println("Bem-vindo ao Jogo da Adivinhação!");
System.out.println("Tente adivinhar o número entre 1 e 100.");
```

```
while (palpite != numSecreto) {
    System.out.print("Digite seu palpite: ");
    palpite = scanner.nextInt();
    tentativas++;

    if (palpite < numSecreto)
        System.out.println("Tente um número maior!");
    else if (palpite > numSecreto)
        System.out.println("Tente um número menor!");
    else
        System.out.println("Parabéns! Você acertou o número em "+ tentativas+"Tentativas");
}
scanner.close();
```

```
import java.util.Random;
import java.util.Scanner;
```

// Cria um objeto Random para
gerar números aleatórios

// Gera um número
aleatório entre 1 e 100

Exercício 2: solução

Saída:

Bem-vindo ao jogo da adivinhação!
Tente adivinhar o número entre 1 e 100.
Digite um palpite: 8
Tente um número maior!
Digite um palpite: 34
Parabéns! Você acertou o número secreto em 2 tentativas.

Comando do - while

Comando do - while

- O comando do-while em Java é uma estrutura de repetição que executa um bloco de código pelo menos uma vez, e continua executando-o enquanto uma condição for verdadeira.
- A diferença principal entre o do-while e o while é que o do-while garante que o bloco de código seja **executado pelo menos uma vez**, pois a condição é verificada após a execução do bloco.

```
do {  
    // Código a ser executado  
} while (condicao);
```

Onde:

- **do { ... }:** O bloco de código dentro do do é executado primeiro.
- **while (condicao);:** Após a execução do bloco, a condição é verificada. Se a condição for verdadeira, o bloco de código é executado novamente. Se for falsa, a execução do loop é encerrada.

Comando do - while: Exemplo 1

- Imprime os números de 1 a 10

```
public class Exemplo {  
    public static void main(String[] args) {  
        int i = 1;  
  
        do {  
            System.out.println(i);  
            i++;  
        } while (i <= 10);  
    }  
}
```

Saída:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Comando do - while: Exemplo 2

- Programa que solicita ao usuário digitar um número positivo

```
Scanner scanner = new Scanner(System.in);  
int numero;  
  
do {  
    System.out.print("Digite um número positivo: ");  
    numero = scanner.nextInt();  
} while (numero < 0);  
  
System.out.println("O número digitado é " + numero + ".");  
  
scanner.close();
```

Saída:

```
Digite um número positivo: -5  
Digite um número positivo: -2  
Digite um número positivo: 0  
Digite um número positivo: 10  
O número digitado é 10.
```

Comando do - while: Exemplo 3

- Programa que imprime um menu e solicita ao usuário digitar uma opção válida

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

do {
    System.out.println ("Escolha uma opção:");
    System.out.println ("1 - Abrir arquivo");
    System.out.println ("2 - Salvar arquivo");
    System.out.println ("3 - Sair");
    opcao = scanner.nextInt ();
} while (opcao < 1 || opcao > 3);

switch (opcao) {
    case 1:
        System.out.println ("Abrindo arquivo...");
        break;
    case 2:
        System.out.println ("Salvando arquivo...");
        break;
    case 3:
        System.out.println ("Encerrando programa...");
        break;
}
```

Prof. scanner.close ();

Saída:

```
Escolha uma opção:
1 - Abrir arquivo
2 - Salvar arquivo
3 - Sair
4
Escolha uma opção:
1 - Abrir arquivo
2 - Salvar arquivo
3 - Sair
56
Escolha uma opção:
1 - Abrir arquivo
2 - Salvar arquivo
3 - Sair
3
Encerrando programa...
```

Exercício

- Escreva um programa em Java usando o comando **do-while** que solicita ao usuário que insira uma **senha** de **4 dígitos**. Se a senha estiver correta, o programa exibe uma mensagem de boas-vindas e sai do loop. Se a **senha** estiver **incorreta**, o programa exibe uma **mensagem** de **erro** e solicita que o usuário **tente novamente**. O programa deve permitir que o usuário tente novamente **até** um máximo de **3 tentativas**.

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.

