

Programação I

Aula 3

Professor: Diogo Passos Ranghetti

Sumário

Estruturas de Repetições.....	3
Declaração <i>while</i>	4
Declaração <i>do-while</i>	5
Declaração <i>for</i>	6
Referências bibliográficas.....	7

Estruturas de Repetições

Dentre os diversos comandos disponíveis em uma linguagem de programação é preciso conhecer e saber manipular bem aqueles que alteram o fluxo de execução dos programas, sendo as estruturas de repetição o principal recurso para o programador reaproveitar diversas linhas do seu código fonte.

Estruturas de controle e repetições são comandos em linguagem de programação que permitem executar partes específicas de um código determinada quantidade de vezes, ou seja, caso seja necessário realizar uma equação mais de uma vez basta colocar a lógica que realiza a equação dentro de uma estrutura de controle e repetição.

Estruturas de controle e repetições são bastante usadas para a realização de lógicas onde seja necessário ler varias vezes os valores de um determinado processo, uma situação para exemplificar seria imprimir em tela os valores de uma determinada tabela quando os valores dessa tabela atenda uma determinada condição.

O Java dispõe de três tipos de estruturas de controle e repetição, sendo eles o **while** o **do-while** e o **for**, cada uma dessas estruturas, além de ser uma palavra reservada dentro da linguagem, possui características distintas e isso determina o momento ideal para serem aplicadas, haverá situações em que uma estrutura será mais bem aplicada a determinada situação do que outra.

Declaração *while*

A declaração ***while*** executa um determinado bloco do código enquanto a condição lógica aplicada a ele for verdadeira.

Por exemplo:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    System.out.println("===== Meu Programa Java =====");  
    Scanner s = new Scanner(System.in);  
  
    System.out.println("\nDigite sua Idade: ");  
    int idade = s.nextInt();  
  
    System.out.println("-----");  
  
    while(idade > 0){  
        System.out.println(idade + " - anos já processados!");  
        idade--;  
    }  
}
```

A declaração ***while*** valida se a idade é maior do que zero e enquanto essa condição for verdadeira ira imprimir o valor da idade, isso ocorre porque o laço de repetição da instrução ***while*** inicia apenas se a condição for verdadeira.

Nesse caso o resultado seria o seguinte:

```
-----  
10 - anos já processados!  
9 - anos já processados!  
8 - anos já processados!  
7 - anos já processados!  
6 - anos já processados!  
5 - anos já processados!  
4 - anos já processados!  
3 - anos já processados!  
2 - anos já processados!  
1 - anos já processados!
```

Declaração *do-while*

A declaração ***do-while*** é parecida com a declaração ***while***, a diferença entre elas é que a declaração ***do-while*** ira ser processada ao menos uma vez já que ao invés de validar uma condição para entrar no laço como a declaração ***while*** faz o ***do-while*** entra no bloco de lógica e se a condição aplicada a ele for verdadeira ele permanece no laço de repetição.

Por exemplo:

```
public static void main(String[] args) {
    // TODO code application logic here
    System.out.println("===== Meu Programa Java =====");
    Scanner s = new Scanner(System.in);

    System.out.println("\nDigite um numero: ");
    int numero = s.nextInt();
    int contador;

    System.out.println("-----");

    contador = numero;
    do{
        System.out.println(contador + " - processo do-while!");
        contador--;
    }while(contador > 10);

    contador = numero;
    while(contador > 10){
        System.out.println(numero + " - processo while!");
        contador--;
    }
}
```

Para o exemplo acima o resultado é diferente dependendo do numero digitado.

Se o numero digitado for 10:

```
-----
10 - processo do-while!
```

Se o numero digitado for 11:

```
-----
11 - processo do-while!
11 - processo while!
```

Declaração *for*

A declaração ***for*** assim como as demais também executa um bloco de código uma determinada quantidade de vezes, sua principal diferença é que na declaração ***for*** é informado, a declaração inicial, a expressão lógica e o salto ou contador, e enquanto a expressão lógica for verdadeira ira repetir o bloco de código contido no laço.

Por exemplo:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    System.out.println("===== Meu Programa Java =====");  
    Scanner s = new Scanner(System.in);  
  
    System.out.println("\nDigite sua Idade: ");  
    int idade = s.nextInt();  
  
    System.out.println("-----");  
  
    for (int i = 0; i < 10; i++) {  
        System.out.println(i + " - Sua idade é " + idade);  
    }  
}
```

Para o exemplo acima ira imprimir 10 vezes, note que “int i = 0” é a declaração inicial seguido pela condição lógica “i < 10” e por fim o contador “i++” todos separados por um “;” (ponto e virgula).

Resultado:

```
===== Meu Programa Java =====  
Digite sua Idade:  
31  
-----  
0 - Sua idade é 31  
1 - Sua idade é 31  
2 - Sua idade é 31  
3 - Sua idade é 31  
4 - Sua idade é 31  
5 - Sua idade é 31  
6 - Sua idade é 31  
7 - Sua idade é 31  
8 - Sua idade é 31  
9 - Sua idade é 31  
CONSTRUÍDO COM SUCESSO (tempo total: 4 segundos)
```

Referências bibliográficas

DEITEL, H. M.; DEITEL, P. J. Java como Programar. 8ª.ed. São Paulo: Pearson Brasil, 2010.