

Laço de Repetição DO-WHILE

Prof. Dr. Marcelo Fernando Rauber

O laço de repetição Do-While é muito parecido com o laço de repetição while que já vimos.

Uma estrutura de repetição Do-While é utilizada quando um trecho do algoritmo (ou até mesmo o algoritmo inteiro) precisa ser repetido. O número de repetições pode ser fixo ou estar atrelada a uma condição.

Também é um laço de repetição de fim genérico, que tem como principal característica a realização de um teste lógico no **fim** do laço. Isso significa que ao menos uma vez o código dentro do laço será executado, independentemente do resultado do teste lógico no final. Se o teste lógico no final resultar em verdadeiro, serão executados uma nova vez e sequencialmente os comandos dentro de seu marcador de bloco. Ao chegar ao marcador de fim de bloco, o teste lógico é executado e caso resulte em verdadeiro o fluxo de execução do algoritmo retorna a linha onde está o “do”, repetindo a execução do código.

Quando o teste lógico do while resultar em falso, é interrompida a repetição do laço, sendo executada a próxima instrução.

Sintaxe do DO-While:

```
do {  
    <comando1>  
    <comando2>  
    <comando3>  
    <comando....>  
} while ( <teste Lógico> ) ;
```

ATENÇÃO: com do-while observe que alinha onde está o while termina com ;

Onde:

<teste Lógico> Indica o local do teste lógico. São testes como os que utilizamos com o if.

<comando...> Indica o local onde colocaremos os comandos, isto é, as instruções. Podem ser quaisquer uma das que vimos até o momento, inclusive outro laço de repetição.

Recapitulando então:

1) a principal diferença entre o while e do-while é que:

- a) o código será executado ao menos uma vez (independentemente do resultado do teste lógico);
- b) o teste lógico é executado no final

2) O fato do código ser executado uma vez é que determina qual desses laços escolher, de acordo com as características do problema a ser resolvido.

Exemplo de Código:

Um programa em JAVA que apresenta na tela a lista de números de zero a cem com do-while.

```
package javaapplication118;

/**
 *
 * @author Prof_Marcelo_Rauber
 */
public class JavaApplication118 {

    public static void main(String[] args) {
        // TODO code application logic here
        int contador = 0;
        do {
            System.out.println(contador);
            contador = contador + 1;
        } while (contador <= 100);
    }
}
```