



Programação I

</ Estruturas de Repetição />

- **Estruturas de Repetição:**

- While
- Do... while
- For



2021

TekMind

Estruturas de Repetição

É comum a repetição de procedimentos para se realizar tarefas. Facilita quando queremos escrever a mesma instrução várias vezes. Não são repetidos eternamente, mas se encerram quando o objetivo é atingido.

Existem três estruturas (laços) de repetição, são elas: **While**, **Do...While** e **For**.

While

- Esse comando testa a condição no início e só repete se for **verdadeira**.
- Quando o número de repetições for desconhecido, é necessária uma **chave** para interromper a repetição.
- Os algoritmos podem repetir uma ou mais instruções, ser executado várias vezes e ter caminhos alternativos.

Sintaxe:

```
While (expressão lógica) {  
    bloco de comandos  
}
```

Cada execução do bloco de instruções é chamada **interação**. Para termos a informação de quantas interações já foram realizadas, precisamos de uma variável com a função de **contador**.

→ Contador

Utilizado para contar a **quantidade** de vezes que uma **instrução** é executada.

Sintaxe:

```
VARIÁVEL = VARIÁVEL + CONSTANTE
```

Observações:

1) A variável contador normalmente é inicializada com zero, ou seja, zera-se a variável antes de utilizá-la:

```
VARIÁVEL = 0
```

2) A constante (valor 1) determina o valor do incremento da variável (contador), ou seja, o que será somado a ela.

→ Acumuladores (somadores)

É utilizado para obter somatórios (Σ).

Sintaxe:

```
VARIÁVEL1 = VARIÁVEL1 + VARIÁVEL2
```

Observações:

1) A variável do acumulador deve sempre ser inicializada, normalmente com zero, igual ao contador.

2) A variável2 indica o valor a ser acumulado, somado e armazenada na variável1.

Obs.: A utilização de comandos de **repetição encadeados** geralmente causa uma certa dificuldade para seu acompanhamento à primeira vista. Podemos ter tantos laços quisermos, e ainda tê-los combinados com comandos de decisão e assim por diante.

Do... while

Esse comando funciona de forma similar ao comando while, exceto pelo fato de que a condição de controle só é testada **após** a execução do bloco de comandos.

Sintaxe:

```
do{  
    bloco de instruções  
}  
While(valor booleano);  
bloco de comandos
```

Assim, podemos utilizá-lo quando termos certeza de que o bloco de instruções será executado ao menos uma vez, sem a necessidade do teste na entrada do bloco.

For

A lista de comandos é executada somente se o valor da variável for **menor ou igual** ao valor final, depois ela é incrementada a variável de controle. Isso se repete até que o valor da variável de controle seja maior que o valor final.

Sintaxe:

```
for(valor inicial; condição; valor do incremento){  
    bloco de comandos  
}
```

Portanto, temos a separação do comando em três cláusulas: **preparação, condição e passo.**

Resumindo...

While (expressão lógica){ bloco de comandos }	O while analisa a expressão lógica , se resultado for verdadeiro, o bloco de comandos é executado.
do { bloco de instruções } While (valor booleano); bloco de comandos	O do executa o bloco de instruções até que o valor retornado pelo bloco de comandos seja Verdadeiro.
for (valor inicial; condição; valor do incremento){ bloco de comandos }	O for incrementa, a variável, a partir do valor inicial de uma unidade, até que atinja o valor final. E para cada incremento o bloco de comandos é executada.