

Capítulo 8 - Loops pré-definidos

No último capítulo você aprendeu a fazer LOOPS. Você descobriu que é possível fazer loops no seu algoritmo através de duas estruturas de repetição ENQUANTO-FAÇA e REPITA-ATÉ e aprendeu a diferença entre estas duas estruturas.

Neste capítulo nós vamos ver a estrutura de LOOP mais utilizada na programação: A estrutura de repetição PARA-FAÇA.

Entender bem esta estrutura determinará se você será um bom ou um mau programador, portanto preste bastante atenção neste capítulo. Releia quantas vezes forem necessárias. No final tem a solução do exercício do último capítulo e um novo exercício para você resolver. Vamos lá?

O que é um LOOP Pré-definido?

Quando fazemos um algoritmo, muitas vezes já sabemos a quantidade de vezes que um loop deve executar. Por exemplo, some todos os números de 1 a 100. Neste caso, sabemos que o nosso loop deverá ser executado 100 vezes.

O caso mais usado deste tipo de LOOP na programação é quando você deve acessar todos os itens de um vetor, matriz ou lista. (Veremos o que são vetores e matrizes no próximo capítulo)

Por exemplo, uma situação muito comum para programadores. Imagine que você deve enviar um e-mail para todos os clientes cadastrados no seu banco de dados...

Você sabe que tem uma tabela com 3298 clientes no seu banco de dados. Neste caso, você deve fazer um loop de 1 até 3298, e enviar um e-mail para cada cliente.

Entendido o que é um LOOP pré-definido, vejamos qual estrutura de repetição utilizada para este caso.

A estrutura PARA-FAÇA

Você deve estar imaginando que é possível implementar loop pré-definido utilizando as estruturas de repetição que você aprendeu no capítulo passado.

Sim, é perfeitamente possível! Para isto você precisaria utilizar uma variável que chamamos de "contador".

Esta variável nada mais é do que uma simples variável do tipo inteiro que é responsável por contar quantas iterações (execuções do loop) foram executadas.

Vamos tomar como exemplo o caso que disse anteriormente. Como somar todos os números de 1 a 100.

Um algoritmo com a estrutura ENQUANTO-FAÇA para este problema ficaria assim:

```
Algoritmo "Somar1a100ComEnquanto"
Var
    contador : INTEIRO
    soma : INTEIRO
Inicio
    contador := 1
    soma := 0
    ENQUANTO contador <= 100 FAÇA
        soma := soma + contador
        contador := contador + 1
    FIMENQUANTO
    ESCREVA("A soma de 1 a 100 é: ", soma)
Fimalgoritmo
```

Embora seja possível utilizar estas estruturas de repetição para implementar um loop pré-definido, há uma estrutura criada especificamente para isto. A estrutura de repetição PARA-FAÇA.

O que o PARA-FAÇA faz é justamente implementar um contador implicitamente. Ou seja, as operações de inicializar o contador (contador := 1), incrementar o contador (contador := contador + 1) e verificar se o LOOP deve continuar (contador <= 100) é realizada implicitamente pela estrutura PARA-FAÇA.

O esquema de utilização desta estrutura é assim:

PARA <contador> **DE** <valor inicial> **ATE** <valor final> [**PASSO** <valor de incremento>] **FAÇA**

<instruções a serem executadas repetidamente até a <contador> atingir o valor final>

FIM-PARA

A inicialização do contador é realizado implicitamente com o informado na declaração da estrutura. A condição para executar a iteração é que o valor da variável contadora não tenha atingido o <valor final>. E ao final de cada iteração, o valor da variável contadora é incrementado em 1 (ou o valor declarado como PASSO).

Repare que o passo de incremento é opcional, por padrão o contador é incrementado de 1 em 1, mas você pode especificar que quer um outro valor de incremento, por exemplo de 2 em 2 ou de 3 em 3.

Se for usar o incremento padrão de 1 em 1, você pode ignorar o PASSO.

PARA <contador> **DE** <valor inicial> **ATE** <valor final> **FAÇA**

<instruções a serem executadas repetidamente até a <contador> atingir o valor final>

FIM-PARA

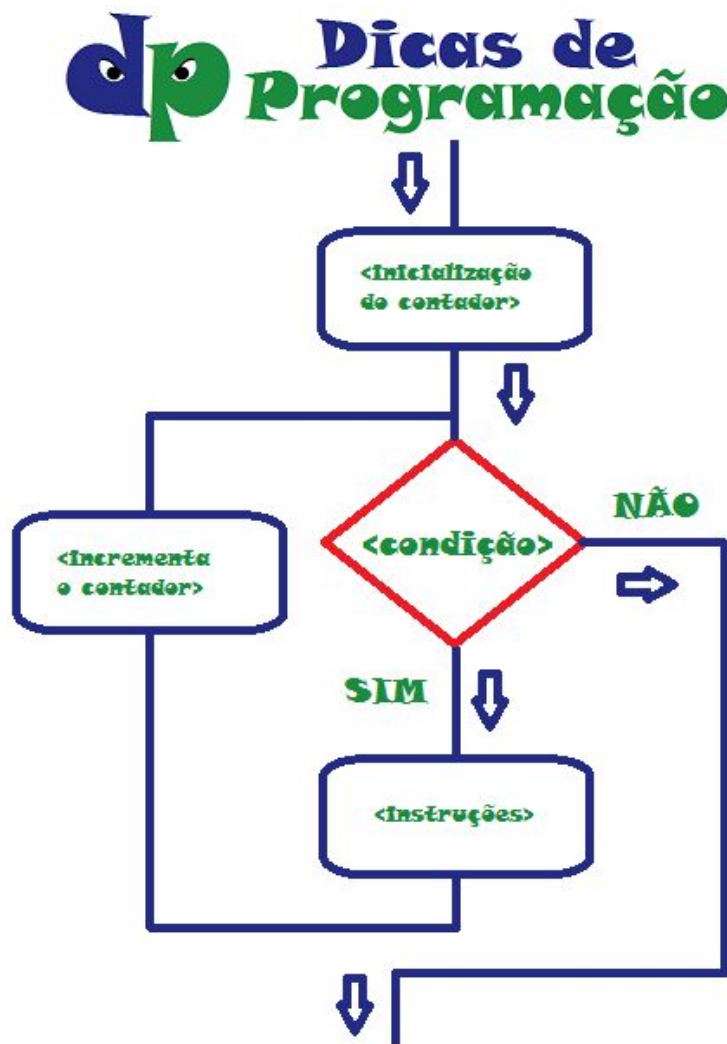
Para o nosso problema de somar todos os números de 1 a 100, um algoritmo com a a estrutura PARA-FAÇA ficaria assim:

```
Algoritmo "Somar1A100ComPara"
Var
    contador : INTEIRO
    soma : INTEIRO
Inicio
    soma := 0
    PARA contador DE 1 ATÉ 100 FAÇA
```

```
soma := soma + contador  
FIMPARA  
    ESCREVA("A soma de 1 a 100 é: ", soma)  
Fimalgoritmo
```

Viu a diferença? No fundo é a mesma coisa, mas para loops pré-definidos a estrutura mais utilizada é a PARA-FAÇA.

Vejamos um fluxograma desta estrutura de repetição:



Hora de praticar!

Para dar mais um exemplo de LOOP pré-definido. Vamos fazer um algoritmo para resolver um problema matemático: O fatorial de um número.

Se você não sabe, fatorial é a multiplicação de todos os números de 1 até o número que se está calculando. Por exemplo: Fatorial de 5 = $1 * 2 * 3 * 4 * 5 = 120$. Fácil né?

Na matemática a notação de fatorial o número e uma exclamação. Por exemplo 5! significa fatorial de 5.

Primeiro vamos fazer um algoritmo utilizando o ENQUANTO.

```
algoritmo "FatorialComENQUANTO"

var
    numero : INTEIRO
    fatorial : INTEIRO
    contador : INTEIRO
inicio

    ESCREVA ("Digite o número para calcular o fatorial: ")
    LEIA (numero)

    fatorial := 1
    contador := 1
    ENQUANTO contador <= numero FAÇA
        fatorial := fatorial * contador
        contador := contador + 1
    FIMENQUANTO

    ESCREVA ("O fatorial de ", numero, " é : ", fatorial)

fimalgoritmo
```

Veja que foi necessário incrementar o contador explicitamente, ou seja, iniciar a variável contador com 1 e incrementar o seu valor no final de cada iteração do LOOP. Com a estrutura de repetição PARA-FAÇA, isso não é necessário. Vejamos agora o mesmo algoritmo implementado com o PARA-FAÇA:

```
algoritmo "FatorialComPARA"

var
    numero : INTEIRO
    fatorial : INTEIRO
    contador : INTEIRO
inicio

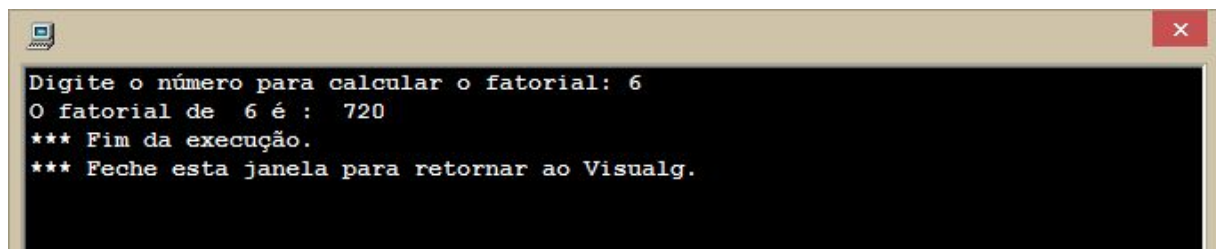
    ESCREVA ("Digite o número para calcular o fatorial: ")
    LEIA (numero)

    fatorial := 1
    PARA contador DE 1 ATE numero FACA
        fatorial := fatorial * contador
    FIMPARA

    ESCREVA ("O fatorial de ", numero, " é : ", fatorial)

finalgoritmo
```

Nesta estrutura, não é necessário incrementar nem inicializar o contador, isso é feito automaticamente. O resultado dos dois algoritmos é o mesmo, veja um exemplo de execução deste algoritmo.



LOOPS podem ser implementados com qualquer estrutura de repetição, porém, em alguns casos uma estrutura se mostra mais adequada que outras, como nesse caso do fatorial a mais adequada é a estrutura PARA. Conhecer essas estruturas de repetição é muito importante para criar programas melhores.

Como eu disse, a estrutura de repetição PARA-FAÇA é muito utilizada para acessar os valores de vetores, matrizes e listas.

Novo exercício para você resolver com a estrutura PARA-FAÇA!

Como sempre digo, lógica de programação só se aprende praticando. Então é a sua vez de tentar resolver um novo problema utilizando algoritmos. O exercício deste capítulo é o seguinte:

Faça um algoritmo para informar se um determinado número é primo ou não.

Número primo é todo número que só é divisível por 1 e por ele mesmo sem deixar resto. Exemplos de números primos são: 2, 3, 5, 7, 11, 13, 17 ...

Fácil né!? Dica, você precisará criar um LOOP (de preferência utilizando o PARA) e verificar se o resto das divisões é 0 utilizando o operador **mod**. Por exemplo, a expressão "6 MOD 4" resulta 2, pois é o resto da divisão de 6 por 4.

No final do próximo capítulo você poderá conferir o meu algoritmo para este problema. Assim, você poderá comparar o seu algoritmo com o meu. Mas tente resolver antes heim!