

Laços de Repetição

1 – Utilização de Laços de Repetição

Os laços de repetição são muito úteis e muito utilizados, pois possibilitam a facilidade de repetir um certo trecho de programa por um determinado número de vezes. O número de vezes que um laço pode repetir um trecho de programa pode ser previamente conhecido ou não, vai depender da característica de execução do laço.

Um laço de repetição em que se conhece o número de vezes que ele será executado é predeterminado, pois sabe-se quando começa e quando termina. Porém, há ainda a possibilidade de desenvolver um laço de repetição indeterminado, ou seja, sabe-se quando ele começa, mas não se sabe quando ele termina, pois apenas o usuário do programa é que determina o final.

Para os laços predeterminados, de forma geral podem ser usados os tipos de laços de repetição: **enquanto**, **repita** e **para**. Para os laços indeterminados, podem ser usados apenas os laços de repetição dos tipos **enquanto** e **repita**.

Os laços de repetição dos tipos **enquanto** e **repita** são controlados pelo uso de condição, logo são laços de repetição condicionais. Já o laço de repetição do tipo **para** é incondicional, pois não faz uso do conceito de condições.

De forma geral, quando há necessidade de trabalhar com um laço de repetição predeterminado, utiliza-se o conceito **para**, deixando os conceitos de laços de repetição **enquanto** e **repita** para situações de laços indeterminados. Apesar desta característica operacional, nesta obra serão utilizados todos os tipos de laço para demonstrar suas aplicações nas mais variadas situações.

2 – Os Laços de Repetição

Como exposto anteriormente, os laços de repetição são três, cada qual para uma finalidade específica. Mas, apesar desta característica específica, um laço de repetição pode, em algumas ocasiões, substituir outro laço de repetição, além de executar sua ação específica, desde que um consiga atender à necessidade do outro. Caso contrário, o laço de repetição não pode ser usado em substituição a outro laço.

Dos três laços existentes o laço **repita** é o mais versátil, pois executa sua ação específica e consegue, com alguns cuidados, executar também as ações dos laços de repetição **enquanto** e **para**. Já o laço **enquanto** consegue, além de sua ação específica, executar com alguns cuidados a ação do laço de repetição **para**. Por sua vez, o laço de repetição **para** executa apenas sua ação específica. A tabela seguinte demonstra o nível de ação de cada estrutura de repetição.

Laço de repetição	Repita	Enquanto	Para
Repita	Executa	Executa	Executa
Enquanto	-	Executa	EXECUTA
Para	-	-	Executa

Outro detalhe a ser considerado é o fato de ocorrer a necessidade de encadear entre si as estruturas de repetição, segundo alguma necessidade específica.

2.1 – Laço Enquanto

O laço de repetição do tipo **enquanto** caracteriza-se por efetuar um teste condicional no início do *laço*. Desta forma, é possível verificar se é permitido executar o trecho de instruções subordinado à condição. Para este tipo de estrutura será utilizado o código em português estruturado **enquanto...faça** e **fim_enquanto**.

Português estruturado

Enquanto (<condição>) faça Instruções executadas enquanto condição for verdadeira Fim_enquanto

2.2 – Laço Repita

O laço de repetição do tipo **repita** caracteriza-se por efetuar um teste condicional no final do *laço*. Desta forma, as instruções subordinadas ao laço serão executadas no mínimo uma vez. Para este tipo de estrutura será utilizado o código em português estruturado **repita** e **até_que**.

Português Estruturado

Repita Instruções executadas até que condição torne-se verdadeira Até_que (<condição>)
--

2.3 – Laço Para

O laço de repetição do tipo **para** caracteriza-se por efetuar um laço por um determinado número de vezes previamente definido. Para este tipo de estrutura será utilizado o código em português estruturado **para...de...até...passo...faça** e **fim_para**.

Português Estruturado

Para variável de início até final passo incremento faça Instruções executadas até contador atingir seu final Fim_para

3- Exercícios de Aprendizagem (Laços pré-determinados)

Com o objetivo de mostrar um exemplo prático de uso do conceito de laços de repetição predeterminados, são apresentados alguns exemplos de aprendizagem, criados com os conceitos de laços de repetição do tipo enquanto, repita e para.

Para os exemplos em questão considere de forma genérica o seguinte problema computacional:

Elaborar um programa de computador que efetue e apresente o resultado do cálculo da fatorial do número 5.

Neste sentido, vale lembrar que fatorial é o produto sucessivo dos números naturais iniciados em 1 até o inteiro n, no caso apresentado 5. Desta forma, o cálculo de uma fatorial é conseguido pela multiplicação sucessiva do número de termos.

No caso do cálculo de uma fatorial de número 5 (5!), seria então necessário efetuar a multiplicação sucessiva de: $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$ ou $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$, que resultaria no valor 120.

O programa deve executar as multiplicações sucessivamente e acumulá-las a fim de possuir o valor 120 após cinco passos.

3.1 – Primeiro Exemplo

O número de passos deve ser controlado por um contador por intermédio do laço de repetição **enquanto**.

Algoritmo

1. Inicializar as variáveis FATORIAL e CONTADOR com 1;
2. Multiplicar sucessivamente a variável FATORIAL pela variável CONTADOR;
3. Incrementar 1 à variável CONTADOR, efetuando o controle até 5;
4. Apresentar ao final o valor obtido.

Pelo fato de ter que efetuar o cálculo de uma fatorial de 5 (5!), isto implica que o contador deve variar de 1 a 5, e por este motivo deve ser a variável CONTADOR inicializada com valor 1. Pelo fato de a variável FATORIAL possuir ao final o resultado do cálculo da fatorial pretendida, ela deve ser inicializada com valor 1. Se ela for inicializada com zero, não existirá resultado final, pois qualquer valor multiplicado por zero resulta zero.

Multiplicar sucessivamente implica em efetuar a multiplicação da variável CONTADOR pelo valor atual da variável FATORIAL a cada passo controlado pelo *laço*. No caso, por cinco vezes.

Veja em seguida a resolução do problema do cálculo de fatorial, utilizando a estrutura de repetição do tipo enquanto. São apresentados também o diagrama de quadros e o código em português estruturado.

Note dentro do laço de repetição a indicação de dois contadores. O primeiro funciona como um acumulador, pois ele terá no final o valor do resultado da fatorial, e o segundo é utilizado para controlar a execução do laço de repetição e ser a base para o cálculo do acumulador.

Logo no início do diagrama de quadros, as variáveis CONTADOR e FATORIAL são igualadas em 1. Na primeira passagem dentro do laço, a variável FATORIAL é implicada pelo seu valor atual, no caso 1, multiplicado pelo valor da variável CONTADOR também com valor 1 que resulta 1. Em seguida a variável CONTADOR é incrementada por mais 1, tendo agora o valor 2. Como 2 é menor ou igual a 5, ocorre um novo cálculo. Desta vez a variável FATORIAL que possui o valor 1 é multiplicada pela variável CONTADOR que possui o valor 2, resultando 2 para FATORIAL. Daí a variável CONTADOR é incrementada de mais 1, tendo agora o valor 3. Desta forma, serão executados os outros cálculos até que a condição se torne falsa e seja então apresentado o valor 120. Veja em seguida a tabela com os valores das variáveis antes e durante a execução do laço:

Contador	Fatorial	Fatorial ← Fatorial*Contador	Comentários
1	1	1	Valor inicial das variáveis e da fatorial
2	1	2	Cálculo da fatorial com o contador em 2
3	2	6	Cálculo da fatorial com o contador em 3
4	6	24	Cálculo da fatorial com o contador em 4
5	24	120	Cálculo da fatorial com o contador em 5

Perceba que quando a variável CONTADOR está com valor 5, a variável FATORIAL está com o valor 120. Neste ponto, o laço é encerrado e é apresentado o valor da variável FATORIAL.

Português Estruturado

Programa FATORIAL1

Var

CONTADOR : inteiro

FATORIAL : inteiro

Início

FATORIAL ← 1

CONTADOR ←

```

Enquanto (CONTADOR <= 5) Faça
    FATORIAL ← FATORIAL * CONTADOR
    CONTADOR ← CONTADOR + 1
Fim_enquanto
Escreva "Fatorial de 5 é = ", FATORIAL
Fim

```

No código anterior, estão sendo apresentadas pela instrução escreva duas informações, sendo uma mensagem e uma variável. Note que isto é feito utilizando uma vírgula para separar as duas informações.

3.2– Segundo Exemplo

Com base no mesmo problema do cálculo da fatorial de 5, é apresentada em seguida a solução utilizando o laço de repetição do tipo **repita**.

Português Estruturado

```

Programa FATORIAL 2
Var
    CONTADOR : inteiro
    FATORIAL  : inteiro
Início
    FATORIAL ← 1
    CONTADOR ← 1
    Repita
        FATORIAL ← FATORIAL * CONTADOR
        CONTADOR ← CONTADOR + 1
    Até_que (CONTADOR > 5)
    Escreva "Fatorial de 5 é = ", FATORIAL
Fim

```

3.3 – Terceiro Exemplo

Com base no mesmo problema do cálculo da fatorial de 5, é apresentado em seguida a solução utilizando o laço de repetição do tipo **para**.

Português Estruturado

```

Programa FATORIAL 3
Var
    CONTADOR ; inteiro
    FATORIAL : inteiro
Início

```

<i>FATORIAL</i> ← 1 Para <i>CONTADOR</i> de 1 até 5 passo 1 faça <i>FATORIAL</i> ← <i>FATORIAL</i> * <i>CONTADOR</i> Fim_para Escreva “FATORIAL de 5 é = ‘ , <i>FATORIAL</i> Fim

4 – Exercícios de Aprendizagem (Laços Indeterminados)

Com o objetivo de mostrar um exemplo prático de uso do conceito de laços de repetição indeterminados, veja em seguida dois exemplos de aprendizagem, criados com os conceitos de laços de repetição do tipo enquanto e repita. Nesta situação em específico não é possível a solução com a utilização do laço de repetição do tipo para.

No tópico anterior, foi apresentado um exemplo para a resolução do problema do cálculo da fatorial de 5. Neste sentido, seria melhor possuir a solução aberta para que um programa calcule a fatorial de um número qualquer. Seria também interessante capacitar o programa a calcular outras fatoriais até que o usuário não mais deseje utilizar o programa. Sendo assim, o programa deve pedir ao usuário a sua continuidade ou não.

4.1 – Primeiro Exemplo

O primeiro exemplo apresenta um programa que efetua o cálculo da fatorial de um número qualquer. O programa em questão solicita ao usuário autorização de continuidade de execução e controla esta opção com o laço de repetição do tipo enquanto.

Algoritmo

- | |
|--|
| <ol style="list-style-type: none">1. Inicializar as variáveis <i>FATORIAL</i> e <i>CONTADOR</i> com 1;2. Definir as variáveis <i>RESP</i> (resposta) para confirmação e <i>N</i> para receber o limite de valor para o cálculo da fatorial;3. Enquanto <i>RESP</i> do usuário for sim, executar os passos 3, 4, 5, 6 e 7;4. Repetir a execução dos passos 4 e 5 por <i>N</i> vezes;5. <i>FATORIAL</i> ← <i>FATORIAL</i> * <i>CONTADOR</i>;6. Incrementar 1 à variável <i>CONTADOR</i>;7. Solicitar autorização de continuidade do programa;8. Apresentar ao final o valor obtido. |
|--|

Português estruturado

Programa <i>FATORIAL</i> 4

```

Var
    CONTADOR, FATORIAL, N : inteiro
    RESP : caractere
Início
    RESP ← “SIM”
    Enquanto (RESP = “SIM”) faça
        FATORIAL ← 1
        Escreva “Fatorial de que número: “
        Leia N
        Para CONTADOR de 1 até N passo 1 faça
            FATORIAL ← FATORIAL * CONTADOR
        Fim_para
        Escreva “Fatorial de “ , N , “é = “ , FATORIAL
        Escreva “Deseja continuar?”
        Leia RESP
    Fim_enquanto
Fim

```

4.2 – Segundo Exemplo

O segundo exemplo apresenta um programa que efetua o cálculo da fatorial de um número qualquer. O programa em questão solicita ao usuário autorização de continuidade de execução e controla esta opção com o laço de repetição do tipo repita.

Algoritmo

1. Inicializar as variáveis *FATORIAL* e *CONTADOR* com 1;
2. Definir as variáveis *RESP* (resposta) para confirmação;
3. Solicitar o valor de *N* para receber o limite de valor para o cálculo da fatorial;
4. Repetir a execução dos passos 4 e 5 por *N* vezes;
5. *FATORIAL* ← *FATORIAL* * *CONTADOR*;
6. Incrementar 1 à variável *CONTADOR*;
- 7.
8. Solicitar autorização de continuidade do programa;
9. Apresentar ao final o valor obtido.

Português estruturado

```

programa FATORIAL 5
var

```

CONTADOR, FATORIAL, N : inteiro

RESP : caractere

Início

RESP ← “SIM”

Repita

FATORIAL ← 1

Escreva “Fatorial de que número: “

Leia *N*

Para *CONTADOR* de 1 até *N* **passo 1 faça**

FATORIAL ← *FATORIAL* * *CONTADOR*

Fim-para

Escreva “Fatorial de “, *N* , “é = “, *FATORIAL*

Escreva “ Deseja continuar?”

Leia *RESP*

Até_que (*RESP* < > “ SIM ”)

Fim