



Programação de Computadores I

Estruturas de Repetição

PROFESSORA CINTIA CAETANO

Introdução

- ▶ Existem situações onde é necessário repetir um determinado trecho de um programa um certo número de vezes.
- ▶ Assim, existem as estruturas de repetição, que permitem que você execute estas tarefas de forma mais simplificada.
- ▶ Podemos chamar as estruturas de repetição de laços ou loops, sendo que podem existir dois tipos de laços de repetição:
 - ▶ **Laços finitos:** neste tipo de laço se conhece previamente o número de repetições que serão executadas; e
 - ▶ **Laços infinitos:** neste tipo de laço não se conhece previamente o número de repetições que serão executadas. São também chamados de condicionais, pelo fato de encerrarem sua execução mediante uma determinada condição.

Para... Faça...

- ▶ O comando PARA... FAÇA... executa repetitivamente um comando enquanto é atribuído uma série de valores a uma variável de controle (contador do PARA).
- ▶ **Sintaxe:**

```
para <variável de controle> de <valor inicial> ate <valor final> [passo  
<incremento>] faca  
    <lista de comandos>  
fimpara
```

Para... Faça...

- ▶ Na estrutura para..faça, a variável de controle é inicializada com <valor inicial> e no início de cada iteração, seu valor é comparado com <valor final>. Se o valor da variável for menor ou igual a <valor final>, a lista de comandos é executada e após ser executado o último comando da lista, a variável de controle é incrementada. Isto repete-se até que o valor da variável de controle seja maior que <valor final>, quando então é executado o comando imediatamente após a palavra fimpara. A instrução passo é necessária se o incremento for diferente de 1.

Exemplo

- ▶ Ler os números de 1 a 100 e somá-los armazenando em uma variável. Ao final imprimir a soma.

```
algoritmo "Usando_Para_Faca"  
var  
soma, i: inteiro  
inicio  
soma ← 0  
Para i de 1 ate 100 faca  
    soma ← soma + i  
Fimpara  
Escreva ("Soma dos números de 1 a 100: ", soma)  
finalgoritmo
```

Exemplo

- ▶ Ler e escreve os números ímpares de 1 a 100.

```
algoritmo "Usando_Para_Faca"  
var  
i: inteiro  
inicio  
Para i de 2 ate 100 faca passo 2  
    Escreval(i)  
Fimpara  
fimalgoritmo
```

Para...Faca...

- ▶ Para decrementar o comando Para..., torna-se necessário que o <valor inicial> seja maior que <valor final>, e o incremento deve ser negativo.

```
Algoritmo "Numeros de 10 a 1"  
Var  
    j: inteiro  
Inicio  
Para j de 10 ate 1 passo -1 faca  
    Escreva(j:3)  
Fimpara  
Fimalgoritmo
```

Enquanto...Faça...

- ▶ Esta estrutura de repetição se caracteriza por efetuar um teste lógico no início de um loop, verificando se é permitido executar o trecho de instruções abaixo dela.
- ▶ A estrutura ENQUANTO...FAÇA... tem o seu funcionamento controlado por condição.
- ▶ Desta forma, poderá executar um determinado conjunto de instruções enquanto a condição verificada permanecer verdadeira.
- ▶ No momento em que a condição se torna falsa, o processamento da rotina é desviado para fora do loop.
- ▶ Se a condição falsa logo no início do loop, as instruções contidas nele são ignoradas.

Enquanto...Faça...

▶ Sintaxe:

```
enquanto <expressão lógica ou relacional> faca  
  
    <lista de comandos>  
  
fimenquanto
```

Exemplo

```
Algoritmo "Usando_Enquanto"  
Var  
resp: caracter  
Inicio  
resp <- "s"  
Enquanto ((resp = "S") ou (resp = "s")) Faca  
    Escreval ("Entrei no Loop...")  
    Escreva ("Deseja continuar (S/N)? ")  
    Leia(resp)  
Fimenquanto  
Fimalgoritmo
```

Repita...Ate...

- ▶ Imagine agora, uma outra situação, onde o usuário deseja executar a rotina do programa várias vezes, mas este não sabe quantas vezes ao certo deverá executar o trecho de programa. Neste caso não seria conveniente manter um contador para controlar o looping, e sim, seria melhor que o programa fizesse ao usuário uma pergunta, solicitando se o mesmo deseja ou não continuar executando o programa.

Repita...Ate...

- ▶ Esta estrutura caracteriza-se por efetuar um teste lógico no final de um looping, sendo parecida com a estrutura ENQUANTO...FACA...
- ▶ Seu funcionamento é controlado também por decisão.
- ▶ Esta instrução irá efetuar a execução de um conjunto de instruções pelo menos uma vez antes de verificar a validade da condição estabelecida.

Repita...Ate...

▶ Sintaxe:

repita

<lista de comandos>

ate <expressão lógica ou relacional>

Exemplo

```
Algoritmo "Menu"  
Var  
opcao: inteiro  
Inicio  
Repita  
    Escreval ("Cadastro de clientes")  
    Escreval ("1-Incluir")  
    Escreval ("2-Alterar")  
    Escreval ("3-Consultar")  
    Escreval ("4-Excluir")  
    Escreval ("5-Sair")  
    Leia(opcao)  
Ate opcao = 5  
Fimalgoritmo
```

Exercícios

1. Escreva um programa que leia 50 valores e encontre o maior e o menor deles. Ao final apresente a soma do resultado.
2. Dado um número N , imprimir a tabuada de qualquer número n .
3. Dado um número N , calcular a soma de todos os números pares, inferiores ou iguais a esse número N .
4. Imprimir os múltiplos de 7 menores que 200.
5. Escrever um programa que calcula e escreve a soma dos números primos entre 10 e 100.

Exercícios

6. Faça um algoritmo que armazene os votos de cinco candidatos. Quando o usuário solicitar o resultado o algoritmo deve apresentar o total de votos de cada candidato, assim como o candidato vencedor.
7. Escreva um algoritmo que solicita um número inteiro positivo ao usuário e imprima a soma dos N primeiros números pares maiores do que zero (Por exemplo, se for informado $N = 4$, o algoritmo deve imprimir o valor 20, pois $2 + 4 + 6 + 8 = 20$).

Exercícios

8. Faça um algoritmo para ler cinco números e imprimir o cubo e o quadrado de cada um deles.
9. Elabore um algoritmo para calcular a soma dos números pares de 1000 a 10.
10. Faça um algoritmo para imprimir a quantidade de números ímpares entre um intervalo especificado pelo usuário, ou seja o usuário deverá fornecer dois números.