



 Faculdade de Economia, Administração e Contabilidade de Ribeirão Preto
Universidade de São Paulo 

RAD 2119 – Tecnologia de Informação Aplicada a Administração

Prof. Ildeberto A. Rodello
 rodello@fearp.usp.br
 <http://www.fearp.usp.br/~rodello>
 3602.0514

Desenvolvimento de Algoritmos



Estruturas de Repetição



Estruturas de Repetição

- Permitem que uma ou mais instruções sejam executadas:
 - um número determinado de vezes,
 - ou enquanto uma determinada condição não seja satisfeita.



Estruturas de Repetição - Tipos

- Condicional
 - Repetem até satisfazer uma condição de repetição.
 - São usadas quando não se sabe previamente quantas vezes deve-se executar as instruções do bloco de repetição.
- Contada
 - Repetem um número contado (pré-definido) de vezes.
 - São usadas quando se sabe previamente quantas vezes deve-se executar as instruções do bloco de repetição.



Estruturas de Repetição

- São classificadas em:
 - Condicional
 - Repetição com teste no início;
 - Repetição com teste no final;
 - Contada
 - Repetição com variável de controle.

Repetição com teste no início



Repetição com teste no início

- Permite repetir diversas vezes um mesmo trecho do algoritmo.
- Sempre verifica **antes** de cada execução se é permitido executar o mesmo trecho.
- Estrutura **enquanto...faça**

Sintaxe

```
enquanto <expressão lógica> faça  
    <comando 1>;  
    ...  
    <comando N>;  
fim enquanto;
```

Semântica

- Permite que um comando ou bloco de comandos sejam repetidos enquanto o resultado da <expressão lógica> for **verdadeiro**.
- Quando for **falso** a estrutura de repetição é abandonada.
- Se já da primeira vez o resultado for falso, o comando ou bloco de comandos não serão executados **nenhuma vez**.

Formas de controle

- **Contador:** inicializado antes da estrutura de repetição e incrementado no interior desta por um valor constante.
- **Acumulador:** inicializado antes da estrutura de repetição e incrementado no interior desta por um valor variável.

Contador - Sintaxe

```
...  
declare  
CON inteiro;  
início  
CON ← 0;  
enquanto CON < 3 faça  
    CON ← CON + 1;  
fim enquanto;  
fim.  
...
```

CON
3

Exemplo com contador

```
Algoritmo RepetiçãoContador  
declare  
    Nota 1, Nota 2, Média : Real;  
    Cont : Inteiro; {contador}  
início  
    Cont ← 0; {inicialização do contador}  
    enquanto ( Cont < 50) faça  
        Leia (Nota 1, Nota 2);  
        Média ← (Nota 1 + Nota 2) / 2;  
        se (Média >= 7) então  
            escreva ("Aluno aprovado. Média: ", Média);  
        senão  
            escreva ("Aluno reprovado. Média: ", Média);  
        fim se  
        Cont ← Cont + 1; {incremento do contador}  
    fim enquanto;  
fim algoritmo
```

Acumulador - Sintaxe

```
...  
declare  
CON, X, ACM inteiro;  
início  
  CON ← 0;  
  ACM ← 0;  
  enquanto CON < 3 faça  
    CON ← CON + 1;  
    leia (X);  
    ACM ← ACM + X;  
  fim enquanto;  
fim.  
...
```

CON	ACM	X
3	11	4

Exemplo com acumulador

```
Algoritmo RepetiçãoAcumulador;  
declare  
  MA, ACM, MAT real; { Média Anual Aluno, Acumulador, Média Anual Turma }  
  CON inteiro { contador }  
início  
  CON ← 0; { inicialização do contador }  
  ACM ← 0; { inicialização do acumulador }  
  enquanto (CON < 50) faça  
    leia (MA);  
    ACM ← ACM + MA; { soma em ACM os valores lidos em MA }  
    CON ← CON + 1; { incremento do contador }  
  fim enquanto;  
  MAT ← ACM / 50; { calculo da média anual da turma }  
  escreva ("média anual da turma = ", MAT);  
Fim algoritmo.
```

Repetição com teste no final

Repetição com teste no final

- Laço que verifica **depois** de cada execução, se é "permitido" continuar executando o trecho do algoritmo
- Trata-se de um laço que se mantém repetindo até que uma dada condição se torne verdadeira.
- Estrutura **repita...até**

Sintaxe

```
repita  
  <comando 1>;  
  <comando 2>;  
  ...  
  <comando n>;  
até <condição>;
```

Semântica

- Efetua um teste lógico **no fim** do laço, garantindo que **por pelo menos uma vez** as instruções deste são executadas.
- Ao contrário do enquanto, esta só repete o laço se o resultado do teste for **falso**.

Exemplo

```
Algoritmo Repetição;
declare
  MA, ACM, MAT real; { Média Anual Aluno, Acumulador, Média Anual Turma }
  CON inteiro { contador }
início
  CON ← 0; { inicialização do contador }
  ACM ← 0; { inicialização do acumulador }
repete
  leia (MA);
  ACM ← ACM + MA; { soma em ACM os valores lidos em MA }
  CON ← CON + 1; { incremento do contador }
até (CON >= 50); { teste da condição de parada }
MAT ← ACM / 50; { cálculo da média anual da turma }
escreva ("média anual da turma = ", MAT);
fim algoritmo.
```

Repetição com variável de controle

Repetição com variável de controle

- Laço simplificado para utilização em repetições de quantidade predeterminada.
- Incorpora internamente o funcionamento de um contador de repetições.

Sintaxe

```
para V de Vi até Vf passo P faça
  <comando 1>;
  <comando 2>;
  ...
  <comando n>;
fim para;
```

- Onde:
 - V é a variável de controle;
 - Vi é o valor inicial da variável V
 - Vf é o valor final da variável V
 - P é o valor do incremento dado a variável V (opcional)

Semântica

- Repete as instruções enquanto a variável contador não atingir o valor final.
- Ressalta-se que a variável contador é previamente inicializada e incrementada ou decrementada de uma constante a cada repetição.

Observações

- Sabe-se de antemão quantas vezes o bloco de repetição será executado. Isto é, repete enquanto o valor final não atingir o valor final da variável de controle.
- Incrementa automaticamente a variável de controle cada vez que o bloco é executado.
 - incremento 1 é o padrão.
- A variável de controle deve ser um número inteiro.
- A variável de controle não pode ser modificada dentro do bloco.

Exemplo

```

Algoritmo RepetiçãoVariavelControle;
declare
  MA, ACM, MAT real; { Média Anual Aluno, Acumulador, Média Anual Turma}
  V inteiro { contador }
início
  ACM ← 0; { inicialização do acumulador }
  para V de 1 até 50 faça
    leia (MA);
    ACM ← ACM + MA; { soma em ACM os valores lidos em MA }
  fim para;
  MAT ← ACM / 50; { cálculo da média anual da turma }
  escreva ("média anual da turma = ", MAT);
fim algoritmo.
    
```

Variável de controle com enquanto

- É possível implementar uma estrutura de repetição contada (**para..até**) com uma estrutura condicional (**enquanto ou repita**), entretanto o controle do incremento/decremento da variável contadora é seu.

Exemplo

```

Algoritmo ParaFaçaComEnquanto
declare
  x, i, contP: inteiro
Início
  contP ← 0;
  i ← 1;
  enquanto (i <= 20) faça
    escreva ("Digite um valor");
    leia (x);
    se (x > 0) então
      contP ← contP + 1;
    i ← i + 1; {Incremento explícito da variável contadora}
  fim enquanto;
  escreva ("positivos = "+contP);
fim algoritmo.
    
```

Comparação entre Estruturas de Repetição

Comparação

Estrutura	Condição	Quantidade de Execuções	Condição de Existência
Enquanto	Início	zero ou muitas	Condição verdadeira
Repita	Final	uma ou muitas	Condição falsa
Para	Não tem	$((vf - vi) \text{ div } p) + 1$	$v \leq vf$

Estrutura de Repetição Composta

Estrutura de Repetição Composta

- Semelhante às estruturas de decisão composta, as estruturas de repetição também podem ser **encadeadas e/ou aninhadas**.
- Esta abordagem é usada quando há a necessidade de se usar laços dentro de laços.

Exemplo

```
Algoritmo Tabuada
Declare
  resultado, n1, n2: inteiro
Início
  n1:=1;
  Enquanto (n1<=10) faça
    Para n2 ← 1 até 10 faça
      resultado ← n1 + n2;
      Escreva (n1," + ",n2," = ",resultado);
    Fim para
    n1 ← n1+1;
  Fim enquanto
Fim algoritmo.
```

Repetição e Seleção

- Também é possível encadear estruturas de repetição com estruturas de decisão.

Exemplo

```
Algoritmo TabuadaPareseÍmpares
declare
  resultado, num1, num2: inteiro
início
  n1 ← 1;
  enquanto (num1<=10) faça
    para num2 ← 1 até 10 faça
      resultado ← num1 * num2;
      se (r mod 2 = 0)
        escreva (num1," * ",num2," = ",resultado, " = Par");
      senão
        escreva (num1," * ",num2," = ",resultado, " = Impar");
    fim para
    num1 ← num1+1;
  fim enquanto
fim algoritmo.
```