

Módulo 2

Lógica de Programação II

REPETIÇÃO: Serve para efetuar um conjunto de ações repetidas vezes. Existem três tipos básicos de repetições, sendo elas.

Enquanto (Expressão Lógica) faça Seqüência	O comando Enquanto analisa a Expressão Lógica e enquanto o seu resultado for o valor lógico Verdade a Seqüência é executada.
Para variável VAR valor_inicial até valor_final faça Seqüência	O comando Para incrementa a variável a partir do valor_inicial de uma unidade até que esta atinja o valor_final. E para cada incremento a Seqüência é executada.
Repita Seqüência Até (Expressão Lógica)	O comando Repita executa a Seqüência até que o valor retornado pela Expressão Lógica seja Verdadeiro.

ESTRUTURA DE REPETIÇÃO DETERMINADA

Quando uma seqüência de comandos deve ser executada repetidas vezes, tem-se uma estrutura de repetição. A estrutura de repetição, assim como a de decisão, envolve sempre a avaliação de uma condição. Na repetição determinada o algoritmo apresenta previamente a quantidade de repetições.

Forma Geral 1:

```
PARA <<VARIABLE DE TIPO INTEIRO>>:=<<VALOR INICIAL>> ATE <<VALOR FINAL>> FAÇA
  <<COMANDO1>> ;
```

Forma Geral 2:

```
PARA <<VARIABLE DE TIPO INTEIRO>>:=<<VALOR INICIAL>> ATE <<VALOR FINAL>> FAÇA
  INICIO
    <<COMANDO1>> ;
    <<COMANDON>> ;
  FIM;
```

A repetição por padrão determina o passo do valor inicial até o valor final como sendo 1. Determinadas linguagens possuem passo -1 ou permitem que o programador defina o passo.

ALGORITMO SETE

Segue um algoritmo que escreve 10 vezes a frase "VASCO DA GAMA"

```
PROGRAMA REPETICAO;
VAR I:INTEIRO
INICIO
  PARA I:=1 ATE 10 FACA
    ESCREVA ('VASCO DA GAMA')
  FIM.
```

ALGORITMO OITO

Segue um algoritmo que escreve os 100 primeiros números pares.

```
>>
PROGRAMA PARES;
VAR I, PAR: INTEGER;
INICIO
  PAR:=0;
  PARA I:=1 ATE 100 FAÇA INICIO
    ESCREVA (PAR);
    PAR := PAR+2;
  FIM
FIM.
```

ESTRUTURA DE REPETIÇÃO INDETERMINADA COM VALIDAÇÃO INICIAL

É usada para repetir N vezes uma ou mais instruções. Tendo como vantagem o fato de não ser necessário o conhecimento prévio do número de repetições.

Forma Geral 1:

```
ENQUANTO <<CONDIÇÃO>> FAÇA
  <<COMANDO1>>;
```

Forma Geral 2:

```
ENQUANTO <<CONDIÇÃO>> FAÇA ÍNICIO
  <<COMANDO1>>;
  <<COMANDON>>;
FIM;
```

ALGORITMO NOVE

Segue um algoritmo que calcule a soma dos salários dos funcionários de uma empresa. O programa termina quando o usuário digitar um salário menor que 0.

```
PROGRAMA SOMA_SALARIOS;
VAR SOMA, SALARIO : REAL;
INICIO
  SOMA:=0;
  SALARIO:=1;
  ENQUANTO SALARIO>=0 FAÇA INICIO
    LEIA (SALARIO);
    SOMA:=SOMA+SALARIO
  FIM;
  ESCREVA (SOMA)
FIM.
```

OBS: Todas as variáveis que acumulam valores devem receber um valor inicial.

ESTRUTURA DE REPETIÇÃO INDETERMINADA COM VALIDAÇÃO FINAL

Assim como a estrutura ENQUANTO É usada para repetir N vezes uma ou mais instruções. Sua validação é final fazendo com que a repetição seja executada pelo menos uma vez.

Forma Geral;

```
REPITA
  <<COMANDO1>>;
  <<COMANDON>>;
ATE <<CONDIÇÃO>>;
```

ALGORITMO DEZ

Segue um algoritmo que calcule a soma dos salários dos funcionários de uma empresa. O programa termina quando o usuário digitar um salário menor que 0.

```
PROGRAMA SOMA_SALARIOS;
VAR
SOMA, SALARIO : REAL;
```

```

INICIO
  SOMA:=0;
  REPITA
    LEIA(SALARIO);
    SOMA:=SOMA+SALARIO
  ATE SALARIO<0;
  ESCREVA (SOMA)
FIM.

```

ALGORITMO ONZE

Segue um algoritmo que escreve os 100 primeiros números pares.

```

PROGRAMA PARES_2;
VAR
  I, PAR, CONTADOR : INTEIRO;
INICIO
  CONTADOR:=0;
  PAR:=0;
  REPITA
    ESCREVA(PAR);
    PAR:=PAR+2;
    CONTADOR:=CONTADOR+1;
  ATE CONTADOR=100
FIM.

```

VARIÁVEIS COMPOSTAS UNIDIMENSIONAIS (VETORES)

Até agora variáveis podiam comportar apenas uma informação por vez. Veremos agora que uma variável pode comportar um conjunto de informações primitivas, desde que devidamente ORGANIZADAS, DIVIDIDAS.

Para entender as variáveis compostas unidimensionais, imaginemos uma locomotiva com um número finito de vagões, representando uma estrutura de dados, e seus vagões, partições desta estrutura.

DECLARAÇÃO

tipo LISTA = vetor(CI CF) de < tipo >;
 LISTA = lista de variáveis;

ONDE

CI (Coluna Inicial) – representa o limite inicial do vetor;
 CF (Coluna Final) – representa o limite final do vetor;
 < tipo > - representa qualquer um dos tipos primitivos de variáveis;

EXEMPLO

tipo CLASSE = vetor(1 ... 40) de reais;
 CLASSE : NÚMEROS;

NÚMEROS

8.5	7.4	5.8	2.9	9.2	7.3	5.4
1	2	3	4	5		39	40

Podemos notar que CI e CF são sempre números inteiros e CF sempre será maior que CI.

O número de elementos do vetor será dado por $CF - CI + 1$. Isto significa que as posições do vetor são identificadas a partir de CI, com incrementos unitários, até CF.

CI	CI + 1	CI + 2	CI + 3	CI + 4	CF - 1	CF
----	--------	--------	--------	--------	-------	--------	----

>>

OUTROS EXEMPLOS

tipo CLASSE = vetor(1 ... 50) de inteiros;
 CLASSE : VET1, VET2, VET3;

VET1								
9	8	7	6	5	4	2	
1	2	3	4	5		39	40	
VET2								
8	7	5	2	9	7	5	
1	2	3	4	5		39	40	
VET3								
2	2	2	2	2	2	2	
1	2	3	4	5		39	40	

EXERCÍCIOS DE FIXAÇÃO

01- Ler e armazenar um conjunto de 30 números inteiros.

```

programa trinta;
variáveis
tipo NUMEROS = vetor ( 1 ... 30 ) de inteiros;
NUMEROS : CONJUNTO;
NUM, CONT: inteiro;
inicio
. CONT := 1 ;
. faça enquanto CONT <= 30 ;
.   . leia( NUM ) ;
.   . CONJUNTO ( CONT ) := NUM;
.   . CONT := CONT + 1;
. fimfaça;
fim.

```

02- Construa um vetor de nome ALUNOS com 39 posições onde cada uma delas será preenchida pelo nome dos alunos desta classe.

```

programa alunos;
variáveis
tipo NMALUNO = vetor ( 1 ... 39 ) de caracteres;
NMALUNO : ALUNOS;
NOMES: caracter;
CONT: inteiro;
. CONT := 1 ;
. faça enquanto CONT <= 39 ;
.   . leia( NOMES ) ;
.   . ALUNOS ( CONT ) := NOMES;
.   . CONT := CONT + 1;
. fimfaça;
fim.

```

VARIÁVEIS COMPOSTAS BIDIMENSIONAIS (MATRIZES)

Os vetores têm como principal característica a necessidade de apenas um índice para endereçamento – são estruturas unidimensionais.

Uma estrutura que precisa de mais de um índice são denominadas estruturas multidimensionais.

Suponha que, além do acesso pelo elevador até um determinado andar, tenhamos também a divisão desse andar em apartamentos. Para chegar a algum deles não basta só o número do andar, precisamos também do número do apartamento. Neste caso precisamos de dois números para chegarmos onde queremos, ou seja, dois índices. Chamamos de estrutura bidimensional (MATRIZ) aquela que precisa de dois índices para chegar ao seu conteúdo.

DECLARAÇÃO

tipo IDENTIFICADOR = matriz(LI...LF, CI...CF) de < tipo >;
IDENTIFICADOR: lista de variáveis;

ONDE

LI...LF, CI...CF – são os limites dos intervalos de variação dos índices da variável, onde cada par de limites está associado a um índice;
LI...LF à linhas da matriz;
CI...CF à colunas da matriz.
< tipo > - representa qualquer um dos tipos primitivos de variáveis;

EXEMPLO

tipo SALA = matriz(1 ... 4, 1...4) de inteiros;
SALA: MSALA;

	1	2	3	4
1				
2				
3				
4				

Matriz MSALA

O intervalo em destaque é a célula (APARTAMENTO) MSALA(2, 3)

Para utilizar o vetor, nós o inserimos em um único laço de repetição, fazendo com que haja variação do seu índice. Como numa estrutura multidimensional possuímos mais de um índice, faz-se necessária a utilização de mais laços de repetição. As matrizes mais utilizadas são as bidimensionais que devem possuir dois laços de repetição.

EXERCÍCIOS DE FIXAÇÃO

01- Ler e armazenar um conjunto de 36 números inteiros em uma matriz.

```
Programa trinta e seis;
Variáveis
tipo NUMEROS = matriz ( 1...6, 1...6 ) de inteiros;
NUMEROS : NUM1;
NUM, CONTL, CONTC: inteiro;
inicio
CONTL := 1 ;
faça enquanto CONTL <= 6 ;
.   CONTC := 1 ;
.   faça enquanto CONTC <= 6 ;
.   .   leia( NUM ) ;
.   .   NUM1 ( CONTL, CONTC ) = NUM;
.   .   CONTC := CONTC + 1;
.   fimfaça;
.   CONTL := CONTL + 1 ;
fimfaça;
fim.
```

02- Construa uma matriz de nome IDADE com 18 posições onde cada uma delas será preenchida pela idade de um conjunto de pessoas.

```

>>
Programa idade;
Variáveis
tipo ALUNOS = matriz ( 1..6, 1..3 ) de inteiros;
ALUNOS : IDADE;
ID, CONTL, CONTC: inteiro;
início
CONTL := 1 ;
faça enquanto CONTL <= 6 ;
.   CONTC := 1 ;
.   faça enquanto CONTC <= 3 ;
.   .   ID := -1 ;
.   .   faça enquanto ID <= 0 ;
.   .   .   leia( ID )
.   .   fimfaça ;
.   .   IDADE ( CONTL, CONTC ) = ID;
.   .   CONTC := CONTC + 1;
.   fimfaça;
.   CONTL := CONTL + 1 ;
fimfaça;
fim.

```

EXERCÍCIOS (Resolva todos os exercícios e concorra a um brinde no final do curso)

- 01) Faça um algoritmo que imprima a tabuada do número 7.
- 02) Faça agora o mesmo algoritmo para calcular a tabuada de qualquer número.
- 03) Faça um algoritmo que leia 50 números reais e escreva o número maior e o menor deles.
- 04) Construa um algoritmo que leia um conjunto de dados contendo altura e sexo (MAS ou FEM) de N pessoas e calcule e escreva:
- a maior e a menor altura do grupo e de quem é esta altura. (Se HOMEM ou MULHER.)
 - a média de altura entre as mulheres.
 - O número de homens e a diferença percentual entre estes e as mulheres.
- 05) Elabore um algoritmo que efetue a soma de todos os números inteiros ímpares que são múltiplos de três e que se encontram no conjunto dos números de 1 até 500.
- 06) Preciso fazer um levantamento do meu estoque de mercadorias que tenho na minha loja de artigos de construção. Para isso preciso saber qual o material que vendo e a quantidade de cada material. Me ajude, fazendo a entrada e a saída de mercadorias da minha loja no último mês de venda. O algoritmo termina quando encontrar a palavra "ACABOU". Minhas mercadorias com o estoque de um mês atrás são:
- | | | |
|------------------|-----------------------|------------------------|
| CIMENTO 40 sacos | CAL 50 alqueires | TIJOLO 15.000 unidades |
| AREIA 200 m3 | TELHA 25.650 unidades | |
- 07) Durante uma corrida de automóveis com N voltas de duração foram anotados para um piloto os tempos, em minutos, registrados em cada volta. Faça um algoritmo que calcule:
- seu melhor tempo.
 - A volta em que o melhor tempo ocorreu.
 - O tempo médio das N voltas.
- 08) Uma locadora de CDs e DVDs está precisando fazer um sistema de computação para organizar seu negócio. As informações mais importantes para o dono é:
- quantos filmes estão alugados.
 - quantos clientes estão com seus filmes.
 - Quantos clientes pagam a vista.
 - Quantos clientes só pagam quando devolvem o filme.
 - Quanto será o apurado da loja na devolução de todos os filmes.
- Faça um algoritmo que escreva estas informações para a locadora.
- Observação à O algoritmo termina quando em uma variável de nome SAIR estiver o conteúdo "SIM".
- 09) Faça um algoritmo para ler e montar um vetor de números inteiros com 50 posições. Depois de montado, o vetor ler 10 números inteiros quaisquer e para cada número lido imprimir se este existe ou não no vetor.
- 10) Faça um algoritmo que leia e monte dois vetores 1 e 2 de N posições com números reais positivos. Depois de montados procure por cada elemento do vetor 1 no vetor 2 e armazene as suas respectivas posições (somente do primeiro número achado) em um vetor 3, sendo que os elementos do vetor 1 que não forem encontrados no vetor 2 deverão ter como valor o número zero no vetor 3.
- 11) Ler um vetor de nomes e imprimir os nomes das posições PARES e em seguida os nomes das posições IMPARES.
- 12) Ler um conjunto de nomes e notas dos alunos de uma turma e imprimir os nomes dos alunos que tiveram nota acima da média das notas da turma.

- 13) Faça um algoritmo que construa uma matriz M por N de números reais e se esta matriz for quadrada ($M = N$) imprimir sua diagonal principal.
- 14) Faça um algoritmo que construa uma matriz 50×30 e depois de construída inverter suas linhas. Exemplo: a última linha será a primeira e a primeira a última. A antepenúltima será a segunda e a segunda será antepenúltima e assim por diante.
- 15) Faça um algoritmo para ler 3 notas de um conjunto de 60 alunos de uma turma armazenando numa matriz, onde os números das linhas representam os números dos alunos e cada coluna uma de suas três notas. Depois de construída a matriz imprimir o número do aluno e sua respectiva média e no final a média geral da turma.

ADQUIRA JÁ A SÉRIE DE CDS COMPLETE



260 Programas Com Fontes Prontos para Usar e Comercializar!

Entre eles:
SIGE PLUS 7.0
LAN-MAXX
SIS CLÍNICA 3.5
SIS COMÉRCIO
SIS HOTEL

E mais:
Mais de 150 apostilas
6 Aulas em Vídeo
Mais de 740 Componentes
Mais de 30 Programas Utilitários
Mais de 11.000 Glyphs
Mais de 18.000 Ícones
Mais de 400 Cursores Animados

3.158 Documentos

Apostilas, Livros, E-Books, Tutoriais, Dicas, How-Tos, Cursos e muito mais...

2.052 Documentos de Informática
Apenas de programação são 969 documentos.

E mais:

513 Documentos Profissionais
423 E-Books (Excelente para Vestibular)
21 Documentos para Prosperidade
6 Cursos de Idiomas (Espanhol, Francês, Latim, Esperanto, Japonês, Italiano)

E outros 143 Documentos!



114 Aulas em Vídeo

- * Curso Completo de FireWorks em 20 Aulas
- * Curso Completo de PHP em 25 Aulas
- * Curso de Delphi em 6 Aulas
- * Curso Visual Basic em 8 Aulas
- * Curso Completo Flash MX
- * Curso PhotoShop 7
- * Curso Completo Visual Studio.Net em 45 Aulas



Acesse o site www.alberteije.com e veja mais detalhes sobre cada CD e como Adquirí-los.