

Programação de Computadores III

Aula 16

Professor Leandro Augusto Frata Fernandes
laffernandes@ic.uff.br

Material disponível em
<http://www.ic.uff.br/~laffernandes/teaching/2011.1/tcc-03.063>

Vetores (aula anterior)

- Estruturas de dados **composta**
 - Permitem a uma variável armazenar **múltimos valores** de um **mesmo tipo** primitivo

- Por exemplo, ao invés de declarar 40 variáveis assim:

```
real: Nota1, Nota2, Nota3, ..., Nota40
```

podemos declarar uma única variável assim:

```
real: Nota(40)
```

e acessar o i -ésimo valor por indexação,
para $1 \leq i \leq 40$:

```
Nota(i) ← 3.14
```

Declaração de Vetores em FORTRAN

- Forma geral

`<tipo> <nome> (<início>:<fim>)`

- `<início>` e `<fim>` são valores inteiros que podem ser negativos, nulos ou positivos
- `<fim>` precisa ser maior ou igual a `<início>`
- Quando `<início>` é omitido, o valor 1 é assumido

Declaração de Vetores em FORTRAN

- Exemplo 1

`integer Num (3:8)`

	3	4	5	6	7	8
Num						

- Exemplo 2

`integer Num (-3:2)`

	-3	-2	-1	0	1	2
Num						

- Exemplo 3

`integer Num (6)`

	1	2	3	4	5	6
Num						

Declaração e Utilização de Vetores em FORTRAN

- Exemplo 4 (cada entrada armazena 20 caracteres)

<code>character*20 Nome(0:2)</code>	Nome	<table><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>☠️🧠☹️</td><td>☠️🧠☹️</td><td>☠️🧠☹️</td></tr></table>	0	1	2	☠️🧠☹️	☠️🧠☹️	☠️🧠☹️
0	1	2						
☠️🧠☹️	☠️🧠☹️	☠️🧠☹️						
<code>Nome(0) = 'Maria'</code>	Nome	<table><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>'Maria'</td><td>☠️🧠☹️</td><td>☠️🧠☹️</td></tr></table>	0	1	2	'Maria'	☠️🧠☹️	☠️🧠☹️
0	1	2						
'Maria'	☠️🧠☹️	☠️🧠☹️						
<code>Nome(2) = 'Paula'</code>	Nome	<table><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>'Maria'</td><td>☠️🧠☹️</td><td>'Paula'</td></tr></table>	0	1	2	'Maria'	☠️🧠☹️	'Paula'
0	1	2						
'Maria'	☠️🧠☹️	'Paula'						
<code>Nome(1) = 'Afonso'</code>	Nome	<table><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>'Maria'</td><td>'Afonso'</td><td>'Paula'</td></tr></table>	0	1	2	'Maria'	'Afonso'	'Paula'
0	1	2						
'Maria'	'Afonso'	'Paula'						

TCC-03.063 Programação de Computadores III (2011.1)

5

Utilização de Vetores em FORTRAN

- Uma posição do arranjo é acessada pelo seu índice, seja para atribuir ou recuperar o valor

```
real Nota(3)

Nota(1) = 5.0
Nota(2) = 8.0
Nota(3) = (Nota(1) + Nota(2)) / 2.0
```

	1	2	3
Nota	5.0	8.0	6.5

TCC-03.063 Programação de Computadores III (2011.1)

6

Exemplo

- Calcule a média da nota de 40 alunos e informe quantos conseguiram nota acima da média

Veja o arquivo p13vetor.f

Exercício

1. Transcreva o algoritmo criado para o Exercício 1 da aula anterior em um programa FORTRAN.

Facilitadores fornecidos pelo FORTRAN

- O FORTRAN fornece comandos que ajudam na manipulação de variáveis compostas
- Comando `PARAMETER`
 - Permite criar constantes que podem ser utilizadas para definir o tamanho de variáveis compostas
 - Vantagem: a alteração do tamanho é feita em um lugar
- Comando `DO` implícito
 - Permite combinar de forma resumida o comando `DO` com os comandos `PRINT` e `READ`
 - Vantagem: reduz o tamanho do código fonte

TCC-03.063 Programação de Computadores III (2011.1)

9

Exemplo de `PARAMETER`

ANTES

```
integer i
real Nota(40)

do i = 1, 40
  read *, Nota(i)
end do

do i = 1, 40
  print *, Nota(i)
end do
```

DEPOIS

```
integer TNota
parameter (TNota = 40)

integer i
real Nota(TNota)

do i = 1, TNota
  read *, Nota(i)
end do

do i = 1, TNota
  print *, Nota(i)
end do
```

Mude somente aqui!



TCC-03.063 Programação de Computadores III (2011.1)

10

Exemplo de DO Implícito

ANTES

```
integer TNota
parameter (TNota = 40)

integer i
real Nota(TNota)

do i = 1, TNota
  read *, Nota(i)
end do

do i = 1, TNota
  print *, Nota(i)
end do
```

DEPOIS

```
integer TNota
parameter (TNota = 40)

integer i
real Nota(TNota)

read *, (Nota(i), i=1, TNota)

print *, (Nota(i), i=1, TNota)
```

Exemplo

- Calcule a média da nota de 40 alunos e informe quantos conseguiram nota acima da média, mas agora utilize PARAMETER e DO implícito.

Veja o arquivo p14vetor.f

Exercícios

2. Utilize o comando `PARAMETER` e `DO` implícito para simplificar o código fonte `FORTRAN` escrito para o exercício anterior.
3. Escreva um programa `FORNTRAN` que leia o nome e idade de 10 pessoas e, depois, mostre os dados lidos. Utilize o comando `PARAMETER` e `DO` implícito neste programa.
4. Modifique o programa do Exercício 3 para fazer a leitura de nome e idade de 15 pessoas, e mostrar os dados lidos em ordem crescente de idade.