



UNIVERSIDADE ESTADUAL DE MS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
AEDI

Registro

Prof. Nilton

- Para algumas aplicações são necessários conjuntos de elementos do mesmo tipo, e para tais usamos vetores ou matrizes;
- Alguns problemas há necessidade de definirmos conjuntos onde os elementos não são do mesmo tipo;
- Por exemplo, conjunto de informações que representa um aluno: nome, cpf, rg, data de nascimento, idade, sexo, coeficiente de rendimento, etc;

- Uma solução para definir as informações de um aluno é:

var

```
nome, cpf, data_nasc, sexo : caracter;  
rg, idade : inteiro;  
coef_rend : real;
```

- Se considerarmos a existência de 100 alunos uma possível definição é:

var

```
nome, cpf, data_nasc, sexo : vetor [1..100] de caracter;  
rg, idade : vetor [1..100] de inteiro;  
coef_rend : vetor[1..100] de real;
```

– 7 vetores!

- Podemos trocar alguns vetores por matrizes, onde cada linha representa informações de um aluno e cada coluna uma informação específica do aluno:

```
var
    aluno1 : matriz[1..100, 1..4] de : caracter;
           //c1:nome, c2:cpf, c3:data_nasc, c4:sexo
    aluno2: matriz[1..100, 1..2] de inteiro;
           //c1:rg, c2:idade
    coef_rend : vetor[1..100] de real;
```

- Podemos criar algumas associações lógicas entre as informações de uma entidade usando estruturas homogêneas, mas para manter os dados consistentes torna-se trabalhoso;
- Com a utilização de um registro podemos minimizar o trabalho;

Registro

- Estrutura que agrupa diferentes tipos de dados em uma única variável;

Tipo

<identificador> = **registro**

nome1 : <tipo_primitivo1>;

nome2 : <tipo_primitivo2>;

...

nomeN : <tipo_primitivoN>;

fim_registro;

Um registro para aluno: definição

Tipo

```
CAD_ALUNO = registro  
    nome : character;  
    cpf : character;  
    data_nasc : character;  
    sexo : character;  
    rg : inteiro;  
    idade : inteiro;  
    coef_rend : real;  
fim_registro;
```

Usando o registro aluno

Var

```
    RALUNO : CAD_ALUNO;  
  
...  
leia(RALUNO.nome)
```

Outro exemplo de registro

Tipo

```
CAD_PESSOA = registro
    nome : character;
    cpf : character;
    telefone : vetor[1..5] de character;
fim_registro;
```

Usando o registro pessoa

Var

```
    RPESSOA : CAD_PESSOA;
...
para i de 1 ate 5 faça
    leia(RPESSOA.telefone[i])
fimpara
```

Combinando registros e vetores

- **Registros e vetores podem se compostos a fim de resolver problemas mais complexos;**
- **Sendo um vetor um conjunto de elementos do mesmo tipo, é natural que, quando precisarmos agrupar vários registros (por exemplo, vários cadastros), utilizaremos vetores (conjuntos) de registros;**

Usando um conjunto de registros

Tipo

```
CAD_ALUNO = registro  
    nome : character;  
    cpf : character;  
    data_nasc : character;  
    sexo : character;  
    rg : inteiro;  
    idade : inteiro;  
    coef_rend : real;  
fim_registro;
```

Var

```
    RALUNOS : vetor[1..100] de CAD_ALUNO;  
  
...  
para i de 1 até 100 faça  
    leia(RALUNOS[i].nome)  
fimpara
```

Definindo um conjunto de registros

Tipo

```
CAD_ALUNO = registro  
    nome, cpf, data_nasc, sexo : character;  
    rg , idade: inteiro;  
    coef_rend : real;  
fim_registro;  
  
VALUNOS : vetor[1..100] de CAD_ALUNO;
```

Usando um conjunto de registros

Var

```
    RALUNOS : VALUNOS;  
  
...  
para i de 1 até 100 faça  
    leia(RALUNOS[i].nome)  
Fimpara  
  
...  
RALUNOS[5].coef_rend ← 7.5;
```

Registro e parâmetro por valor

Tipo

```
CAD_ALUNO = registro  
    nome, cpf, data_nasc, sexo : character;  
    rg , idade: inteiro;  
    coef_rend : real;  
fim_registro;
```

Var

```
RALUNOS : vetor[1..100] de CAD_ALUNO;
```

...

```
procedimento mostraAluno(r : CAD_ALUNO)
```

```
inicio
```

```
    escreva(r.nome, r.cpf, r.data_nasc, r.sexo, r.rg, r.idade, r.coef_rend)
```

```
fimprocedimento
```

...

```
para i de 1 até 100 faça
```

```
    mostraAluno( RALUNOS[i] )
```

```
fimpara
```

Registro e parâmetro por referência

Tipo

```
CAD_ALUNO = registro  
    nome, cpf, data_nasc, sexo : character;  
    rg , idade: inteiro;  
    coef_rend : real;  
fim_registro;
```

Var

```
RALUNOS : vetor[1..100] de CAD_ALUNO;
```

...

```
procedimento leAluno(var r : CAD_ALUNO)
```

```
inicio
```

```
    leia(r.nome, r.cpf, r.data_nasc, r.sexo, r.rg, r.idade, r.coef_rend)
```

```
fimprocedimento
```

...

```
para i de 1 até 100 faça
```

```
    leAluno( RALUNOS[i] )
```

```
fimpara
```

Conjunto de registros e parâmetro por valor

Tipo

CAD_ALUNO = **registro**

nome, cpf, data_nasc, sexo : caracter;

rg , idade: inteiro;

coef_rend : real;

fim_registro;

VALUNOS : vetor[1..100] de CAD_ALUNO;

Var

RALUNOS : VALUNOS;

...

procedimento mostraAlunos(cr : VALUNOS)

var

i : inteiro

inicio

para i de 1 até 100 faça

escreva(cr[i].nome, cr[i].cpf, ...)

fimpara

fimprocedimento

...

mostraAlunos(RALUNOS)

Conjunto de registros e parâmetro por referência

Tipo

CAD_ALUNO = **registro**

nome, cpf, data_nasc, sexo : caracter;

rg , idade: inteiro;

coef_rend : real;

fim_registro;

VALUNOS : vetor[1..100] de CAD_ALUNO;

Var

RALUNOS : VALUNOS;

...

procedimento leAlunos(var cr : VALUNOS)

var

i : inteiro

inicio

para i de 1 até 100 faça

leia(cr[i].nome, cr[i].cpf, ...)

fimpara

fimprocedimento

...

leAlunos(RALUNOS)

Exercícios

1- Defina um registro capaz de armazenar as seguintes informações sobre um determinado cliente de um banco: nome, CPF, RG, número da conta, data de abertura da conta e saldo. A data deve ser definida também como um registro composto de dia, mês e ano.

2- Dados os seguintes campos de um registro: nome, telefone, dia de aniversário e mês de aniversário, desenvolver um algoritmo que mostre em um dado mês do ano, quem são as pessoas que fazem aniversário, exibir também o dia. Considere um conjunto de 40 pessoas.

3- Uma pessoa cadastrou um conjunto de 15 registros contendo o nome da loja, telefone e preço de um eletrodoméstico. Desenvolver um algoritmo que permita exibir qual foi a média dos preços cadastrados e uma relação contendo o nome e o telefone das lojas cujo preço estava abaixo da média.

Exercícios

4- Suponha um cadastro de participantes com um registro contendo Nome e CPF do aluno, tipo de participação (A, B, C ou D) e sócio da SBC (S-sim ou N-não). Desenvolver um algoritmo para calcular o valor que cada aluno vai pagar para participar do **SBBD 2019**, sabendo-se que:

Tipo de Participação	Valor a Pagar
A - 1 curso	R\$ 30,00
B - 2 cursos	R\$ 60,00
C - 3 cursos	R\$ 90,00
D - outros	R\$100,00

Para os sócios da SBC o valor a pagar terá um desconto de 50%. O algoritmo deverá permitir a entrada de vários registros (no máximo 1000) até que uma condição de finalização seja satisfeita. Calcular e exibir também o total geral arrecadado com o evento e quantos alunos se matricularam em cada um dos tipos de participação.

Exercícios

5- Produtos são comercializados por uma loja que precisa manter as informações como nome do produto, código do produto, nomes de fornecedores, quantidade mínima em estoque, quantidade real em estoque, ultima data de venda e de compra e qual fornecedor fez o último fornecimento. A loja mantém 50 produtos diferentes e cada produto pode ser fornecido por até 4 fornecedores. Construa um algoritmo que:

- a) Tenha módulos para inicialização do conjunto de registros e para cadastro dos produtos;
- b) Tenha um módulo para imprimir quais produtos cuja quantidade real em estoque é inferior ou igual a quantidade mínima em estoque. Deve ser impresso o nome do produto, seus fornecedores, o fornecedor que fez último fornecimento e as quantidades em estoque;
- c) Tenha um módulo que determine quantos produtos um determinado fornecedor fornece. Um dos parâmetros do módulo é o fornecedor;
- d) Determine os códigos dos produtos fornecidos por cada fornecedor. Use tantos módulos que forem necessários;