



Universidade Federal do Vale do São Francisco
Curso de Engenharia de Computação



Introdução a Algoritmos – Parte 07

(Baseado no Material do Prof. Marcelo Linder)

Prof. Jorge Cavalcanti

jorge.cavalcanti@univasf.edu.br

www.univasf.edu.br/~jorge.cavalcanti

www.twitter.com/jorgecav

Estruturas de dados heterogêneas

Registros

- Vimos inúmeras aplicações onde são necessários conjuntos de elementos do mesmo tipo, e para tal utilizamos os vetores.
- No entanto em alguns problemas há necessidade de definirmos conjuntos onde os elementos não sejam do mesmo tipo.
- Um típico exemplo de nosso cotidiano é utilização do conjunto de informações que caracterizam um aluno: Nome(caractere), CPF(inteiro), RG(inteiro), data de nascimento(caractere), coeficiente de rendimento(real), etc..

Estruturas de dados heterogêneas

Registros

- Em uma análise superficial poderia se pensar que uma solução para a questão apresentada pode ser obtida declarando-se cinco variáveis:

algoritmo "exemplo"

var Nome: caractere

CPF: inteiro

RG: inteiro

data_de_nascimento: caractere

coeficiente_de_rendimento: real

...

- Para uma melhor visualização da utilidade dos registros basta imaginarmos que ao invés de manipular as informações de um aluno exista a necessidade de gerenciamento de uma turma com cinquenta alunos.

Estruturas de dados heterogêneas

Registros

- Um estudante desatento imaginaria ser necessário a declaração de 250 variáveis. Porém um estudante com uma visão adequada sugeriria a utilização de cinco vetores:

algoritmo "exemplo"

var Nomes: vetor [1..50] de caractere

CPFs: vetor [1..50] de inteiro

RGs: vetor [1..50] de inteiro

datas_de_nascimento: vetor [1..50] de caractere

coeficientes_de_rendimento: vetor [1..50] de real

...

- Porém, manipular de forma adequada os vetores, mantendo seus dados consistentes, se torna trabalhoso. Com a utilização de um **registro** podemos resolver este problema apenas com um vetor de cinquenta registros.

Estruturas de dados heterogêneas

Registros

- Uma maneira mais interessante de representar esse tipo de informação seria o agrupamento dessas informações de origem comum em um única estrutura: **o Registro**.
- Analogamente a um registro em uma ficha de papel que é utilizada para armazenar as informações de um aluno, paciente, cliente etc., há as variáveis compostas heterogêneas ou registros, que podem armazenar tais informações no algoritmo.

```
var Ficha_Cadastro: registro  
inicio  
matricula: inteiro  
nome: vetor [1..50] de caractere  
CPF: vetor [1..11] de inteiro  
RG: vetor [1..20] de inteiro  
data_de_nascimento: vetor [1..9] de caractere  
fim
```

- A variável `Ficha_Cadastro` agrupa as informações referentes a um aluno da academia.

Estruturas de dados heterogêneas

Registros

- Com base no exemplo anterior podemos extrair a estrutura geral para a declaração de um registro:

<nome_da_variavel>: registro

inicio

<nome_campo_1>: <tipo_campo_1>

<nome_campo_2>: <tipo_campo_2>

...

<nome_campo_n>: <tipo_campo_n>

fimregistro

- Cada campo funciona como uma variável primitiva, mas todos os campos estão agrupados no registro, conforme a variável **Ficha_Cadastro**.

Estruturas de dados heterogêneas

Registros

- Para acessar esse tipo de variável, é necessário especificar o registro nominando o campo que se deseja utilizar.
- Os registros não podem ser acessado sem a especificação individual dos campos. As operações a seguir são inválidas:

leia (Ficha_Cadastro)

escreva (Ficha_Cadastro)

- Em vez disso, é necessário acessar cada campo individualmente:

leia (Ficha_Cadastro.matricula)

escreva (Ficha_Cadastro.nome)

Ficha_Cadastro.cpf <- 123.456.789-10

- A definição de tipos específicos para a declaração de registros auxiliar na melhora da legibilidade do código. Essa opção será vista logo adiante, na utilização de registros com vetores.

Estruturas de dados heterogêneas

Registros e vetores

- Registros e vetores podem se compostos a fim de resolver problemas mais complexos.
- Considerando que um vetor é um conjunto de elementos do mesmo tipo, é natural que, quando precisarmos agrupar vários registros (por exemplo, várias Fichas Cadastro), utilizaremos vetores (conjuntos) de registros.
- Eventualmente, podemos ter necessitar que um registro tenha como um dos seus campos um vetor (como o de caracteres para o Nome).
- Os vetores de registros visam a armazenar conjuntos de elementos complexos. Ao invés de armazenar apenas as notas dos alunos de uma turma, um vetor de registro pode armazenar as informações cadastrais de todos os alunos da turma.

Estruturas de dados heterogêneas

Registros e vetores

- Exemplo:

algoritmo "exemplo"

var alunos: vetor [1..50] de registro

inicio

nome: caractere

CPF: inteiro

RG: inteiro

datas_de_nascimento: caractere

coeficientes_de_rendimento: real

fimregistro

...

- A cada um dos elementos que constituem um registro é dado o nome de campo. No exemplo acima, temos os campos: nome, CPF, RG, datas_de_nascimento e coeficientes_de_rendimento.

Estruturas de dados heterogêneas

Definição de novos tipos de dados

- Abriremos um parêntese em nosso estudo sobre registros para falarmos sobre **definição de tipo de dado**.
- Com o objetivo de facilitar a leitura e, conseqüentemente, o entendimento dos algoritmos construídos, foi criado o conceito de definição de tipo de dado.

Sintaxe:

tipo <nome_do_tipo>: <definicao_do_tipo>

Exemplo:

tipo vetor_de_inteiros: vetor [1..100] de inteiro

- As definições de tipos devem ser feitas entre a constante caractere que nomeia o algoritmo e a declaração de variáveis globais ou dos módulos.
- Assim, após criado um novo tipo, variáveis podem ser definidas para esse novo tipo. Ex: var num: vetor_de_inteiros

Estruturas de dados heterogêneas

Registros e vetores

■ A definição de tipos diferentes dos primitivos auxilia a legibilidade do código, além de flexibilizar a criação de variáveis homogêneas e heterogêneas.

Ex: tipo vetor100: Vetor [1..100] de real
var notas: vetor100

tipo ficha_cad: registro
início

<campo 1>: <tipo do campo>

<campo 2>: <tipo do campo>

...

<campo n>: <tipo do campo>

var cadastro: ficha_cad

Estruturas de dados heterogêneas

Registros e vetores

- Com a utilização dos conceitos de criação de novos tipos, podemos resolver o problema do cadastro de alunos da seguinte forma:

algoritmo "exemplo"

tipo registro_aluno: registro

inicio

nome: caractere

CPF: inteiro

RG: inteiro

datas_de_nascimento: caractere

coeficientes_de_rendimento: real

fimregistro

tipo vetor_de_registros: vetor [1..50] de registro_aluno

var alunos: vetor_de_registros

Estruturas de dados heterogêneas

Registros

- Um ou mais campos de um registro pode ser do tipo vetor.
- A construção do registro é feita da mesma forma, porém o vetor a ser utilizado em sua estrutura deve ser declarado anteriormente.

algoritmo "exemplo"

tipo vet : vetor [1..4] de real

tipo registro_aluno: registro

inicio

nome: caractere

CPF: inteiro

RG: inteiro

datas_de_nascimento: caractere

notas: vet

fimregistro

tipo vetor_de_registros: vetor [1..50] de registro_aluno

var alunos: vetor_de_registros

Estruturas de dados heterogêneas

Conjuntos de registros

- No conjunto de registros são armazenados dados de várias ocorrências de um determinado tipo de registro. Por exemplo, para armazenar os dados de diversos alunos.
- Da mesma forma com que trabalhamos com um vetor acessando-o elemento a elemento, seja para atribuição ou seja para consulta de um valor, o mesmo ocorre com relação aos registros, devemos acessá-lo campo a campo.
- Para acessarmos um determinado campo de um registro devemos utilizar o operador "." da seguinte forma:
 - No caso do exemplo com o qual temos trabalhado, a leitura do campo nome do décimo segundo aluno da turma é feita através de
leia(alunos[12].nome)
 - ou a impressão na saída padrão do CPF do terceiro aluno seria feita da seguinte forma
escreva(alunos[3].cpf)

Estruturas de dados heterogêneas

Registros e vetores

Exercício 1:

Defina um tipo de dado 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.

16

fimregistro

Estruturas de dados heterogêneas

Registros - Exercícios

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.



Estruturas de dados heterogêneas

Registros - 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 da **ERBASE 2013**, 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.

Estruturas de dados heterogêneas

Exercício 5:

Com base no exercício 1, construa um algoritmo que manipule um vetor com 15 registros de clientes, onde cada registro é um elemento do tipo de dado definido.

A manipulação do vetor é feita através dos seguintes módulos: inicializar vetor, imprimir um determinado registro com base no valor do campo CPF e imprimir um determinado registro com base em sua posição no vetor.

O algoritmo deve se utilizar de forma satisfatória dos módulos mencionados.



Estruturas de dados heterogêneas

algoritmo "Exercício sobre registros"

tipo data: registro

inicio

dia: inteiro

mes: inteiro

ano: inteiro

fimregistro

tipo registro_conta: registro

inicio

nome: caractere

cpf: caractere

rg: caractere

numero_conta: inteiro

data_abertura: data

saldo: real

fimregistro

tipo vetor_de_registros: vetor [1..15] de registro_conta



```
procedimento inicializar_vetor (var v: vetor_de_registros)
var i: inteiro
inicio
```

```
  para i de 1 ate 15 faca
```

```
    escreva ("Entre com as informações do registro número ")
```

```
    escreval (i, ".")
```

```
    escreva ("Número da conta: ")
```

```
    leia (v[i].numero_conta)
```

```
    escreva ("Nome do cliente: ")
```

```
    leia (v[i].nome)
```

```
    escreva("CPF: ")
```

```
    leia (v[i].cpf)
```

```
    escreva("RG: ")
```

```
    leia (v[i].rg)
```

```
    escreval("Data de abertura: ")
```

```
    escreva ("Ano: ")
```

```
    leia (v[i].data_abertura.ano)
```

```
    escreva ("Mês: ")
```

```
    leia (v[i].data_abertura.mes)
```

```
    escreva ("Dia: ")
```

```
    leia (v[i].data_abertura.dia)
```

```
    escreva("Saldo: ")
```

```
    leia (v[i].saldo)
```

```
  fimpara
```

```
fimprocedimento
```

```
funcao imprimir_registro_CPF (var v: vetor_de_registros; cpf:
    caractere):logico
var i: inteiro
inicio
    para i de 1 ate 15 faca
        se (v[i].cpf=cpf) entao
            escreval ("Registro da conta número ", v[i].numero_conta)
            escreval ("Cliente: ",v[i].nome)
            escreval("CPF: ",v[i].cpf)
            escreval("RG: ",v[i].rg)
            escreva ("Data de abertura: ",v[i].data_abertura.dia,"/")
            escreval (v[i].data_abertura.mes, "/", v[i].data_abertura.ano)
            escreval("Saldo: ",v[i].saldo)
            retorne (verdadeiro)
        fimse
    fimpara
    retorne (falso)
fimfuncao
```

funcao imprimir_registro_posicao (var v: vetor_de_registros; posicao:
inteiro):logico

inicio

se (posicao<1 ou posicao>15) entao

retorne (falso)

senao

escreva ("Registro da conta número ")

escreval (v[posicao].numero_conta)

escreval ("Cliente: ",v[posicao].nome)

escreval("CPF: ",v[posicao].cpf)

escreval("RG: ",v[posicao].rg)

escreva("Data de abertura: ",v[posicao].data_abertura.dia,"/")

escreval ([posicao].data_abertura.mes,"/")

escreval (v[posicao].data_abertura.ano)

escreval("Saldo: ",v[posicao].saldo)

retorne (verdadeiro)

fimse

fimfuncao

```

procedimento principal()
var vet_reg: vetor_de_registros
    cpf: caractere
    p: inteiro
inicio
    inicializar_vetor(vet_reg)
    escreva ("Entre com um CPF para impressão do registro da ")
    escreval ("conta correspondente: ")
    leia (cpf)
    se (nao imprimir_registro_CPF (vet_reg, cpf)) entao
        escreva ("Não existe nenhum conta de cliente com o CPF ")
        escreval ("especificado.")
    fimse
    escreval ("Entre com a posição do registro da conta para ")
    escreval ("impressão, intervalo [1,15]: ")
    leia (p)
    se (nao imprimir_registro_posicao (vet_reg, p)) entao
        escreval ("Posição inválida.")
    fimse
fimprocedimento
inicio
    principal()
fimalgoritmo

```