

ALGORITMOS E ESTRUTURAS DE DADOS I

Prof. Caio César de Freitas Dantas

Estrutura de Dados

Definição:

- Uma estrutura de dados é um modo particular de armazenamento e organização de dados em um computador de modo que possam ser usados eficientemente.
- Vetores, Matrizes, Imagens e diversas outras composições de tipos de dados podem ser chamados de estruturas de dados.
- As Estruturas de Dados permitem melhor organização para o armazenamento e manipulação de dados.

Estrutura de Dados

- ÊNFASE ao estudo de DADOS:
 - Armazenamento;
 - Manipulação;
 - Refinamento;
 - Estrutura para representação.

Tipos Estruturados

As estruturas de dados estruturados possuem as seguintes características:

- Os tipos estruturados permitem fazer por meio de uma simples referência a manipulação de um conjunto de informações (um único identificador).

Existem dois tipos estruturados:

- Os homogêneos que são um conjunto de informações de um mesmo tipo, como os vetores em C/C++ os arrays unidimensional e multidimensional;

Tipos Estruturados

- Os dados eram armazenados através de variáveis simples do tipo char, int, float ou double.
- Estruturas de dados homogêneas
 - Como Vetores e Matrizes
 - Armazenam vários valores todos de um mesmo tipo.

Estrutura de Dados Homogêneas

Suponha que você necessite armazenar os dados de um livro. Certamente precisará armazenar o nome do livro (um conjunto de character), o preço (um valor real) e o número de páginas que possui (um número inteiro). Ao considerar que se deseje armazenar essas informações de três desses livros.

	Livro 1	Livro 2	Livro 3
Nome	nomeLiv1	nomeLiv2	nomeLiv3
Preço	precoLiv1	precoLiv2	precoLiv3
Paginas	pagLiv1	pagLiv3	pagLiv3

Estrutura de Dados Homogêneas

Uma outra abordagem seria a utilização de vetores.

ALGORITMO

```
nome    vetor[1..3] de caracter    // nome do livro
preco    vetor[1..3] de real        // preço do livro
paginas  vetor[1..3] de inteiro     // páginas do livro
i inteiro                                // declaração do índice
```

```
Escreva("Entre com os nomes, preços e números de páginas de três livros")
```

```
para i de 1 ate 3 faca                // leitura dos dados
```

```
    Leia(nome[i], preco[i], paginas[i])
```

```
fimpara
```

```
Escreva("Esses foram os dados digitados")
```

```
para i de 1 ate 3 faca
```

```
    Escreva(nome[i], preço[i], paginas[i])
```

```
fimpara
```

FINALGORITMO

Estrutura de Dados Homogêneas

- É uma abordagem que não deixa claro o fato de se estar lidando com um grupo de características relacionadas a uma única entidade - o livro.
- É evidente que com o programa se tornará cada vez mais difícil trabalhar com as estruturas e entender sua lógica conforme aumentar o número de características dos livros a serem armazenadas.

Estrutura de Dados Homogêneas

- Um programa que gerencie as contas de todos os clientes de um determinado banco. Consideremos que cada conta deve conter: o número da conta (identificador), o nome do cliente e o saldo.
- Poderíamos fazer o uso de uma variável simples para cada um destes elementos, implicaria, como são várias contas, a necessidade de vários vetores, ou seja um vetor para cada atributo.
- Porém essa estratégia atrapalharia a legibilidade e a manutenção do programa, além de dificultar a possibilidade de armazenamento dos dados em disco.

Tipos Estruturados

Existem dois tipos estruturados:

- Os homogêneos que são um conjunto de informações de um mesmo tipo, como os vetores em C/C++ os arrays unidimensional e multidimensional;
- Estruturas heterogêneas permitem armazenar em uma mesma variável diferentes tipos.
 - Elas são chamadas de registros:
 - Um registro pode ser composto de vários campos (componentes ou elementos) e cada um deles pode ser de um tipo diferente.

Estrutura de Dados Heterogêneas

- Por exemplo, uma ficha de cadastro tem como componentes: nome, telefone, endereço, cidade e cep.
- Portanto uma ficha de cadastro é um registro com os campos: nome, telefone, endereço, cidade e cep. Os registros serão declarados como tipos semelhantes a inteiro, real, etc.

Tipo Ficha = **registro**

Início

literal: nome

inteiro: telefone

literal: endereço, cidade

inteiro: cep

Fim

Estrutura de Dados Heterogêneas

São estruturas de dados heterogêneas e evidentemente compostas, que agregam estruturas de dados mais simples e com as seguintes características:

- É sempre composto por campos.
- Contém um número fixo e determinado de campos
- Cada campo funciona como uma variável mais simples e primitiva;
- Os campos formam estruturas chamadas de REGISTROS, que podem ser manipulados sem a especificação individual dos campos para algumas operações, ou quando necessário permite a manipulação de um determinado campo.
- Os registros podem ser agregados em estruturas mais complexas chamadas de vetores de registros (acessados por meio de listas ou filas) e arquivos.

Registros

A utilização dos registros:

- Facilita na arquitetura dos dados e na elaboração de algoritmos mais complexos que serão implementados em uma linguagem que suporte tais estruturas.
- Permite a associação de campos de diferentes tipos mais simples;
- Permite a associação de variáveis que possuam algum tipo de vínculo lógico;
- Facilita no armazenamento de dados em diferentes dispositivos de armazenamento;
- Facilita na implementação de estruturas dinâmicas em memória principal RAM;
- Facilita os processos e procedimentos para acessar dados/ informações complexas.

Registros

As informações de um aluno dentro da Universidade/Faculdade:

- Matrícula
- Nome
- Pai e Mãe
- CPF
- Numero Identidade
- Nascimento
- Naturalidade
- Endereço
- Instituição Ensino Médio
- Data de Matrícula
- Desempenho Acadêmico

Registros

Definição em C

```
struct nome-estrutura {  
    declaração dos membros  
} definição de variáveis (optional);
```

```
//declaração do tipo de dado  
struct livro  
{  
    char nome;  
    float preco;  
    int pagina;  
};
```

Registros

Declaração de variáveis

```
tipo Ficha = registro
    nome: string
    matricula: inteiro
    nota1: real
    nota2: real
```

Fimregistro

```
///Declaração de Variáveis
```

```
i, j : inteiro
```

```
aluno : ficha
```

```
//declaração do tipo de dado
```

```
struct livro
```

```
{
```

```
    char nome;
```

```
    float preco;
```

```
    int pagina;
```

```
};
```

```
//declaração das variáveis
```

```
struct livro livro1, livro2, livro3;
```


Registros

Em algum momento podemos precisar de todas as informações contidas em um registro, e outras apenas a informação contida em apenas um campo do registro.

Para acessar o valor de cada campo de uma variável Aluno:

`nome_da_variável.nome_campo`

Registros

```
tipo Ficha = registro
  nome: string
  matricula: inteiro
  nota1: real
  nota2: real
Fimregistro
```

```
///Declaração de Variáveis
i, j : inteiro
aluno : ficha
```

- Em algum momento podemos precisar de todas as informações contidas em um registro, e outras apenas a informação contida em apenas um campo do registro.

Manipulação Genérica

```
todo o registro:
leia(aluno)
escreva(aluno)
```

Manipulação Específica

```
apenas um campo:
leia(aluno.nota1)
escreva(aluno.nota1)
```

Registros

Para acessar o valor de cada campo de uma variável Aluno:

`nome_da_variável.nome_campo`

```
aluno.matricula = 1234;
```

```
aluno.nome = "Chico";
```

```
aluno.nota1 = "5.8";
```

```
aluno.nota2 = "8.2";
```

Registros

Os registros também podem conter tipos compostos homogêneos e heterogêneos.

```
tipo Ficha = registro
  nome: string
  matricula: inteiro
  notas: vetor[1..3] de real
fimregistro
```

```
///Declaração de Variáveis
i, j : inteiro
aluno : ficha
```

Registros

```
tipo Ficha = registro  
    nome: string  
    matricula: inteiro  
    notas: vetor[1..3] de real  
fimregistro
```

```
///Declaração de Variáveis  
i, j : inteiro  
aluno : ficha
```

Manipulação Específica

Estrutura condicional considerando apenas um campo:

```
se aluno.nota[3] >= 7,5 então  
    escreva(“Nota acima da média”)  
fimse
```

Manipulação Específica

Estrutura Vetorial Bidimensional:

```
se aluno.pagamento[1,11] = verdadeiro então  
    escreva(“Pagamento da 11º. Mensalidade OK”)  
fimse
```

Registros

Um vetor de registros.

```
tipo Ficha = registro
    nome: string
    matricula: inteiro
    notas: vetor[1..3] de real
fimregistro
```

```
///Declaração de Variáveis
i, j : inteiro
aluno : ficha
```

```
tipo Ficha = registro
    nome: string
    matricula: inteiro
    notas: vetor[1..3] de real
fimregistro
```

```
///Declaração de Variáveis
i, j : inteiro
cadastro : vetor [1..50] de ficha
```

Registros

Os registros também podem conter registros.

```
tipo Ficha = registro
    nome: string
    matricula: inteiro
    notas: vetor[1..3] de real
    endereco: string
fimregistro

///Declaração de Variáveis
i, j : inteiro
aluno : ficha
```

Registros

Os registros também podem conter registros.

```
tipo Ficha = registro
    nome: string
    matricula: inteiro
    notas: vetor[1..3] de real
    endereco: string
fimregistro
```

```
///Declaração de Variáveis
i, j : inteiro
aluno : ficha
```

```
tipo ENDER = registro
    Rua: string
    Num: inteiro
    Bairro: string
fimregistro
```


Registros

Os registros também podem conter registros.

```
tipo Ficha = registro
  nome: string
  matricula: inteiro
  notas: vetor[1..3] de real
  endereco: ENDER
fimregistro
```

```
///Declaração de Variáveis
i, j : inteiro
aluno : ficha
```

```
tipo ENDER = registro
  Rua: string
  Num: inteiro
  Bairro: string
fimregistro
```

```
Ex: Escreva(“O aluno mora na Rua: ”, aluno.endereco.rua);
```

Registros – Exemplos 1/5

O Tribunal Eleitoral necessita controlar os dados dos candidatos da última eleição.
Para isso...

- Defina um tipo registro para o cadastro de candidatos a uma eleição composto dos seguintes campos: nome, endereço, partido, cargo e número de votos.

Registros – Exemplos 1/5

O Tribunal Eleitoral necessita controlar os dados dos candidatos da última eleição.
Para isso...

- Defina um tipo registro para o cadastro de candidatos a uma eleição composto dos seguintes campos: nome, endereço, partido, cargo e número de votos.

Tipo CadastroEleicao = registro

nome: string

endereço: string

partido: string

cargo: string

votos: inteiro

fimregistro

Registros – Exemplos 2/5

Faça um algoritmo que use uma variável do tipo registro criado e leia dados de um candidato para esta variável e os escreva.

```
Tipo CadastroEleicao = registro
    nome: string
    endereco: string
    partido: string
    cargo: string
    votos: inteiro
Fimregistro
```

Registros – Exemplos 2/5

Faça um algoritmo que use uma variável do tipo registro criado e leia dados de um candidato para esta variável e os escreva.

Tipo CadastroEleicao = registro

nome: string

endereco: string

partido: string

cargo: string

votos: inteiro

Fimregistro

///Declaração de Variáveis

candidato : CadastroEleicao

Registros – Exemplos 2/5

Faça um algoritmo que use uma variável do tipo registro criado e leia dados de um candidato para esta variável e os escreva.

Início

Leia(candidato.nome)

Leia(candidato.endereco)

Leia(candidato.partido)

Leia(candidato.cargo)

Leia(candidato.votos)

Escreva(candidato.nome)

Escreva(candidato.endereco)

Escreva(candidato.partido)

Escreva(candidato.cargo)

Escreva(candidato.votos)

Fim

Tipo CadastroEleicao = registro

nome: string

endereco: string

partido: string

cargo: string

votos: inteiro

Fimregistro

///Declaração de Variáveis
candidato : CadastroEleicao

Registros – Exemplos 3/5

Agora modifique o tipo registro para incluir o campo data da eleição que é um tipo registro composto dos campos dia, mês e ano.

Tipo CadastroEleicao = registro

nome: string

endereco: string

partido: string

cargo: string

votos: inteiro

Fimregistro

Registros – Exemplos 3/5

Agora modifique o tipo registro para incluir o campo data da eleição que é um tipo registro composto dos campos dia, mês e ano.

```
Tipo CadastroEleicao = registro
    nome: string
    endereco: string
    partido: string
    cargo: string
    votos: inteiro
    dataEleicao: ? ? ?
Fimregistro
```


Registros – Exemplos 3/5

Agora modifique o tipo registro para incluir o campo data da eleição que é um tipo registro composto dos campos dia, mês e ano.

```
Tipo data = registro
    dia: inteiro
    mes: inteiro
    ano: inteiro
Fimregistro
```

```
Tipo CadastroEleicao = registro
    nome: string
    endereco: string
    partido: string
    cargo: string
    votos: inteiro
    dataEleicao: data
Fimregistro
```

Registros – Exemplos 4/5

Faça um algoritmo que use uma variável do tipo registro criado e leia e escreva os dados relativos ao dia da eleição.

```
Tipo CadastroEleicao = registro
    nome: string
    endereco: string
    partido: string
    cargo: string
    votos: inteiro
    dataEleicao: data
Fimregistro
```

```
Tipo data = registro
    dia: inteiro
    mes: inteiro
    ano: inteiro
Fimregistro
```

Registros – Exemplos 4/5

Faça um algoritmo que use uma variável do tipo registro criado e leia e escreva os dados relativos ao dia da eleição.

```
///Declaração de Variáveis  
eleicao : CadastroEleicao
```

```
Tipo CadastroEleicao = registro  
    nome: string  
    endereco: string  
    partido: string  
    cargo: string  
    votos: inteiro  
    dataEleicao: data  
Fimregistro
```

```
Tipo data = registro  
    dia: inteiro  
    mes: inteiro  
    ano: inteiro  
Fimregistro
```

Registros – Exemplos 4/5

Faça um algoritmo que use uma variável do tipo registro criado e leia e escreva os dados relativos ao dia da eleição.

```
///Declaração de Variáveis  
eleicao : CadastroEleicao
```

```
Início  
Leia(eleicao.dataEleicao.dia)  
Leia(eleicao.dataEleicao.mes)  
Leia(eleicao.dataEleicao.ano)  
Escreva(eleicao.dataEleicao.dia)  
Escreva(eleicao.dataEleicao.mes)  
Escreva(eleicao.dataEleicao.ano)  
Fim
```

```
Tipo CadastroEleicao = registro  
    nome: string  
    endereco: string  
    partido: string  
    cargo: string  
    votos: inteiro  
    dataEleicao: data  
Fimregistro
```

```
Tipo data = registro  
    dia: inteiro  
    mes: inteiro  
    ano: inteiro  
Fimregistro
```

Registros – Exemplos 4/5

Faça um algoritmo que use uma variável do tipo registro criado e leia e escreva os dados relativos ao dia da eleição.

```
///Declaração de Variáveis  
eleicao : data
```

```
Início  
Leia(eleicao.dia)  
Leia(eleicao.mes)  
Leia(eleicao.ano)  
Escreva(eleicao.dia)  
Escreva(eleicao.mes)  
Escreva(eleicao.ano)  
Fim
```

```
Tipo CadastroEleicao = registro  
    nome: string  
    endereco: string  
    partido: string  
    cargo: string  
    votos: inteiro  
    dataEleicao: data  
Fimregistro
```

```
Tipo data = registro  
    dia: inteiro  
    mes: inteiro  
    ano: inteiro  
Fimregistro
```

Registros – Exemplos 5/5

E se quiséssemos guardar dados de 200 candidatos? Que estrutura de dados poderíamos usar?

```
Tipo CadastroEleicao = registro
    nome: string
    endereco: string
    partido: string
    cargo: string
    votos: inteiro
    dataEleicao: data
Fimregistro
```

```
Tipo data = registro
    dia: inteiro
    mes: inteiro
    ano: inteiro
Fimregistro
```

Registros – Exemplos 5/5

E se quiséssemos guardar dados de 200 candidatos? Que estrutura de dados poderíamos usar?

///Declaração de Variáveis

i : inteiro

Varioscand : vetor [1..200] de CadastroEleicao

Tipo CadastroEleicao = registro

nome: string

endereco: string

partido: string

cargo: string

votos: inteiro

dataEleicao: data

Fimregistro

Tipo data = registro

dia: inteiro

mes: inteiro

ano: inteiro

Fimregistro

Registros – Exemplos 5/5

E se quiséssemos guardar dados de 200 candidatos? Que estrutura de dados poderíamos usar?

///Declaração de Variáveis

i : inteiro

Varioscand : vetor [1..200] de CadastroEleicao

Início

Para i de 0 até 199

 Leia(Varioscand[i].nome, Varioscand[i].endereco, Varioscand[i].partido,
Varioscand[i].cargo, Varioscand[i].votos, Varioscand[i].dataeleicao.dia,
Varioscand[i].dataeleicao.mes, Varioscand[i].dataeleicao.ano)

Escreva(Varioscand[i].nome, Varioscand[i].endereco, Varioscand[i].partido,
Varioscand[i].cargo, Varioscand[i].votos, Varioscand[i].dataeleicao.dia,
Varioscand[i].dataeleicao.mes, Varioscand[i].dataeleicao.ano)

Fimpara

Fim

Tipo CadastroEleicao = registro

 nome: string

 endereco: string

 partido: string

 cargo: string

 votos: inteiro

 dataEleicao: data

Fimregistro

Tipo data = registro

 dia: inteiro

 mes: inteiro

 ano: inteiro

Fimregistro

FIM!
