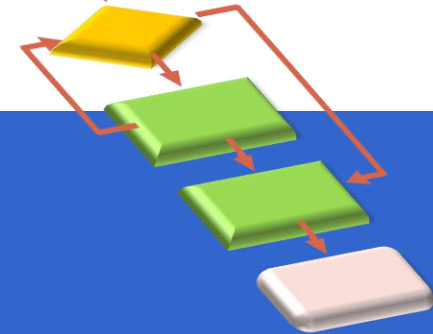




Edited by Foxit Reader
Copyright(C) by Foxit Corporation,2005-2009
For Evaluation Only.



Algoritmos I

Dados Estruturados Heterogêneos: Registros

Prof.^a DSc. Vanessa de Oliveira Campos

Os campos não necessariamente têm o mesmo tipo

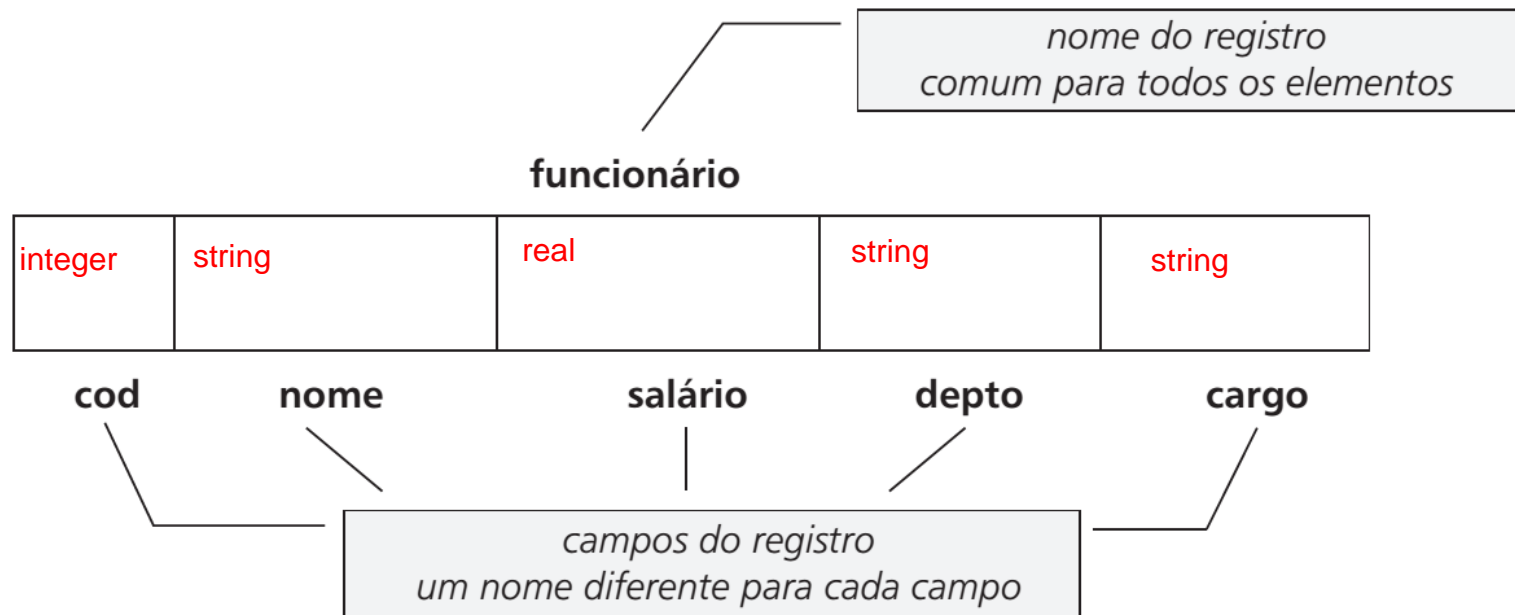
Introdução

- **Registros** são *estruturas heterogêneas*, ou seja, *coleções de dados* de quaisquer tipos.
- Os dados que integram um registro são denominados **campos**.
- Os registros e seus campos devem ser nomeados com nomes que sejam únicos.



Introdução

- Campos do registro de um funcionário:



Podemos encontrar qualquer tipo de registro: inteiro real, string, char, array e até o próprio registro



Declaração de Registro

- Na declaração de um tipo registro devem ser definidos os nomes dos campos que irão compor o registro
- Para cada campo deve também ser definido o tipo de dado que nele será armazenado.
- Sintaxe:

```
registro  
    <nome do campo 1> : <tipo do campo 1>  
    <nome do campo 2> : <tipo do campo 2>  
    ...  
    <nome do campo n> : <tipo do campo n>  
fimregistro
```



Declaração de Registro

- Exemplo:

```
var  
    um_funcionario: registro  
        cod      : inteiro  
        nome     : string  
        salario  : real  
        depto    : inteiro  
        cargo    : caractere  
fimregistro
```



Declaração de Registro

- A declaração da variável `um_funcionario`, feita em dois passos, primeiro o tipo e após a variável:

Tipo

```
tfuncionario = registro
```

```
    cod    : inteiro
```

```
    nome   : string
```

```
    salario : real
```

```
    depto  : inteiro
```

```
    cargo  : caractere
```

```
fimregistro
```

var

```
um_funcionario: tfuncionario
```

*Podemos criar um tipo personalizado

*Aqui criamos um tipo `tfuncionario`

*Aqui declaamos uma variável `um_funcionario` do tipo `tfuncionario`



Registro

```
var
```

```
funcionarios: vetor [1..6] de tfuncionario
```



Declaração de Registro

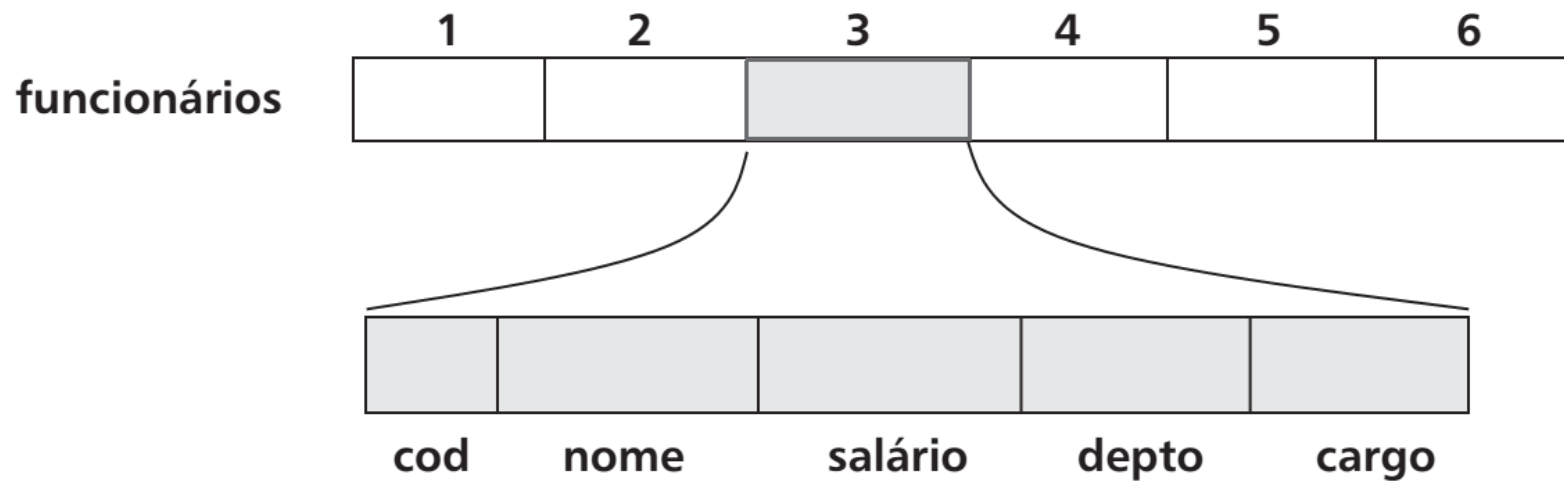
Tipo

```
tfuncionario = registro  
    cod   : inteiro  
    nome  : string  
    salario : vetor [1..4] de real  
    depto: inteiro  
    cargo: caractere  
fimregistro
```



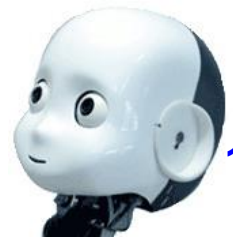
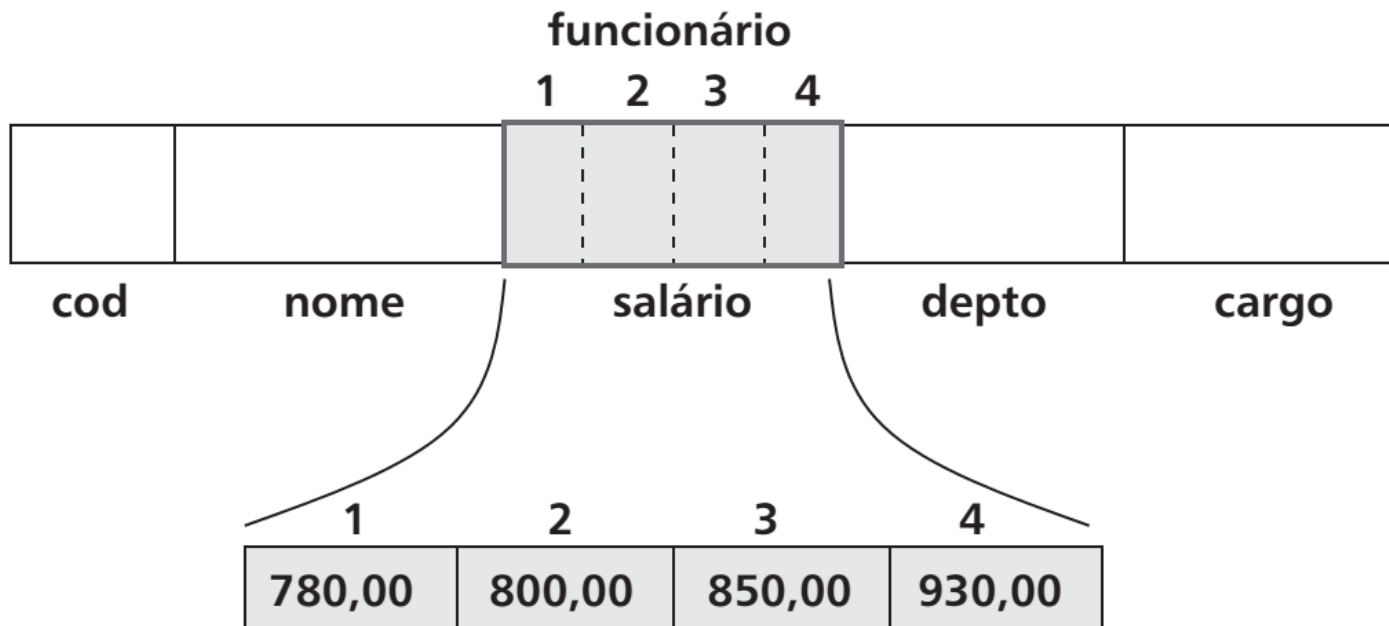
Registro

- Arranjo de Registro



Registro

- Arranjo como campo de um registro



Referência a Elementos de Registros

- A referência a um campo de um registro é feita indicando o nome do registro seguido do nome do campo, **separados por um ponto**:

`<nome do registro>.<nome do campo>`



Referência a Elementos de Registros

- Exemplo:

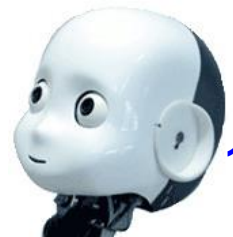
```
{PREENCHER POR LEITURA O CAMPO NOME}  
leia(um_funcionario.nome)  
{ATRIBUIR UM VALOR AO 2o ELEMENTO DO CAMPO  
SALÁRIO, QUE É UM VETOR}  
um_funcionario.salario[2] ← 800  
{TESTAR O VALOR DO CAMPO CARGO}  
se um_funcionario.cargo = 'A' entao  
    um_funcionario.depto ← 10 {ATRIBUI VALOR AO  
    CAMPO DEPTO}
```



Referência a Elementos de Registros

- A referência a um elemento deve ser feita indicando todo o caminho necessário para acessar esse elemento.
- Exemplo:

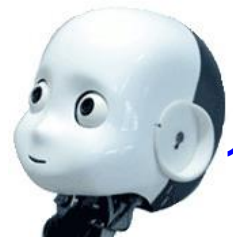
```
funcionarios[3].salario[2].valor
```



Registros em Pascal

- Sintaxe:

```
record  
    <nome do campo 1> : <tipo do campo 1> ;  
    <nome do campo 2> : <tipo do campo 2> ;  
    ...  
    <nome do campo n> : <tipo do campo n>  
end
```

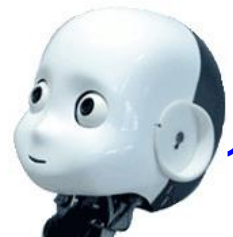


Registros em Pascal

- A declaração do tipo funcionário :

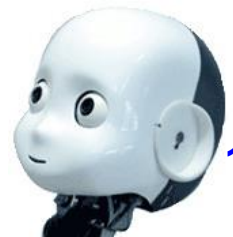
Type

```
    tfuncionario = record  
        cod      : integer;  
        nome     : string;  
        salario  : real;  
        depto    : integer;  
        cargo    : char;  
    end;
```



Registros em Pascal

- Não existe limitação para o número de campos que um registro pode ter.
- Qualquer tipo de dado pode ser utilizado na definição de um campo.



Registros em Pascal

■ Exemplo:

Type

```
tcliente = record  
    nome : string;  
    endereco : record  
        rua : string;  
        numero : integer;  
        complemento : integer;  
        cidade : string  
    end; {FIM REGISTRO CAMPO ENDERECO}  
end; {FIM REGISTRO TIPO_CLIENTE}
```

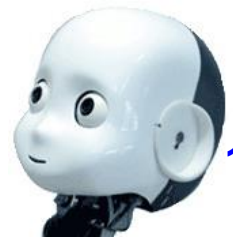
*Registro dentro do registro



Referência a Campo de Registro

- A referência a um campo de um registro é feita indicando o nome do registro seguido do nome do campo, separados por um ponto:

`<nome do registro>.<nome do campo>`



Referência a Campo de Registro

- Exemplo:

```
var
    cliente: tipo_cliente;
begin
    readln(cliente.nome);
    readln(cliente.endereco.rua);
    readln(cliente.endereco.numero);
    ...
```



Referência a Campo de Registro

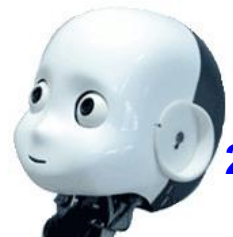
- Exemplo:

```
type
  T = record
    valor : integer;
    vetor : array [1..10] of real;
    refer : char;
  end;
var
  indice : integer;
  caractere : char;
  x: T;
```



Referência a Campo de Registro

```
begin
  readln(x.valor); {PREENCHE PRIMEIRO CAMPO}
  For indice := 1 to 10 do {PREENCHE SEGUNDO CAMPO}
    readln(x.vetor[indice]);
  Repeat {PREENCHE TERCEIRO CAMPO}
    readln (x.refer);
  until (x.refer = 'a') or (x.refer = 'b') or
    (x.refer = 'c')
    ...
end.
```



Atribuição de Registros Inteiros

- É possível fazer a atribuição integral de uma variável registro para uma outra variável registro do mesmo tipo.

```
x : tfuncionario  
func : funcionario  
y : array [1..10] of tfuncionario
```

```
Begin  
x := func;  
y[1] := func;
```



Atribuição de Registros Inteiros

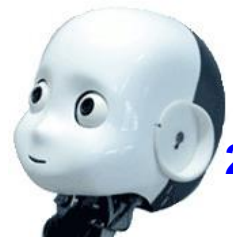
```
type
    tipo_reg = record
        nome : string;
        valor : integer;
    end;
var
    reg1, reg2: tipo_reg;
begin
    readln(reg1.nome);
    readln(reg1.valor);

    {ATRIBUICAO DE UM REGISTRO INTEIRO A OUTRO}
    reg2 := reg1;
    ...
```



Exercícios

1. Escrever um programa que cadastre o nome, a altura, o peso, o cpf e sexo de 10 pessoas. Crie um registro para cadastrar essas informações. Faça o cadastro de todas as pessoas. Em seguida, localizar uma pessoas através do seu CPF e imprimir o seu IMC.



Exercícios

2. Latitude e longitude são especificados em graus (°), minutos ('), segundos ("), e direção (N, S, L, O). Por exemplo, a cidade A fica na latitude 22° 17' 34" N e longitude 53° 41' 9" O. Defina um tipo em Pascal cujo nome seja *Localizacao*, e que seja constituído de longitude e latitude, ambas do tipo coordenadas. Para isto, você terá que definir o tipo *coordenadas* como sendo constituído de grau, minutos, segundos e direção. Leia e exiba uma localização no seu programa.



Exercícios

3. A prefeitura de uma cidade fez uma pesquisa entre seus habitantes, coletando dados sobre salário, idade e número de filhos. Faça um programa que leia esses dados de 20 pessoas, calcule e mostre:
- a média de salário da população;
 - a média do número de filhos;
 - o maior salário;
 - o percentual de mulheres com salário superior a R\$1000,00;



