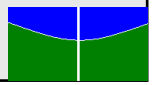
COMPUTAÇÃO BÁSICA

Disciplina: 116301

Profa. Carla Denise Castanho

Universidade de Brasília – UnB
Instituto de Ciências Exatas – IE
Departamento de Ciência da Computação – CIC



14. REGISTROS

Registros

- Registros são estruturas que podem agregar diferentes informações. Dessa maneira pode-se fazer diferentes combinações gerando **novos tipos de dados.**
- Um registro é uma coleção de campos, em que cada campo pode ser de um tipo de dado diferente. Por isso, os registros são conhecidos como **variáveis compostas heterogêneas.**

Registros

- Exemplo:

Dados de Funcionários					
Código:	9182	**Nome:**	Hermenegildo Florentil	**Sexo:**	Masculino
Endereço:	Rua dos Registros Isolados, 736				
Cargo:	Chefe de Divisória		**Salario:**	$ 455,46	

- Podemos ver que as variaveis Nome, Sexo, Endereço, Cargo são Literais, o Código é um inteiro, e o Salario é do tipo real. Todas essas variaveis e seus tipos, criam um novo tipo de dado: DADOS DE FUNCIONÁRIOS

- Podemos analisar que da mesma forma que existe os tipos, literal, inteiro, real, agora existe o tipo Dados de Funcionários, e variáveis podem ter esse tipo.

Registros

- Em pseudo código há duas formas de declarar um registro.
- Temos:

Exemplo de declaração de um registro em pseudocódigo.

```
Algoritmo FuncionariosDaEmpresa

Variáveis
    i  : inteiro
    funcionarioDaEmpresa : registro( nome, sexo, endereço, cargo : literal ;
          codigo : inteiro ; salario : real)

Início
          ....
Fim
```


Registros

- Ou podemos declarar antes do campo das variaveis um tipo novo de dado, o registro

Algoritmo FuncionariosDaEmpresa

Definições

tipoFuncionarioDaEmpresa : **registro**(nome, sexo, endereço, cargo : **literal** ; codigo : **inteiro** ; salario : **real**)

Variaveis

funcionarioDaEmpresa : tipoFuncionarioDaEmpresa

variável **tipo**

Inicio

....

Fim

Registros

- Em C temos registro com o nome de STRUCT:

```
#include <stdio.h>

typedef  struct {
      int codigo;
      float salario;
      char nome[50], sexo[10], endereco[50], cargo[50];
} tipoDadosDeFuncionario;

int main() {
      tipoDadosDeFuncionario dadosDeFuncionario;
      ...
}
```

tipo

variável

Registros

- Em C podemos também declarar:

```
#include <stdio.h>

struct  dadosDeFuncionario{
        int codigo;
        float salario;
        char nome[50], sexo[10], endereco[50], cargo[50]
};
/*Neste caso o novo tipo de dado se chama struct
dadosDeFuncionario*/

int main() {
        struct dadosDeFuncionario dadosDeFuncionario;
        ...
}
```

variável

Registros

- Em C podemos também declarar:

```
#include <stdio.h>

struct  dadosDeFuncionario{
        int codigo;
        float salario;
        char nome[50], sexo[10], endereco[50], cargo[50]
} dadosFunc;

 /* Neste caso, dadosFunc é variavel global, e não devemos utilizar
variáveis globais!*/

int main() {
        dadosFunc.codigo = 9182;
        ...
}
```

Registros

- Para acessarmos o conteúdo do registro basta colocarmos o nome_da_variavel_do_registro.nome_do_variavel_interna

Algoritmo FuncionariosDaEmpresa

Definições
tipoFuncionarioDaEmpresa : **registro**(nome, sexo, endereço, cargo : **literal** ; codigo : **inteiro** ; salario : **real**)

Variaveis
funcionarioDaEmpresa : tipoFuncionarioDaEmpresa
Inicio
funcionarioDaEmpresa.nome ← ' Hermenegildo Florentil'
funcionarioDaEmpresa.codigo ← 9182
funcionarioDaEmpresa.sexo ← 'masculino'
...
Fim

Registros

- Em C

```
#include <stdio.h>

typedef  struct {
        int codigo;
        float salario;
        char nome[50], sexo[10], endereco[50], cargo[50];
} tipoDadosDeFuncionario;

int main() {
        tipoDadosDeFuncionario dadosDeFuncionario;

        dadosDeFuncionario.codigo = 9182;
        dadosDeFuncionario.salario = 455.46;
        ...
}
```

tipo

variável

Registros

- Chegamos a conclusão que podemos trabalhar com os registros da mesma forma que trabalhavamos com as variáveis, todavia com a versatilidade de conseguirmos agrupar varios dados em um dado maior! Com isso podemos da mesma forma criar um vetor de registros, para uma determinada empresa! Facilitando muito a forma de armazenar os dados e buscá-los.

Registros

- Exemplo de leitura de dados de funcionários de uma empresa:

```
Algoritmo FuncionariosDaEmpresa

Definições
    tipoFuncionarioDaEmpresa : registro (nome, sexo, endereço,
cargo : literal ; codigo : inteiro ; salario : real)

Variaveis
    funcionarioDaEmpresa[100] : tipoFuncionarioDaEmpresa
    i : inteiro
Inicio
    para i←0 até 99 faça
            leia funcionarioDaEmpresa[i].nome
            leia funcionarioDaEmpresa[i].sexo
            leia funcionarioDaEmpresa[i].endereço
    ....
Fim
```

Registros

- Exemplo de leitura de dados de funcionarios de uma empresa em C:

```
#include <stdio.h>

typedef  struct {
        int codigo;
        float salario;
        char nome[50], sexo[10], endereco[50], cargo[50]
} tipoDadosDeFuncionario;

int main() {
        tipoDadosDeFuncionario dadosDeFuncionario[100];
        int i;
        for (i = 0; i<100; i++ ) {
                scanf("%s", dadosDeFuncionario[i].nome);
                scanf("%d", &dadosDeFuncionario[i].codigo);
                scanf("%s", dadosDeFuncionario[i].sexo);
        ...
}
```

Registros

- Cuidado, é muito comum utilizar ponteiros que apontam para registros

```
#include <stdio.h>

typedef  struct {
        int dia;
        int mes;
        int ano
} tipoData;

int main() {
        tipoData *p; /* define ponteiro p para registros data */
        tipoData x;
        p = &x; /* agora p aponta para x */
        (*p).dia = 31; /* mesmo efeito que x.dia = 31 */
        (*p).mes = 8;
        (*p).ano = 1998;
....
}
```


Registros

- O que é semelhante a utilizar:

```
#include <stdio.h>

typedef  struct {
        int dia;
        int mes;
        int ano
} tipoData;

int main() {
        tipoData *p; /* define ponteiro p para registros data */
        tipoData x;
        p = &x; /* agora p aponta para x */
        p->dia = 31; /* mesmo efeito que x.dia = 31 */
        p->mes = 8;
        p->ano = 1998;
....
}
```

Registros

- Podemos também criar registros de registros, por exemplo:

```
#include <stdio.h>

typedef  struct {
        int dia;
        int mes;
        int ano
} tipoData;
        /*o registro tipoData, é muito utilizado quando temos que armazenar datas*/
typedef  struct {
        tipoData  data;
        char nome[50];
        char cpf[13];
}tipoVisitanteDaEmpresa;

int main() {
        tipoVisitanteDaEmpresa visitante;

        visitante.data.dia = 29;
        visitante.data.mes = 11;
        visitante.data.ano = 2007;
        visitante.cpf = '111111111-11';
....
}
```

Registros

- Exercicio em sala:

  Temos um empresa que começou a ser informatizada, queremos cadastrar no máximo 100 funcionários com as seguintes informações de cada funcionário: nome, sexo, salário, codigo (matricula), endereço, cargo.

  Queremos também saber o salario médio dos funcionarios da empresa, o salario médio dos homens, e o salario médio das mulheres da empresa.

  Faça um algoritmo para o problema acima utilizando vetor de registros. Leia o número de funcionários (n), que será menor ou igual a 100.