27. Registros

DIM0321

2015.1

Outline

Tipos de variáveis

Básicas

- `int`
- `float`
- `char`

Compostas

Arranjos estruturas **homogêneas** (conjunto do mesmo tipo dado indexado)

Registros estrutures **heterogêneas** (conjunto de tipos diferentes, **nomeados**)

Registro

Podem conter qualquer tipo de dados

- Tipos básicos / primitivos
- Ponteiros / arranjos
- **Registros**

Cada elemento — chamado de **campo** — tem um nome associado.

Elementos sintáticos

- Uso da palavra-chave `struct`
- Separação dos campos por *';'*

Declaração / inicialização

- É a declaração de um **novo tipo de dados** `struct <nome>`
- A inicialização tem a mesma sintaxe que a dos arranjos.

Acesso

- `<nome>.<campo>`

Exemplo introdutivo

```c
#include <stdio.h>

int main(void)
{
    struct produto {
        int codigo;
        char descricao[50];
        float valor;
        int estoque;
    };

    struct produto q;

    struct {
        int foo;
    } bar;

    struct produto p;

    // Inicialização no ponto de declaração
    struct produto pc = { 104, "caneta", 3.25, 1598 };
    pc.codigo = 104, pc.valor = 3.25;
    // Leitura de campos
    scanf("%d %s %f %d", &p.codigo, &p.descricao, &p.valor, &p.estoque);
    // Atribuição
    bar.foo = 42;
    /* Acesso */
    printf("%d", pc.codigo);
}
```


Acesso, inicialização, declaração

```c
#include <stdio.h>

int main(void)
{
    struct { /* nome do registro omitido */
        int codigo;
        char descricao[50];
        float valor;
        int estoque;
    } produto = { 104, "caneta", 3.25, 1598 };

    printf("descricao      = %s\n",   produto.descricao);
    printf("estoque        = %i\n",   produto.estoque);
    printf("valor unitario = %.2f\n", produto.valor);
    printf("valor total    = %.2f\n", produto.estoque * produto.valor);
    return 0;
}
```

Registros e arranjos

```c
#include <stdio.h>
int main (void)
{
  struct produto {
    int codigo;
    char descricao[50];
    float valor;
    int estoque;
  };
  struct produto produto1 = { 104, "caneta", 3.25, 1598 };
  struct produto produto2 = { 105, "lapis", 2., 2365 };
  printf("Produto 1:\n");
  printf("descricao   = %s\n", produto1.descricao);
  printf("estoque     = %i\n", produto1.estoque);
  printf("valor total = %.2f\n", produto1.estoque * produto1.valor);
  printf("Produto 2:\n");
  printf("descricao   = %s\n", produto2.descricao);
  printf("estoque     = %i\n", produto2.estoque);
  printf("valor total = %.2f\n", produto2.estoque * produto2.valor);
  struct produto produtos[3] = { {104, "caneta", 3.25, 1598},
                                 {105, "lapis", 2., 2365},
                                 {106, "caderno", 10.85, 127} };
  for (int i = 0; i < 3; ++i) {
       printf("Produto %i:\n", i + 1);
       printf("descricao   = %s\n", produtos[i].descricao);
       printf("estoque     = %i\n", produtos[i].estoque);
       printf("valor total = %.2f\n", produtos[i].estoque * produtos[i].valor);
  }
  return 0;
}
```


Definição de tipos: typedef

Uso

A palavra reservada typedef permite que um determinado tipo possa ser denominado de forma diferente de acordo com as necessidades do usuário.

Exemplo

```c
int main(void)
{
    typedef int meu_inteiro;
    meu_inteiro n = 10;
    n = n * 5;

    typedef struct {
         char nome[50];
         char sexo;
         char estado_civil;
         int idade;
    } pessoa; /* não pode criar variáveis */

    pessoa homem, mulher; /* struct eliminado */
}
```


Registros de registros com typedef

```c
typedef struct {
     int dia;
     char mes[3 + 1];  /* 1 caractere reservado para o '\0' */
     int ano;
} data;

typedef struct pessoa {
     char nome[50];
     char sexo;
     char estado_civil;
     data data_nascimento;  /* registro definido acima */
     struct pessoa  *pai;
} pessoa_t, person, *pessoa_p;


pessoa_t a;
```

Referências

Precisões

[KR88] 6

[Bac13] 8.1

- André Backes, *Linguagem C completa e descomplicada*, Elsevier, 2013.
- Brian W. Kernighan and Dennis M. Ritchie, *The (ANSI) C Programming Language*, 2nd ed., Prentice Hall Professional Technical Reference, 1988.

Perguntas ?

http://dimap.ufrn.br/~richard/dim0321