

Lógica de programação

Registro

Walisson Pereira

walisson_pereira@uvanet.br

Universidade Estadual Vale do Acaraú

Introdução

Registros em C

Exercícios

Referências

Introdução

Até este ponto do curso, as variáveis utilizadas nos programas conseguem armazenar apenas um tipo de dado, definido no momento em que foram declaradas.

Além de facilitar o uso das estruturas de controle de um programa, **as variáveis servem para representar entidades identificadas no problema real** que será resolvido computacionalmente.

Nesses casos, é bastante comum a necessidade de armazenar, dentro de uma mesma variável, diferentes tipos de dados.

É aí que entra em cena os **registros**.

Registros conseguem agregar vários dados acerca de uma mesma entidade. Com isso, os programadores podem gerar novos tipos de dados, não se limitando apenas à utilização dos tipos de dados primitivos fornecidos pelas linguagens de programação.

Cada dado contido em um registro é chamado de **campo**.

Os **campos** podem ser de diferentes tipos primitivos, ou ainda, podem representar outros registros.

É por essa razão que os **registros** são conhecidos, também, como **variáveis compostas heterogêneas**.

Registros em C

Na linguagem C, chamamos um registro de estrutura.

Uma **estrutura** (ou registro) é uma coleção de variáveis, que podem ou não ser de tipos diferentes, colocadas sob um único nome para manipulá-las.

As estruturas ajudam na organização do código e facilitam a vida do programador, pois juntam variáveis relacionadas e permitem que elas sejam tratadas como uma unidade maior.

As variáveis de uma estrutura são frequentemente chamadas de campos ou atributos da estrutura.

Exemplo: uma estrutura é uma “caixa” onde podem ser agrupados diversos dados correlacionados. A figura 1 ilustra uma estrutura que representa um estudante, chamada **t_estudante**. Note que a estrutura **estudante** ocupa um espaço equivalente ao espaço necessário para guardar todos os seus atributos.

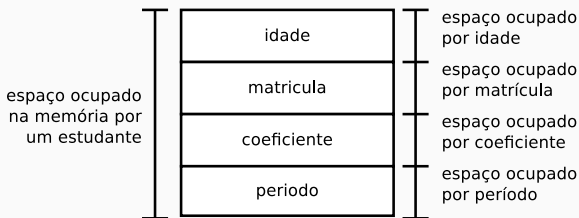


Figura 1: Estrutura **t_estudante**

Sintaxe:

```
1 typedef struct {  
2     <tipo do atributo 1> <nome do atributo 1>;  
3     <tipo do atributo 2> <nome do atributo 2>;  
4     ...  
5     <tipo do atributo n> <nome do atributo n>;  
6 } <nome da estrutura>
```

Exemplo: Sintaxe do tipo estrutura **t_estudante**

```
1 typedef struct {  
2     int idade;  
3     int matricula;  
4     float coeficiente;  
5     int periodo;  
6 } t_estudante;
```

Uso: Existem duas formas básicas de acessar as variáveis do tipo estrutura: selecionando um único elemento do dado (**modo seletivo**) ou manipulando toda a estrutura (**modo integral**).

Modo seletivo: Para manipular cada um dos atributos da estrutura, utiliza-se o mecanismo de seleção conforme a sintaxe apresentada a seguir:

```
1 <nome da variavel>.<nome do atributo>
```

Exemplo:

```
1 #include <stdio.h>
2 typedef struct {
3     int idade; int matricula;
4     float coeficiente; int periodo;
5 } t_estudante;
6 int main () {
7     t_estudante aluno;
8     scanf ("%d %d %f %d", &aluno.idade, &aluno.matricula, &aluno.
9         coeficiente, &aluno.periodo);
10    printf("idade = %d\n", aluno.idade);
11    printf("matricula = %d\n", aluno.matricula);
12    printf("coeficiente = %.2f\n", aluno.coeficiente);
13    printf("periodo = %d\n", aluno.periodo);
14    return 0;
15 }
```

Modo integral: Quando se deseja manipular a estrutura como um todo, utiliza-se simplesmente o nome da variável para acessá-la. O uso integral em atribuições é descrito pela sintaxi a seguir:

```
1 <nome da variavel 1> = <nome da variavel 2>;
```


Exemplo:

```
1 #include <stdio.h>
2 typedef struct {
3     int idade; int matricula;
4     float coeficiente; int periodo;
5 } t_estudante;
6 int main () {
7     t_estudante aluno, outro;
8     scanf ("%d %d %f %d", &aluno.idade, &aluno.matricula, &aluno.
9         coeficiente, &aluno.periodo);
10    outro = aluno;
11    printf("idade = %d\n", outro.idade);
12    printf("matricula = %d\n", outro.matricula);
13    printf("coeficiente = %.2f\n", outro.coeficiente);
14    printf("periodo = %d\n", outro.periodo);
15    return 0;
16 }
```

Exemplo: representação de uma conta corrente

```
1 typedef struct {  
2     int numero;  
3     char nome[30];  
4     float saldo;  
5 } t_conta_corrente;
```

Exemplo:

```
1  #include <stdio.h>
2  typedef struct {
3      int numero;
4      char nome[30];
5      float saldo;
6  } t_conta_corrente;
7  int main () {
8      t_conta_corrente conta;
9      scanf("%d", &conta.numero);
10     scanf("%s", conta.nome);
11     scanf("%f", &conta.saldo);
12     printf("A conta %d pertence a %s e tem saldo R$ %.2f\n", conta.
        numero, conta.nome, conta.saldo);
13     return 0;
14 }
```

Exemplo:

```
1  #include <stdio.h>
2  typedef struct {
3      char nome[30];
4      float salario;
5  } t_funcionario;
6  int main () {
7      t_funcionario funcionario[4];
8      int i;
9      for (i = 0; i < 4; i++) {
10         scanf("%s %f", funcionario[i].nome, &funcionario[i].salario);
11     }
12     printf("Lista de funcionarios:\n");
13     for (i = 0; i < 4; i++) {
14         printf("%s ganha R$ %.2f\n", funcionario[i].nome, funcionario
15             [i].salario);
16     }
```

Exercícios

- 1) Faça um programa que realize o cadastro de contas bancárias com as seguintes informações: número da conta, nome do cliente e saldo. O banco permitirá o cadastramento de apenas 10 contas e não poderá haver mais que uma conta com o mesmo número. Crie o menu de opções a seguir.

Menu de opções:

1. Cadastrar contas.
2. Visualizar todas as contas de determinado cliente.
3. Excluir a conta com menor saldo (supondo a não existência de saldos iguais).
4. Sair.

Exercícios

- 2) Uma empresa contratou 10 funcionários temporários. De acordo com o valor das vendas mensais, os funcionários ganham pontos que determinarão seus salários ao fim de cada mês. Sabe-se que eles trabalharão 3 meses. Faça um programa que:
- a) Cadastre os nomes dos funcionários e suas respectivas vendas mensais.
 - b) Calcule e mostre a pontuação geral de cada funcionário nos três meses. Sabe-se que R\$100,00 equivalem a 1 ponto.
 - c) Calcule e mostre a pontuação geral de todos os funcionários a cada mês.
 - d) Determine e mostre a maior pontuação atingida nos três meses, mostrando o nome do funcionário. Deverão ser desconsiderados empates.
 - e) Determine e mostre o valor total vendido.

Referências

- 1 ASCENCIO, A.F.G.; CAMPOS, E.A.V. de. Fundamentos da Programação de Computadores. Algoritmos, Pascal e C/C++. São Paulo: Pearson Prentice Hall, 2012.
- 2 VAREJÃO, F. M. V. Introdução à programação: uma nova abordagem usando C. Rio de Janeiro: Elsevier, 2015.
- 3 BACKES, A. Linguagem C: completa e descomplicada. Rio de Janeiro: Elsevier, 2013.