

Prática 7

Introdução à programação C/C++
2017/2

Estruturas

Universidade Federal do Rio de Janeiro
Prof. Eduardo Mangeli

Introdução

Uma estrutura é um conjunto de uma ou mais variáveis, que podem ser de tipos diferentes, agrupadas sob um único nome. As variáveis que compõem a estrutura são os seus membros, elementos ou campos. A melhor maneira de declarar estruturas é usar **typedef**. Por exemplo, o comando a seguir define uma tipo chamado `ALUNO`.

```
typedef struct _ALUNO {  
    char nome[TAMNOME];  
    float salario;  
    float imposto;  
} ALUNO;
```

A definição de variável de tipo aluno agora pode ser feita do seguinte modo:

```
ALUNO paulo, carlos, ana, turma[100];
```

Para referenciar um elemento da estrutura usa-se o nome da variável do tipo da estrutura seguida de um ponto e do nome do elemento. Por exemplo, `paulo.ano_entrada = 1999`; armazena o ano em que o aluno paulo entrou na universidade. Para ler o ano em que `ana` entrou na Universidade usa-se o seguinte comando: `scanf("%d", &ana.ano_entrada)`; Para imprimir o salário de carlos usa-se o seguinte comando: `printf("%f\n", carlos.salario)`;

Uma estrutura pode conter vários tipos de variáveis inclusive outras estruturas. Por exemplo, considere as seguintes estruturas:

```
typedef struct _PONTO {  
    float x, y;  
} PONTO;  
  
typedef struct _CIRCUNFERENCIA {  
    float raio;  
    PONTO centro;  
} CIRCUNFERENCIA;
```

Estas estruturas definem elementos geométricos em um espaço de duas dimensões. Observe que a estrutura `CIRCUNFERENCIA` tem como membro uma variável do tipo `PONTO`.

A Listagem 1 mostra como ler as coordenadas de um ponto. A Listagem 2 mostra como ler as coordenadas do centro de uma circunferência.

```
1 #include<stdio.h>  
2  
3 typedef struct _PONTO{  
4     float x, y;  
5 } PONTO;  
6  
7 int main (void){  
8     PONTO p;  
9  
10    printf("Entre com as coordenadas do ponto p\n");  
11    scanf("%f %f", &p.x, &p.y);  
12    printf("Dados lidos\n");  
13    printf("Ponto p: x = %f, y = %f\n", p.x, p.y);  
14    return 0;  
15 }
```

Listagem 1: Exemplo de estrutura

```
1 #include<stdio.h>
2
3 typedef struct _PONTO {
4     float x, y;
5 } PONTO;
6
7 typedef struct _CIRCUNFERENCIA {
8     float raio;
9     PONTO centro;
10 } CIRCUNFERENCIA;
11
12 int main (void){
13     CIRCUNFERENCIA c1;
14
15     printf("Entre com o raio do circulo c1 \n");
16     scanf("%f", &c1.raio);
17     printf("Entre com as coordenadas do centro do circulo c1\n");
18     scanf("%f %f", &c1.centro.x, &c1.centro.y);
19     printf("Dados lidos\n");
20     printf("Circulo c1: raio = %f, x = %f, y = %f\n", c1.raio, c1.centro.x, c1.centro.y);
21     return 0;
22 }
```

Listagem 2: Exemplo de estrutura dentro de estrutura

Exercícios

Exercício 1. Escreva e execute o programa mostrado na Listagem 1.

Exercício 2. Escreva e execute o programa mostrado na Listagem 2.

Exercício 3. Escreva um programa que leia as coordenadas de dois pontos (p1 e p2) e imprima a distância entre eles.

Exercício 4. Escreva um programa que leia as coordenadas de um ponto p1 e os dados de uma circunferência c1. Imprima se o ponto está contido dentro da circunferência.

Exercício 5. Considere a seguinte estrutura:

```
typedef struct _JOGADOR {
    int pontos;
    char nome[42];
} JOGADOR;
```

Escreva um programa que crie um vetor com os dados de 5 jogadores; leia estes dados do teclado e os imprima na ordem em que foram lidos.

Exercício 6. Modifique o programa anterior de modo que os dados dos jogadores sejam impressos em ordem decrescente dos valores dos pontos

Exercício 7. Modifique o programa anterior de modo que após a impressão dos dados dos jogadores seja lido o dado de mais um jogador, e o jogador com o menor número de pontos seja retirado da lista. A nova lista deve ser impressa.