

ED

Ponteiros para Structs

Conteúdo da Aula

- Enxergar um Struct como um novo Datatype no código;
- Referenciar um Struct como um ponteiro para acessar seus atributos internos;

(Alias) Como um Struct se torna um Datatype

Ao criar uma estrutura do tipo struct, praticamente você cria também um novo datatype dentro do programa, por exemplo:

```
typedef struct estr1 tes1;
```

```
struct estr1 {  
    int x[10];  
};
```

O struct **estr1** recebeu um alias (apelido) de **tes1**, agora existe um novo datatype no código que é justamente **tes1**, todas as vezes que quiser utilizar esta estrutura você deve criar um objeto (variável) do tipo **tes1**.

```
tes1 m1;                                for (int i=0; i<10;i++ ) m1.x [ i ] = i;  
m1.x[0] = 1;                             for (int i=0; i<10;i++ ) cout << m1.x [ i ] << endl;  
m1.x[1] = 2;
```

Referenciando um Struct através de um Ponteiro

Considerando que agora temos um novo datatype chamado `tes1`, podemos criar um ponteiro do mesmo tipo (`tes1`) que irá apontar para um struct do tipo `tes1`.

```
tes1 m1;    // cria o objeto m1 do tipo tes1 ( um struct )
tes1 *ptm1; // cria o ponteiro *ptm1 do tipo tes1, isto é que irá receber um endereço do
tipo tes1
ptm1 = &m1; // apontamos ptm1 para m1, isto é, colocamos o endereço de m1 dentro de
ptm1
```

Ao apontarmos o ponteiro ***ptm1** para o endereço **&m1**, o ponteiro ***ptm1** se torna uma espécie de clone do struct **m1**. Agora podemos acessar todos os atributos de **m1** através do **ptm1** como se estivéssemos manipulando diretamente o **m1**. “**O ponteiro ptm1 se tornou o struct m1**”.

```
ptm1-> x[0] = 1;           for (int i=0; i<3; i++) cout << ptm1->x [ i ] << endl;
ptm1-> x[1] = 2;
```

Ao contrário do que acontece ao apontarmos um ponteiro para uma variável, para acessar o valor dos atributos usa-se o ponteiro sem o asterisco é com uma seta no lugar do ponto.

Simulação para estudo

Vamos estudar um código estruturado, contendo subrotinas de cadastro para manipular um struct com os atributos: nome, idade e sexo.

Programa 1 - Struct com Ponteiro

```
#include "iostream"
#include "cstdlib"
#include "math.h"
using namespace std;
typedef struct cadastro cad;
struct cadastro { string nome[2]; int idade[2]; char sexo[2]; };

int main () { setlocale (LC_ALL, "Portuguese");
    cad c1; // criando a tabela c1
    cad *ptrc1; // criando *ptrc1
    ptrc1 = &c1; // apontando para c1

    for (int i=0; i<2;i++)
    { cin.ignore();
        cout << "Nome: "; getline(cin, ptrc1->nome[i]);
        cout << "Idade: "; cin >> ptrc1->idade[i];
        cout << "Sexo: "; cin >> ptrc1->sexo[i]; }
    system("cls");

    cout << "\nAtributos do struct através do ponteiro ";
    for ( int i = 0; i < 2; i++) cout<< "\n" << ptrc1->nome[i]<< " - " << ptrc1->idade[i] << " - " << ptrc1->sexo[i];

    cout << "\n\n\n\nOutra forma de mostrar atributos";
    for (int i=0; i<2;i++) cout<< "\n" << (*ptrc1).nome[i]<< " - " << (*ptrc1).idade[i] << " - " << (*ptrc1).sexo[i];

    cout << "\n\n\n\nImprimindo endereços dos atributos através do ponteiro ";
    for (int i=0; i<2;i++) {
        cout << "\n" << &(*ptrc1).nome[i] << " - " << &(*ptrc1).idade[i] << " - " << &(*ptrc1).sexo[i];
        cout << "\n" << &ptrc1->nome[i] << " - " << &ptrc1->idade[i] << " - " << &ptrc1->sexo[i] << endl; }
    system("pause"); return 0; }
```

TAREFA / AVALIAÇÃO CONTINUADA

Faça um programa com menu de três opções (1-ler linha, 2- mostrar e 3- sair), contendo um STRUCT para cadastrar o nome, a idade, o sexo, salário e salarioNovo de pelo menos duas pessoas. Crie um ponteiro para manipular o STRUCT. O programa deverá conter as seguintes sub rotinas abaixo: (use switch case)

- a) void NovaLinhaStruct ()
- b) void listarLinhasStruct ()
- c) ler_idade ()
- d) ler_sexo ()
- e) ler_nome ()
- f) ler_salario ()
- g) getAumento (double * salario) // aumento de 10% sobre o salário digitado

NOTA: Entregar o código fonte e o print de execução dentro de um único arquivo DOCX ou PDF, somente através do Teams.