

Aula 06 – Structures

1) Introdução

Uma das formas de organizar as informações de um determinado objeto é agrupá-los segundo suas características. Um cliente tem suas características particulares que diferem de um paciente. O primeiro terá informações que são de caráter comercial, como por exemplo o que ele comprou anteriormente, ou seu cnpj, ou o endereço de entrega dos produtos comprados. Um paciente terá outras informações que lhe são particulares, como por exemplo plano de saúde, doenças pré-existentes.

Nas linguagens C e C++ há duas estruturas que possibilitam este agrupamento, em C há o componente chamado “Struct”, em C++ há o componente chamado “Classe”. Nos 2 componentes ou estruturas a ideia inicial é semelhante, ou seja, criar um ambiente que contenha as informações de um determinado objeto, de uma determinada coisa. A diferença entre estas duas estruturas é que a classe possui internamente a possibilidade de usar os recursos da orientação a objetos, como por exemplo o uso de herança entre classes.

2) As características de uma Struct

Uma struct é uma estrutura que tem a seguinte sintaxe.

```
struct nome.da.estruct{  
  
    // Campos que fazem parte da struct  
    // por exemplo Nome, Idade  
  
};
```

Nela colocaremos todos os campos que serão parte da estrutura. Entre as chaves deve-se colocar todos os campos e tipos que são parte da estrutura.

Observe o quadro abaixo.

```
struct Pessoa {  
    string nome;  
    int idade;  
};
```

Nesta struct chamada Pessoa temos 2 campos: nome e idade, cada um com uma tipagem específica.

3) Implementando o código da struct

Vamos criar uma struct que tenha 2 campos: Nome e Idade. Depois de exibir na tela o pedido de entrada do nome e depois da idade, estes valores são escritos na tela.

Observe o código abaixo.

```
#include <iostream>
#include <string>

using namespace std;

struct Pessoa {
    string nome;
    int idade;
};

int main() {
    Pessoa p;

    cout << "Digite o nome: ";
    std::getline(std::cin, p.nome);

    cout << "Digite a idade: ";
    cin >> p.idade;

    cout << "Nome: " << p.nome << ", Idade: " << p.idade << endl;

    return 0;
}
```

4) Considerações da aula

Nesta aula relembramos o conceito de struct. Apesar de ser muito parecida com as classes, possuem diferenças profundas. Nas próximas aulas veremos o uso inicial dos conceitos de OOP.

Bons estudos.