

### Aula 03 – Características da OOP

#### 1) Introdução

A programação orientada à objetos é uma forma de programar que faz uma ruptura com o padrão da programação estruturada. Com a orientação a objetos a programação tende a espelhar em objetos o mundo físico, o mundo real. Nesta aula veremos os conceitos iniciais da OOP.

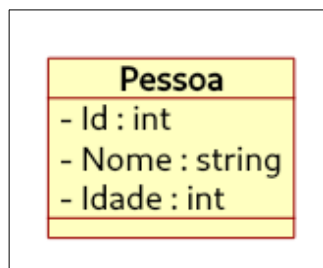
#### 2) A estrutura básica da OOP

Podemos dizer que a OOP é uma quebra de paradigmas na programação. Com ela trazemos alguns conceitos que não usadas na programação estruturada, podemos citar as classes, atributos, métodos, além dos conceitos de herança e polimorfismo. Todos estes conceitos serão abordados nesta disciplina.

Quando pensamos em mundo real temos clientes, pessoas, empresas que são tratadas como objetos e todos estes terão suas propriedades. No mundo real uma pessoa é um objeto com várias propriedades ou atributos. A pessoa tem nome e Idade. Além destas propriedades ou atributos, podemos executar ações ou métodos.

Podemos tomar então que o objeto é uma representação material de algo do mundo real, os atributos são as coisas que ele tem e os métodos são as ações que ele pode fazer ou que você realiza sobre o objeto.

Podemos usar o UML para mostrar visualmente este objeto.



O Classe ou objeto é a pessoa. Os atributos são “Id”, “Nome” e “Idade”.

Neste momento não entraremos no conceito de métodos ou ações do objeto.

### 3) Aplicando a OOP na programação C++

Vamos aplicar no C++ o que foi proposto na modelagem da classe Pessoa.

Observe o código abaixo.

```
#include <iostream>
#include <string>
#include <conio.h>

using namespace std;

class Pessoa{

    public:
        int Id;
        string Nome;
        int Idade;
};

int main(){

    Pessoa pessoa;

    pessoa.Id = 1;
    pessoa.Nome = "José";
    pessoa.Idade = 20;

    cout << "Id: " << pessoa.Id << endl;
    cout << "Nome: " << pessoa.Nome << endl;
    cout << "Idade: " << pessoa.Idade << endl;

    return 0;
}
```

Entendendo o código.

A linha

```
#include <conio.h>
```

Faz o include da biblioteca “conio.h”, sua função é desenhar tela, e é para dos/windows (as funções do conio são úteis para manipular caracteres na tela, especificar cor de carácter e de fundo). As funções típicas utilizadas com o conio.h são clrscr (para limpar a tela), gotoxy (posiciona o cursor na tela nas coordenadas x e y), getch (lê um carácter do teclado e não o mostra na tela), kbhit (testa se alguma tecla foi apertada, é necessário usar o comando getch ou getche antes), delline (apaga a linha que contem o cursor e move todas as linhas abaixo dela uma linha para cima), textcolor (muda

a cor do texto), `textbackground` (muda a cor do fundo), `cprintf` (utilizada para imprimir na tela textos coloridos, é necessário especificar a cor usando a função `textcolor`).

A linha

```
#include <string>
```

Faz o `include` para que o programa trabalhe com letras

A linha

```
#include <iostream>
```

Faz o `include` da biblioteca responsável pelo input e output na entrada e saída padrão.

Temos neste programa a primeira inclusão da orientação a objetos: a classe “Pessoa”. A classe representa o objeto Pessoa que, por sua vez tem alguns atributos, “Id”, “Nome” e “Idade”. Inicialmente podemos fazer uma comparação com o “`struct`”, mas a diferença entre elas é que a classe pode herdar ou ser herdada.

Dentro do “`main`” temos a segunda característica da OOP, veja a linha abaixo.

```
Pessoa pessoa;
```

Temos a criação da variável “`pessoa`” à partir da classe “Pessoa”. Quando fazemos isso criamos uma estrutura que trás todos os atributos e métodos definidos na classe. Isto se chama instanciação.

Logo depois temos o uso da variável “`pessoa`” com os atributos da classe.

```
pessoa.Id = 1;  
pessoa.Nome = "José";  
pessoa.Idade = 20;
```

Por fim usamos a saída padrão para exibir os dados (podem ser todos ou apenas alguns destes dados) desta classe.

```
cout << "Id: " << pessoa.Id << endl;  
cout << "Nome: " << pessoa.Nome << endl;  
cout << "Idade: " << pessoa.Idade << endl;
```

#### 4) Considerações da aula

Nesta aula tivemos a primeira experiência com a orientação a objetos. Na próxima aula trabalharemos com os métodos do objeto.

Bons estudos.