Aula 07 – Orientação a Objetos

Disciplina: Linguagem de Programação
Prof. Allbert Velleniche de Aquino Almeida
E-mail: allbert.almeida@fatec.sp.gov.br
Site: http://www.allbert.com.br
/allbert.almeida

Conceito

- A orientação a objetos é uma ponte entre o mundo real e virtual, a partir desta é possível transcrever a forma como enxergamos os elementos do mundo real em código fonte, a fim de nos possibilitar a construção de sistemas complexos baseados em objetos.

Conceito

- O primeiro passo para utilizar o paradigma da orientação a objetos é perceber o universo em que se deseja trabalhar, descrevendo a estrutura desses elementos, ou seja quais são as características e comportamentos destes para o desenvolvimento do sistema proposto.

Vejamos um exemplo

- Vamos representar um Livro. Um Livro é geralmente composto por um ISBN, título e número de páginas. Vamos guardar essas informações em variáveis:

```
string isbn = "ABC123";
string titulo = "Nome do Livro";
int nropaginas = 123;
```

Vejamos um exemplo

- Para representar um novo livro, precisamos de novas variáveis:

```
string isbn2 = "DEF456";
string titulo2 = "Nome do Livro 2";
int nropaginas2 = 456;
```

Problemas encontrados

- Veja que, como as informações dos livros estão espalhados em diversas variáveis diferentes, é muito fácil misturarmos essas informações dentro do código. Além disso, imagine que precisamos fazer uma validação do ISBN, mas como podemos garantir que essa validação sempre é executada?

Classes

- Desempenham um papel central na orientação a objetos
- Definem um **modelo** para os objetos
- Classes são **tipos de dados**

```
class Livro {
    //escopo
}
```

Propriedades (atributos)

- As propriedades representam características, informações, atributos objetos de uma classe;
- Para ler e escrever nesses atributos, precisamos declará-los utilizando a palavra **public**;

```
class Livro {
    public string Isbn;
    public string Titulo;
    public int NroPaginas;
}
```

Classes

- A partir de uma classe, podemos criar objetos (ou instâncias);

Classe
modelo

Objetos
instâncias

Criando Objetos

- A criação (instanciação) de um objeto é feita usando o operador new;

```
Livro  livro1 = new Livro();
Livro  livro2 = new Livro();
```

Propriedades (atributos)

- Para resolvermos os níveis de acesso dos atributos seria muito mais fácil declararmos todos como “public”, a visibilidade se estenderia a todos a classes da aplicação, no entanto não é uma boa prática fazer isto, pois desta forma toda classe pode alterar os atributos.

Propriedades (atributos)

- Uma forma elegante de tratar isto é manter os atributos como privados e utilizar os "Getters and Setters", como o próprio nome diz "Retornar e alterar". O método "Get()" será responsável por retornar alguma informação do objeto atual, enquanto o "Set(param)" realizará a alteração do objeto.

Definindo Getters e Setters

- Podemos definir uma propriedade autoimplementada;

```
class Livro {
    public string Isbn { get; set; }
    public string Titulo { get; set; }
    public int NroPaginas { get; set; }
}
```

Definindo Getters e Setters

- Forma tradicional:

```
class Livro {
        private string _isbn;
        public string Isbn
        {
            get
            {
                return _isbn;
            }
            set
            {
                _isbn = value;
            }
        }
    }
```

Métodos

- Métodos são operações associadas à classe
- Estas operações podem ser invocadas por alguém que quer interagir com objetos da classe;

```
class Livro {
    void Cadastrar(Livro l) { }
    int TotalLivros() { return 0; }
    static void Ativar(string isbn) { }
}
```

Métodos - Sobrecarga

- Uma classe pode ter várias implementações ou sobrecargas, do mesmo método que diferem quanto ao número de parâmetros ou tipos de parâmetro.

```
class Livro {
    void Cadastrar(Livro l) { }
    void Cadastrar(string titulo) { }
}
```

Construtores

- Construtores são métodos de classe que são executados automaticamente quando um objeto de um determinado tipo é criado;

```
public class Livro
{
    public Livro()
    {
        // Add code here
    }
}
```