

Aula 4

Escrever programas em C# que utilizem classes e objetos

Prof. Ricardo Frohlich da Silva

Programação orientada a objetos

- Conceitos da OO:
- Em orientação a objetos, uma classe é uma estrutura que abstrai um conjunto de objetos com características similares.
- A classe define o comportamento dos objetos através de métodos e atributos
 - Métodos = funções na linguagem estruturada
 - Atributos = variáveis na linguagem estruturada

Programação orientada a objetos

- Modelagem Orientada a Objetos
- Linguagem UML
 - - Padrão para modelagem de sistemas
 - - Amplamente difundida e consolidada
 - - Largamente utilizada na indústria de software
- Mais informações em: <http://www.uml.org>

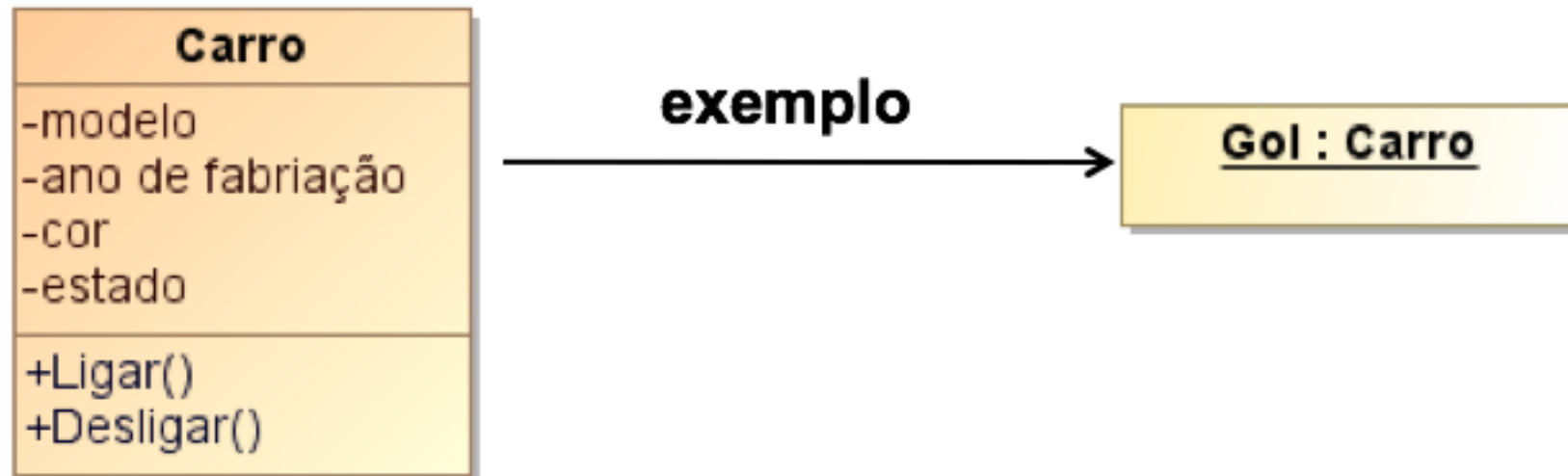
Programação orientada a objetos

- A classe descreve as características e funcionalidades dos objetos



Programação orientada a objetos

- Conceitos da OO:
- O objeto é a instância de uma classe
- Ex: Classe Carro e Objeto Gol



Programação orientada a objetos

- Conceitos da OO:
- É possível ter vários objetos a partir de uma classe

Celta : Carro

Corsa : Carro

Civic : Carro

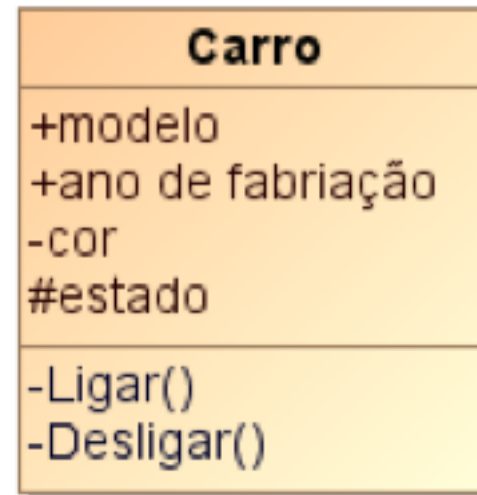
Meriva : Carro

Programação orientada a objetos

- Estado e Comportamento
 - O estado de um objeto é representado por seus atributos
 - O comportamento é representado pelos seus métodos
- Exemplo:
 - Classe Carro
 - Objeto Gol
 - Atributos: modelo, ano, cor, estado
 - Métodos: ligar, desligar

Programação orientada a objetos

- Encapsulamento:
 - É o empacotamento dos atributos e métodos numa classe
 - Proteção dos dados:
 - Público (+)
 - Privado (-)
 - Protegido (#)



Programação orientada a objetos

- Em resumo:
 - Classe: representa um conjunto de objetos
 - Objeto: instância de uma classe
 - Atributo: características do objeto
 - Método: funcionalidades do objeto

Programação orientada a objetos

- E veremos futuramente:
 - Subclasse: classe filha que herda os atributos e os métodos da classe mãe (Herança)
 - Mensagem: troca de informação entre os objetos
 - Associação: utilização de recursos entre objetos
 - Abstração: classe não instanciável
 - Polimorfismo: métodos com muitas formas: sobrecarga, sobrescrita

Programação orientada a objetos

- Perguntas e Respostas:
 - O que é uma Classe?
 - O que é um Objeto?
 - O que é um Método?
 - O que é um Atributo?

Classes e atributos

```
internal class Pessoa
{
    public String nome;
    public int idade;
}
```

- - Qual o nome da classe?
- - A classe possui atributos e métodos? Quais?
- - O que faz a classe Pessoa?

Classes e atributos

- Instanciando a classe Pessoa

```
internal class Pessoa
{
    public String nome;
    public int idade;
}
```

Pessoa.cs

- - Instanciar = criar um objeto
- - Qual o objeto criado?
- - Qual o resultado da execução da classe Principal?

```
internal class Program
{
    static void Main(string[] args)
    {
        Pessoa p = new Pessoa();
        Console.WriteLine("A pessoa foi instanciada!");
    }
}
```

Program.cs

Classes e atributos

- Classe Pessoa com valores nos atributos

```
internal class Pessoa
{
    public String nome = "Ricardo";
    public int idade = 30;
}
```

- - Atributos definidos na classe com valores padrão (default);

Classes e atributos

- Instanciando a classe Pessoa novamente e obtendo o valor dos atributos

```
internal class Program
{
    static void Main(string[] args)
    {
        Pessoa p = new Pessoa();
        Console.WriteLine("O nome da pessoa é: "+p.nome);
        Console.WriteLine("E tem "+p.idade+" anos!");
    }
}
```

- Qual o resultado da execução da classe Principal?
- Acesso direto aos atributos e aos valores dos atributos !!! Como isso é possível ? Pelo fato de os atributos estarem definidos como public.
- Se o(s) atributo(s) estivesse(m) com visibilidade private, apenas a própria classe poderia acessar o(s) atributo(s).

Classes e atributos

- Pergunta: mas se a Classe deve ser um “molde”, todos os objetos desse molde sairão com os valores default quando instanciados? Isso é correto? É o que espera-se?
- Normalmente, cria-se um objeto e, após, define-se os valores dos atributos.
- A ideia é: cada objeto tenha seus atributos valorados de acordo com a necessidade, após a instanciação, e não a classe já passe esses valores no momento da instanciação.
- Salvo exceções !

Classes e atributos

- Criando objetos (instanciando classes) em somente depois definindo valores aos atributos;

```
internal class Pessoa
{
    public String nome;
    public int idade;
}
```

```
static void Main(string[] args)
{
    Pessoa p1 = new Pessoa();
    p1.nome = "João";
    p1.idade = 55;

    Console.WriteLine("O nome da pessoa 1 é: " + p1.nome);
    Console.WriteLine("E tem " + p1.idade + " anos!");
}
```

Classes e atributos

- Jogo rápido: exibir o nome e a idade de três pessoas, ou seja: criar três objetos do tipo Pessoa;

Classes e atributos

```
static void Main(string[] args)
{
    Console.WriteLine("Oi, estamos na classe principal");
    Pessoa p1 = new Pessoa();
    p1.nome = "Ricardo Frohlich";
    p1.idade = 38;

    Pessoa p2 = new Pessoa();
    p2.nome = "André";
    p2.idade = 25;

    Pessoa p3 = new Pessoa();
    p3.nome = "Ana Paula";
    p3.idade = 32;

    Console.WriteLine("Nome: " + p1.nome);
    Console.WriteLine("Idade: " + p1.idade);

    Console.WriteLine("Nome: " + p2.nome);
    Console.WriteLine("Idade: " + p2.idade);

    Console.WriteLine("Nome: " + p3.nome);
    Console.WriteLine("Idade: " + p3.idade);
}
```

```
static void Main(string[] args)
{
    Console.WriteLine("Oi, estamos na classe principal");
    Pessoa p1 = new Pessoa();
    p1.nome = "Ricardo Frohlich";
    p1.idade = 38;

    Pessoa p2 = new Pessoa();
    p2.nome = "André";
    p2.idade = 25;

    Pessoa p3 = new Pessoa();
    p3.nome = "Ana Paula";
    p3.idade = 32;

    Console.WriteLine("Nome: " + p1.nome);
    Console.WriteLine("Idade: " + p1.idade);

    Console.WriteLine("Nome: " + p2.nome);
    Console.WriteLine("Idade: " + p2.idade);

    Console.WriteLine("Nome: " + p3.nome);
    Console.WriteLine("Idade: " + p3.idade);
}
```

Classes e atributos

- Jogo rápido (2): crie uma classe chamada Professor que contenha um atributo público chamado nome, do tipo String. Crie também uma classe chamada Laboratorio, que contenha um atributo público, chamado local, do tipo String.
- Além disso, na main instancie as classes Professor e Laboratorio, definindo valor aos atributos e exibindo na tela o resultado da criação desses objetos.
- Por exemplo: o resultado da execução do programa deve ser algo similar com:
 - O nome do professor é: Ricardo da Silva
 - O local da aula é: Sala 108

Classes e atributos

```
internal class Professor
{
    public String nome;
}
```

```
internal class Laboratorio
{
    public String local;
}
```

```
static void Main(string[] args)
{
    Professor p = new Professor();
    p.nome = "Ricardo Frohlich da Silva";

    Laboratorio lab = new Laboratorio();
    lab.local = "Lab 108/P3";

    Console.WriteLine("O nome do professor é: "+p.nome);
    Console.WriteLine("O local é: "+lab.local);
}
```

Classes e atributos

- Jogo rápido (3): Crie uma classe Pessoa com os atributos nome, idade e gênero. Na Main, faça a leitura dos dados pelo teclado e faça a definição dos atributos da classe e imprima as informações da pessoa na tela.

Classes e atributos

- Jogo rápido (3):

```
internal class Pessoa
{
    public string nome;
    public int idade;
    public string genero;
}
```

```
static void Main(string[] args)
{
    Pessoa p = new Pessoa();
    Console.WriteLine("Digite o nome da pessoa: ");
    p.nome = Console.ReadLine();
    Console.WriteLine("Digite a idade da pessoa: ");
    p.idade = int.Parse(Console.ReadLine());
    Console.WriteLine("O gênero da pessoa: ");
    p.genero = Console.ReadLine();

    Console.Clear();

    Console.WriteLine("Nome: "+p.nome);
    Console.WriteLine("Idade: "+p.idade);
    Console.WriteLine("Gênero: "+p.genero);
}
```


Exercícios

- 1 - Crie uma classe "Jogo" com os seguintes atributos: Título, Gênero, Plataforma (por exemplo, PC, Xbox, PlayStation), Ano de lançamento. Na main, instancie um objeto e faça a leitura pelo teclado.
- 2 - Crie uma classe "Estúdio" com os seguintes atributos: Nome, Ano de fundação, País de origem e Número de jogos produzidos. Na main, instancie um objeto e faça a leitura pelo teclado.

Exercícios

- 3 - Crie uma classe chamada Carro que possua 3 atributos: marca, modelo e anoFabricacao. Além disso, na main crie 2 objetos do tipo Carro e exibir na tela os valores dos atributos criados. Esses valores devem ser solicitados ao usuário, no programa.
- 4 - Crie uma classe Livro com os atributos título, autor e ano de publicação. Faça a instanciação de 3 objetos e imprima as informações do livro na tela.