

Programação Orientada à Objetos

Aula 05

Classes, Objetos, Atributos e Métodos

Henrique Poyatos

henrique.poyatos@bandtec.com.br

Classes x Objetos

A principal diferença entre Classes e Objetos:



CLASSE



OBJETO

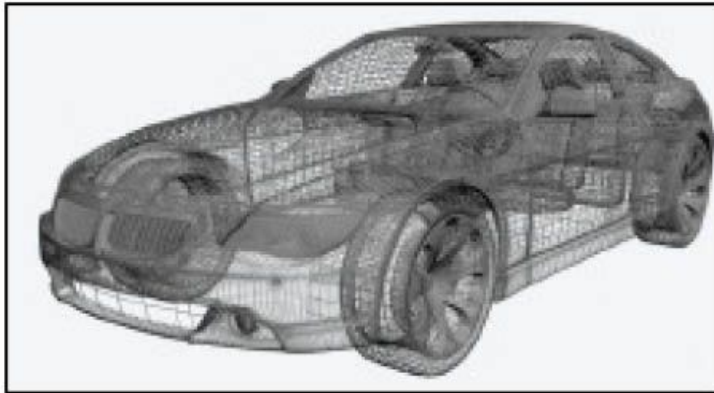
Exemplo Classe Pessoa

A principal diferença entre Classes e Objetos:



Exemplo Carro

A principal diferença entre Classes e Objetos:

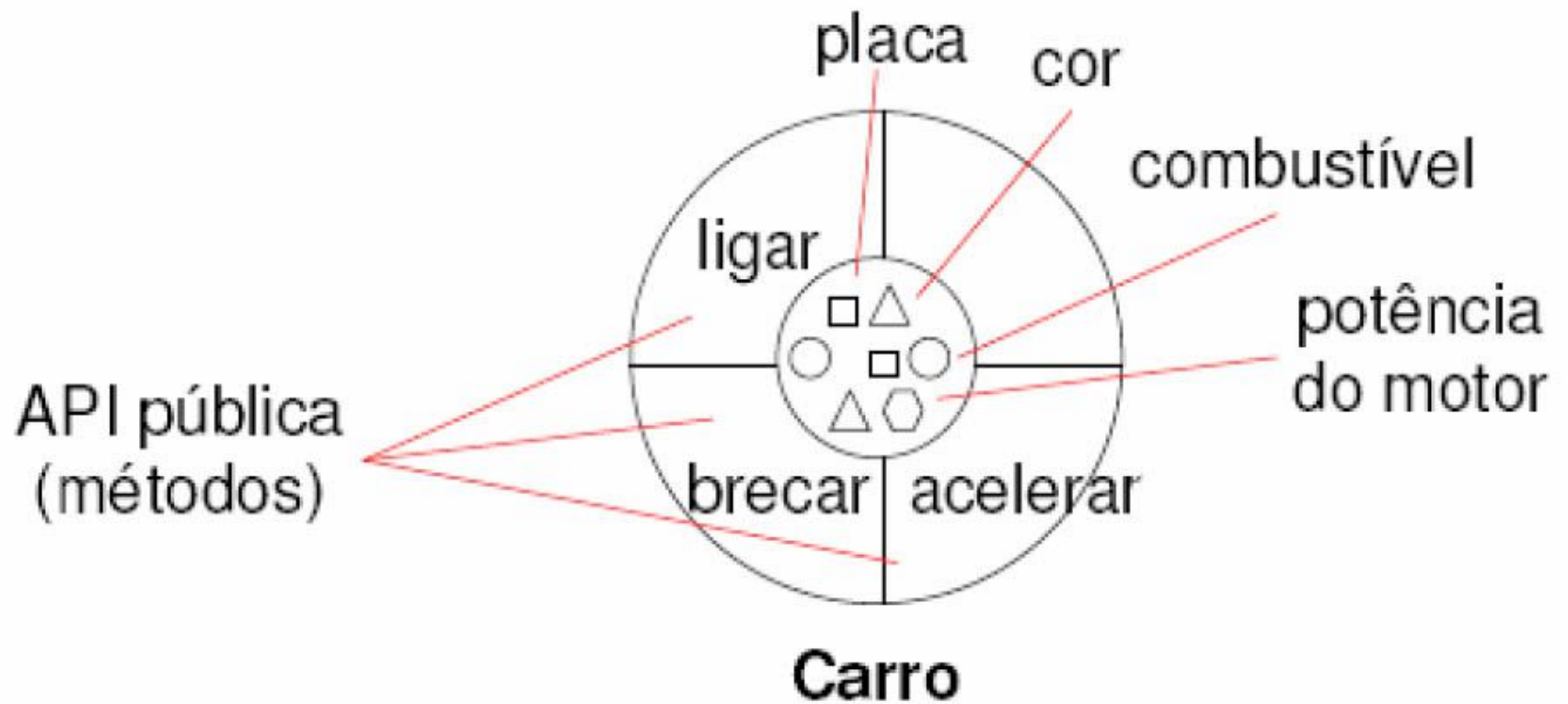


CLASSE



OBJETO

Exemplo Classe Carro



Classe

Declaração

Sintaxe

```
namespace Projeto
{
    public class Carro
    {

    }
}
```

Nome da Classe → Como a classe será identificada dentro do programa.

class → Indicação que o elemento é uma classe.

OOP – Classes x Objetos

- **Classe**

Representa uma categoria de objetos que possuem exatamente as mesmas características (atributos) e executam as mesmas tarefas (métodos).

Exemplo: Automóvel

- **Objeto**

É a instância única de uma classe. Um objeto é único mesmo que seus atributos apresentem valores idênticos ao de outro objeto.

Exemplo: Uno Vivace 1.0 cinza ano 2011 placa ABC-3456

OOP – Classes – Atributos e Métodos

- **Atributos**

Características que todos os objetos daquela classe terão



- ★ Placa
- ★ Cor
- ★ Número de Portas
- ★ Fabricante
- ★ Modelo
- ★ Ano de Fabricação

- **Métodos**

Representam ações que o objeto poderá executar com o intuito de atingir um objetivo.



- ★ Acelerar
- ★ Frear
- ★ Acender faróis
- ★ Buzinar
- ★ Ligar
- ★ Desligar

OOP – Alguns Conceitos

- **Abstração**

O exercício de abstração – seria pegar algumas classes e determinar o que elas têm em comum, generalizando o conceito ainda mais

- ★ **Exemplos:**

O que um carro, avião e um barco possuem em comum? Todos servem para se locomover, são portanto Veículos.

O que um professor, coordenador, aluno e diretor têm em comum? Todos são pessoas, e como pessoas, compartilham características em comum;

Resultado: Reaproveitamento de Código-fonte.

- **Encapsulamento**

Compreende esconder a complexidade de uma execução dentro de um método; Podemos utilizá-lo sem entrar em detalhes como funciona.

- ★ **Exemplo:** Como funciona o processo de aceleração de um veículo? Saiba que ele muda em diferentes tipos de carros (carburados, injeção eletrônica, etc); No entanto, todos são padronizados com uma interface simples: O pedal direito.

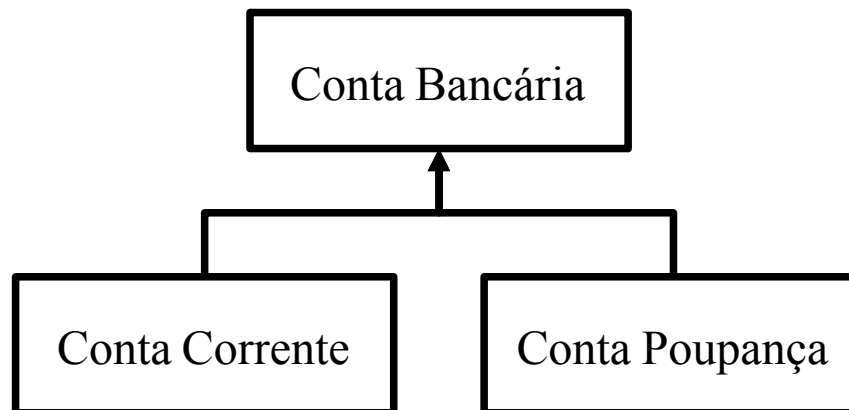
OOP – Alguns Conceitos

- **Herança**

A herança é uma técnica útil para ampliar a abstração de dados. é o mecanismo para expressar a similaridade entre Classes, simplificando a definição de Classes iguais a outras que já foram definidas.

O conceito de herança trabalha com os conceitos de superclasses e sub-classes. Uma superclasse é uma classe que possui classes derivadas a partir dela. As sub-classes, ao serem derivadas a partir de uma superclasse, herdam suas características, ou seja, seus atributos e métodos.

- Sub-classes podem ter seus próprios atributos e métodos.



OOP – Alguns Conceitos

- **Polimorfismo**

O conceito de polimorfismo está associado à herança. O polimorfismo prevê a “redeclaração de métodos” previamente herdados por uma classe. Esses métodos, embora semelhantes, diferem de alguma forma da implementação utilizada na superclasse, sendo necessário, portanto, reimplementá-los na sub-classe.

Podemos considerar uma Classes Veículo, e subclasses desta classe Veículo de Câmbio Comum, Veículo Automático e hoje em dia Veículo com Câmbio Borboleta.



OOP – Alguns Conceitos

- **Associações**

Associações simples descrevem vínculos que normalmente ocorrem entre duas classes.

Para associações simples considere que um objeto pode estar associado a outros objetos da mesma classe num mesmo momento.

Exemplo: um Controlador de Vôo controla vários vôos ao mesmo tempo.

- **Agregações**

Agregação é um tipo especial de associação onde demonstra-se que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe (chamados objeto-parte).

Um objeto-parte nunca pode participar de dois objetos-todo ao mesmo tempo.

Exemplo: num vôo existem passageiros. Podemos considerar que Passageiro é objeto-parte do objeto-todo vôo. Mas um passageiro não pode estar em dois vôos ao mesmo tempo.

OOP – Alguns Conceitos

- **Composições**

Uma agregação de composição representa um vínculo mais forte entre os objetos-todo e os objetos-parte, procurando demonstrar que os objetos-parte têm de pertencer exclusivamente a um único objeto-todo com que se relacionam. Em uma composição, um mesmo objeto-parte não pode associar-se a mais de um objeto-todo.

Exemplo: um piloto de avião possui uma documentação que o habilita a voar (o brevê). Podemos considerar este documento como objeto-parte do objeto-todo piloto. Porém este documento não pode fazer parte de outro objeto-todo que não o piloto ao qual ele pertence. Em momento algum.

Atributos

O que são?

São os tipos de características que aquele tipo de classe possui.
Serão as **VARIÁVEIS** pertencentes à classe.



Atributos → “Os ingredientes desta receita”

Classe com seus Atributos

Declaração

Sintaxe

```
namespace Projeto
```

```
{
```

```
    public class Carro
```

```
    {
```

```
        public int placa;
```

```
        public string marca;
```

```
        public string modelo;
```

```
        public string cor;
```

```
    }
```

```
}
```

Nome do Atributo → Como o atributo será identificado dentro do programa.

public → Tipo de Acesso
Trata-se de atributos públicos, podendo ser carregados e acessados por qualquer um.

tipo de dado → Qual o tipo de dado deste atributo.. como uma variável. O tipo de dado pode ser outra classe ex. String.

Métodos

O que são?

São as ações e procedimentos que aquele tipo de classe possui.
Serão as **FUNÇÕES** pertencentes à classe.



Métodos → “As ações desta receita”

Classe com seus Métodos

Declaração

Sintaxe

```
namespace Projeto {  
    public class Carro  
    {  
        public int placa;  
        public string marca;  
        public string modelo;  
        public string cor;  
  
        public string andar()  
        {  
            return "O Carro está em movimento.";  
        }  
    }  
}
```

public → Tipo de Acesso
Trata-se de atributos públicos, podendo ser carregados e acessados por qualquer um.

tipo de dado → Qual o tipo de retorno do método

Procedimentos do método.

Nome do Método → Como o método será identificado dentro do programa. Sempre acompanhado por parênteses.

Instanciação de um Objeto

- Os objetos são criados ou instanciados - “ganham vida”.
- Objetos são instanciados através da instrução ***new*** na linguagem Java (e em várias outras).
- Um Objeto será sempre instanciado **a partir de uma Classe** (“Como iremos fazer um bolo sem sua receita?”).

Objeto

Instanciação do Objeto a partir de uma Classe

Sintaxe

```
namespace Projeto
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Carro meuCarro = new Carro();
```

```
        }
```

```
    }
```

```
}
```

Nome do Objeto → Como o objeto
Será identificado no programa.

new → Palavra
reservada
do Java que indica
a instanciação do
objeto, a partir
de uma classe.

tipo de classe → Qual o classe
da qual o objeto será instanciado.

Nome do Método Construtor → Aqui vai
O nome do método construtor que sempre
possui o mesmo nome da classe.