

Introdução a orientação a objetos

Introdução

- Orientação a objetos é um paradigma de programação que se baseia na ideia de que um programa de computador deve ser construído em torno de "objetos", que são estruturas de dados que contêm informações e comportamentos relacionados.
- Isso torna nosso software capaz de simular, com eficácia, eventos do dia a dia em sistemas digitais;

História

- Em 1962, Kristen Nygaard e Ole-Joan Dahk, pesquisadores do Centro Norueguês de Computação, da Universidade de Oslo, desenvolveram uma linguagem de simulação de eventos, dando origem ao legado da orientação a objetos;
- Na década de 1970, a linguagem de programação Smalltalk foi desenvolvida por Alan Kay e sua equipe, e foi a primeira linguagem de programação orientada a objetos completa.
- Desde então...

Vantagens

- Fornece uma estrutura modular para a construção de programas;
- O software se torna mais fácil de manter;
- Reuso de código -> desenvolvimento mais rápido;
- Objetos podem ser reutilizados em aplicações diferentes;

Abstração

- Observar (um ou mais elementos de um todo), avaliando características e propriedades em separado;
- De acordo com Correia (2006): *“Pelo principio da abstração, isolamos os objetos que queremos representar do ambiente em se situam, e nesses objetos representamos somente características que são relevantes para o problema em questão.”*
- Representar uma entidade do mundo real na forma de ideias;
- Entidades abstraídas podem se comunicar entre si, por meio de mensagens;

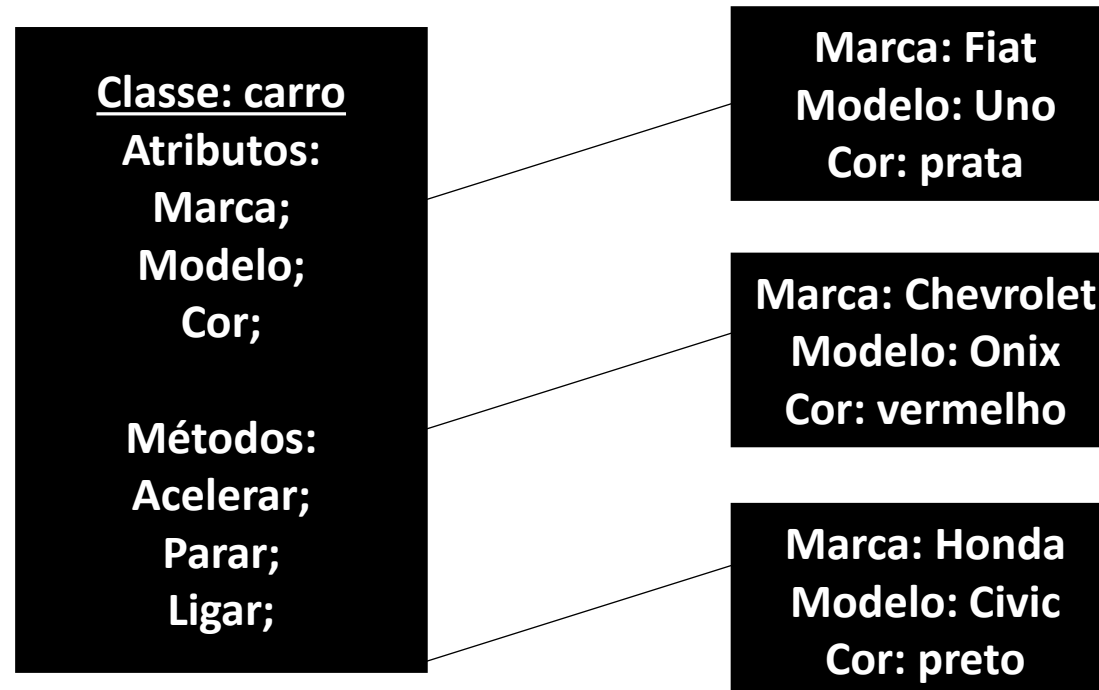
O paradigma

- Baseia-se no conceito de agrupar dados, características e comportamentos (classes) em uma coleção chamada objetos!

Classe: representação abstrata
de coisas do mundo real

Atributos: características da classe

Métodos: o que a classe faz.
A função que ela tem!



Objeto: materialização de uma
classe

Classe

- Representa uma ideia ou conceito;
- A classe é uma “espécie” de molde que define quais serão as propriedades e os métodos que um objeto pode ter! Tipos personalizados de dados;
- Uma boa maneira de definir uma classe é pensar em substantivos!
- A classe deve representar bem o seu propósito, seja no nome, em seus atributos e métodos! Todos devem ser condizentes!

Objeto

- Ocorrência específica de uma classe;
- Representa entidades do mundo real, como: carros, pessoas, contas correntes e outros conceitos;
- Tem características próprias (atributos) e executa ações (métodos) vindos da classe que foi originado!

Atributos

- Representa as características dos objetos criados a partir da classe;
- Ele é responsável por definir a estrutura de dados da classe;
- Há dois tipos de atributos:
 - Estáticos: mantem o mesmo valor durante sua existência, por exemplo: data de nascimento;
 - Dinâmicos: valores que variam com o passar do tempo, por exemplo: idade.

Método

- Lógica contida em uma classe para atribuir comportamentos (sequencia de comandos);
- Serve para identificar serviços e comportamentos que uma classe oferece, como: executar cálculos, manipular dados ou interagir com outros objetos;
- O ato de chamar um método é a passagem de mensagem para o objeto!
- Exemplo: classe Pessoa pode ter os métodos nascer(), comer(), morrer()

Sintaxe em C#

```
[modificadores] class NomeDaClasse  
{  
    // membros da classe  
}
```

Prática

1. Crie uma classe "Carro" com atributos como marca, modelo, ano e velocidade máxima, além de métodos que retorne as informações da classe
2. Crie uma classe "Pessoa" com atributos como nome, idade, endereço e um método de saudação que retorna uma mensagem de boas-vindas.
3. Crie uma classe "Livro" com atributos como título, autor, número de páginas e um método que retorna a descrição completa do livro.
4. Crie uma classe "Conta Bancária" com atributos como nome do titular, número da conta e saldo, além de métodos para depositar e sacar dinheiro.
5. Crie uma classe "Retângulo" com atributos como largura e altura e métodos para calcular e retornar a área e o perímetro do retângulo.