

Programação Orientada a Objetos

Aula 01 - Introdução à Programação Orientada a Objetos

Apresentação

Nesta aula, vamos iniciar o estudo da Programação Orientada a Objetos(POO). Conheceremos, de forma introdutória, alguns conceitos utilizados nessa área, como classes e objetos. Através desses novos conhecimentos, você, caro aluno, será instigado a observar o mundo a sua volta a partir do ponto de vista dessa ciência.



Vídeo 01 - Apresentação

Objetivos

Ao final desta aula, você será capaz de:

- Diferenciar Programação Estruturada e POO.
- Conceituar Classes e Objetos.
- Conceituar Atributos e Métodos.
- Entender a realidade sob o ponto de vista da POO.

Considerações Iniciais



Olá pessoal, eu sou Dígito!

E estarei com você nessa grande aventura sobre a Programação Orientada a Objetos.

Nossa missão é explorar e descobrir os principais conceitos dessa tecnologia e juntos darmos os primeiros passos nesse admirável mundo novo. Vem comigo!!!

A Programação Orientada a Objetos (POO) é um [paradigma](#) (Segundo o dicionário Houaiss, _paradigma significa modelo, _padrão, _exemplo. Assim um paradigma de POO é a maneira como pensamos e iremos fazer nossos programas.) de programação de computadores que usa os conceitos de **Objetos** e **Classes** como elementos centrais para representar e processar dados usados nos programas.

E é dessa maneira que iremos aprender a ver o mundo em nossa volta. O mundo dos objetos!

Como Tudo Começou

De acordo com dados bibliográficos, os conceitos da programação orientada a objetos (POO) surgiram no final da década de 1960, quando a linguagem Simula-68 introduziu os conceitos de objetos e troca de mensagens para construção de programas.

Tais conceitos foram posteriormente amadurecidos e aprimorados durante a década de 1970 pela linguagem de programação Smalltalk, desenvolvida no laboratório de pesquisa da Xerox, nos Estados Unidos. Entretanto, a popularização

da POO só se deu ao longo da década de 80 e 90, com as linguagens C++ e Java.

Java é uma das linguagens mais difundidas nos dias atuais e por conta disso foi escolhida para iniciarmos nossos estudos de POO. Ela surgiu, oficialmente, em 1995 e também tem uma historinha interessante... mas, essa história só veremos na próxima aula.

Um dos principais pesquisadores que introduziu os conceitos de POO, foi o cientista Alan Kay (veja a Figura 1) da Xerox, um dos criadores da linguagem Smalltalk. Durante suas pesquisas, Alan desenvolveu a idéia de que poderíamos construir um programa usando conceitos e abstrações do mundo real, como objetos, troca de mensagens. Houve um dia em que Alan Kay pensou:

Figura 01 - AlanKay



Então, a partir desses questionamentos, ele começou a desenvolver suas ideias sobre um sistema de software fazendo uma comparação com o sistema de seres vivos, como ilustra a Figura 2.

Figura 02 - AlanKay pensando na programação OO



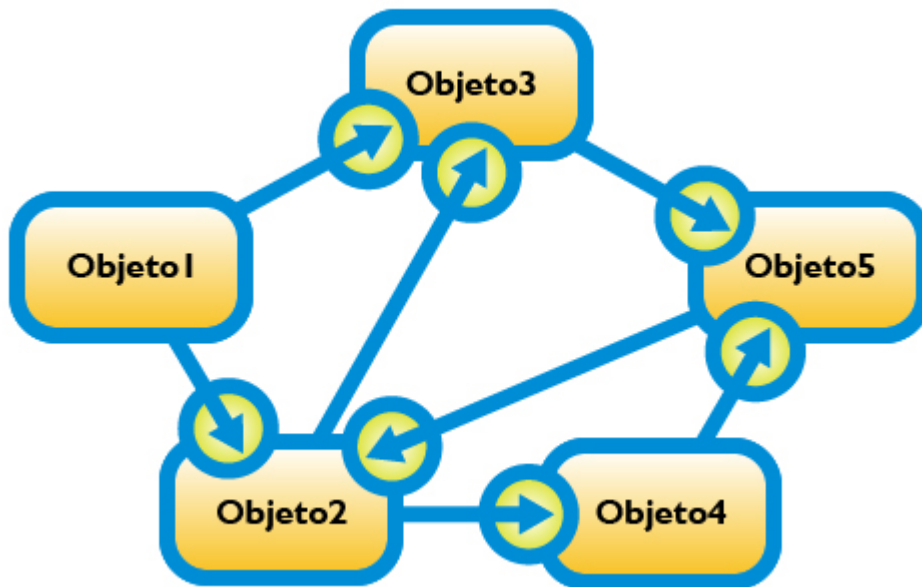
Nossa! Quantas ideias!!! Mas, não se preocupe que iremos falar e aprender sobre cada uma delas ao longo das diversas aulas deste curso.

Conceitos da Programação Orientada a Objetos

Até aqui, você aprendeu o conceito de programação estruturada, na qual um programa é construído por meio de funções e procedimentos. Na programação estruturada na linguagem C, um programa é definido por meio de uma função principal a qual faz chamadas a outras funções implementadas por nós, programadores.

Na programação OO, a forma de pensar (o paradigma) é diferente. Um programa é visto como um conjunto de objetos que se comunicam através de mensagens. Perceba que a Figura 3 ilustra essa ideia. Cada objeto mantém dados internos, chamados de atributos. Dessa forma, um sistema (programa de computador) desenvolvido usando a POO possui objetos que colaboram entre si, executando tarefas específicas em busca de um objetivo comum.

Figura 03 - Colaboração de Objetos



A Programação Orientada a Objetos está sedimentada sobre quatro pilares derivados do princípio da abstração, são eles: **Encapsulamento, Herança, Composição e Polimorfismo.**

O **Princípio da abstração** é a nossa capacidade de abstrair a complexidade de um sistema e se concentrar em apenas partes desse sistema, que é o tema central de nosso estudo ou foco em um dado momento.

Por exemplo: quando um médico torna-se um especialista em algum órgão do nosso corpo (exemplo, o coração), ele abstrai sem desconsiderar as influências dos outros órgãos e foca apenas sua atenção nesse órgão. Outro exemplo: quando um engenheiro civil projeta um edifício, ele abstrai a complexidade como um todo e planeja seu projeto em partes: sistema elétrico, hidráulico, estrutural, etc.

Veremos durante todo esse curso cada um desses pilares que compõem a Orientação a Objetos.

Figura 04 - Pilares da Orientação a Objetos



Vídeo 02 - Programação Estruturada x POO

Objetos

O que são objetos? Na programação OO, objetos são usados para representar entidades do mundo real ou computacional. Ou seja, os objetos são usados para representar aqueles elementos e abstrações que fazem parte da solução do programa que estamos desenvolvendo.

Se observarmos ao nosso redor, veremos várias entidades ou abstrações as quais podem ser representadas como objetos no nosso programa. As pessoas e seus carros podem ser vistas como objetos. Na Figura 5, ilustramos, por exemplo, a pessoa de nome "Camila" que tem um carro "Ferrari".

Os objetos possuem características pelas quais os identificamos e finalidades para as quais os utilizamos. Essas características são tipicamente chamadas de atributos, no linguajar da programação OO. Cada objeto Pessoa, por exemplo, possui um RG, um nome, uma data de nascimento, etc. Já os objetos do tipo Carro

possuem um tipo, uma cor, uma quantidade de portas. A Figura 3 mostra os diferentes atributos que os objetos “Camila” e “Ferrari” possuem, assim como os valores que estão assumindo.

Finalmente, objetos podem também ter comportamentos associados. Objetos do tipo Pessoa, por exemplo, podem andar, correr ou dirigir carros. Já os objetos do tipo Carro podem ter os seguintes comportamentos: ligar, desligar, acelerar, frear. A Figura 5 também ilustra os comportamentos que objetos Pessoa e Carro podem ter.

Assim, na POO os objetos possuem características e comportamentos. As características também podem ser chamadas de dados ou atributos. Enquanto os comportamentos também podem ser chamados de operações ou métodos.

Exemplos de objetos:

Figura 05 - Exemplos de objetos



Vídeo 03 - Objetos

Atividade 01

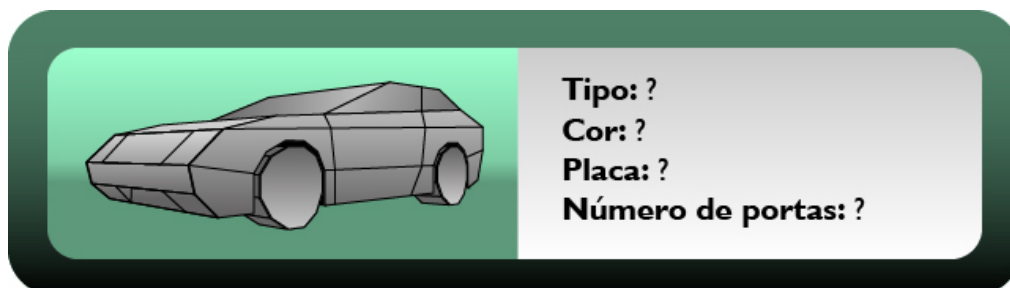
1. Observe os objetos a sua volta, escolha um deles e descreva de 3 a 5 características (atributos) e comportamentos (métodos).

Classes

Agora que você aprendeu o conceito de **Objetos** em programação OO, podemos passar para outro ponto da nossa conversa. Podemos dizer que no fundo todos nós pensamos e organizamos o mundo segundo a nossa visão das características e finalidades das coisas, logo pensamos o mundo Orientado a Objetos. É muito comum (e com muita frequência o fazemos) criar grupos de objetos com base nas suas características e comportamento. Se pensarmos no universo de objetos Carro, por exemplo, vamos perceber que todos eles sempre possuem os atributos tipo, cor e número de portas. O mesmo vale para o universo de objetos Pessoa, que sempre vão possuir um nome, uma data de nascimento e, possivelmente, um RG.

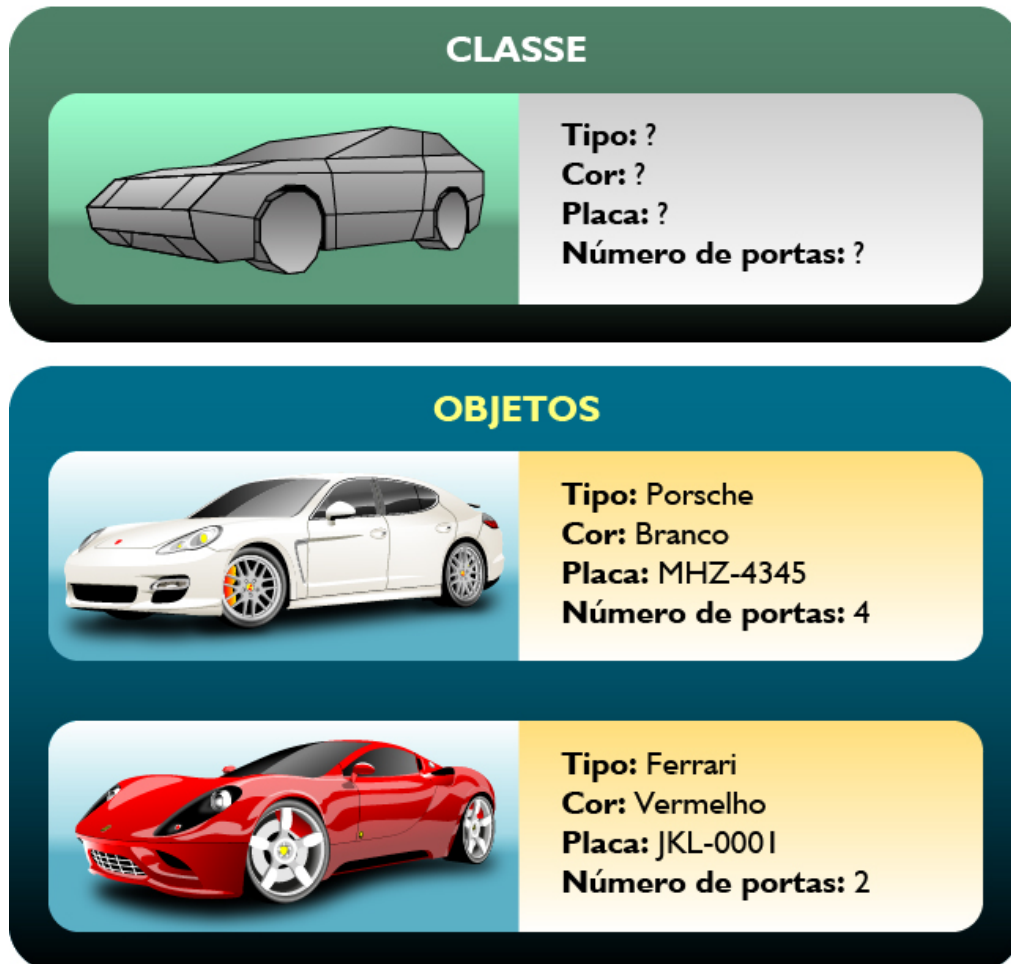
A modelagem e programação de um conjunto de objetos que possuem características (atributos) e comportamentos (métodos) comuns é feita na POO usando o conceito de **Classe**. Cada classe funciona no fundo como um molde para a criação de um dado objeto. Os objetos são vistos como representações concretas (instâncias) das classes. A Figura 6, por exemplo, ilustra uma classe que representa os objetos Carro. Como pode ser observado, a classe define que objetos devem ter tipo, cor, placa e número de portas, mas não indica explicitamente quais são seus valores.

Figura 06 - Classe Carro



A Figura 7 mostra como a partir da classe (entidade abstrata) podemos criar vários objetos diferentes (entidades concretas). Dois diferentes carros foram criados tomando como base a estrutura da classe. Outros objetos carro podem ser criados usando a classe Carro como molde.

Figura 07 - Classe Carro e dois objetos concretos



Vídeo 04 - Classes

Atividade 02

Observando o exemplo da Figura 7, imagine se a classe fosse: Livro, Celular, Mamíferos, Aves e Flores.

1. Como seriam seus objetos?
2. Escreva apenas as características (ou atributos) das classes e objetos!

Exemplos de Classes e Objetos

Atenção!

Em alguns livros, esses dois conceitos – classes e objetos – podem ser usados indistintamente.

Como vimos, diferentes classes (e seus respectivos objetos) podem ser criados em POO, para facilitar a programação do sistema. Na prática, não existem limites ou restrições sobre o que você pode modelar e implementar. A seguir, apresentamos alguns exemplos de classes/objetos para representação de diferentes abstrações em sistemas:

- Objetos físicos (um livro, uma mercadoria, uma nota fiscal);
- Funções de pessoas para os sistemas (cliente, vendedor);
- Eventos (uma compra, um telefonema);
- Lugares (loja matriz, revenda norte).

Leitura Complementar

MASSAGO, Sadao; SCHÜTZER, Waldeck. Programação Java. Disponível em: <<http://www.dm.ufscar.br/~waldeck/curso/java/>>. Acesso em: 13 maio 2010.

Material *on-line* introdutório sobre Java.

Resumo

Nesta aula, você estudou o que é a Programação Orientada a Objetos e o que a diferencia da Programação Estruturada. Conhecemos os conceitos de Classes, Objetos, Atributos e Métodos. Além disso, você começou a observar o mundo a nossa volta sob o ponto de vista da Programação Orientada a Objetos.

Autoavaliação

1. Quais as diferenças entre a Programação Estruturada e a POO?
2. Defina o conceito de classe e objetos.
3. Agora, seguindo o exemplo da Figura 3, descreva os possíveis comportamentos (ou métodos) para cada um desses objetos.
4. Para que servem atributos e métodos?
5. Qual é a diferença entre classe e objeto?
6. Identifique classes e objetos para um sistema escolar responsável por controlar as turmas, professores, alunos, notas dos alunos etc. Apresente em detalhes também quem são os atributos e métodos de cada classe que você modelou.

Referências

BARNES, David J.; KÖLLING Michael. **Programação orientada a objetos com Java**. São Paulo: Editora Pearson Prentice Hall, 2004.

DEITEL, H. M.; DEITEL, P. J. **Java como programar**. Porto Alegre: Bookman, 2003.

LEMAY, Laura. **Aprenda Java em 21 dias**. Tradução: Daniel Vieira. Rio de Janeiro: Campos, 2003.

SANTOS, Rafael. **Introdução à programação orientada a objetos usando Java**. São Paulo: Editora Campus, 2003.

SINTES, Anthony. **Aprenda a programar orientada a objeto em 21 dias**. Tradução: João Eduardo Nóbrega Tortello. São Paulo: Pearson Education do Brasil, 2002.

THE JAVA: tutorials. **[Tutorial Oficial Java]**. Disponível em <http://java.sun.com/docs/books/tutorial/>. Acesso em: 13 maio 2010.