

INSTITUTO FEDERAL DE MATO GROSSO DO SUL
CAMPUS NAVIRAÍ

NOTAS DE AULA DE
ANÁLISE E PROJETO ORIENTADO
A OBJETO
II

Prof. MSc. Luiz F. Picolo

NAVIRAÍ - MS

1 Introdução

*Persistência é a irmã gêmea da excelência.
Uma é a mãe da qualidade, a outra é a mãe do tempo.*

Marabel Morgan

Para que haja um entendimento melhor sobre alguns conceitos que serão utilizados no decorrer desta disciplina, devemos, neste momento, recapitular alguns conhecimentos obtidos em semestres anteriores. Tais conhecimento se referem ao paradigma da orientação a objetos, ou, POO. Portanto, este capítulo introdutório terá como objetivo fazer esta introdução sobre orientação a objeto. Contudo, apenas será revisto conceitos que serão utilizados em aula e não todo o conteúdo referente a POO.

1.1 Orientação a objeto

Programação Orientada a Objetos (também conhecida pela sua sigla POO) é um modelo de análise, projeto e programação de software baseado na composição e interação entre diversas unidades chamadas de objetos. A POO é um dos 4 principais paradigmas de programação (as outras são programação imperativa, funcional e lógica). A programação orientada a objetos se ocupa de realizar um projeto de software utilizando uma linguagem de programação orientada a objetos, por exemplo, a linguagem Java, que aceita a implementação direta de objetos e fornece recursos para definir as classes de objetos [Sommerville 2003].

1.1.1 Projeto orientado e objeto

Já o projeto orientado a objetos é uma estratégia de projeto em que os projetistas de sistema pensam em termos de ‘coisas’, em vez de operações ou funções. O sistema em funcionamento é constituído de objetos que interagem entre si, que mantêm seu próprio estado local e fornecem operações com base nessas informações de estado [Sommerville 2003].

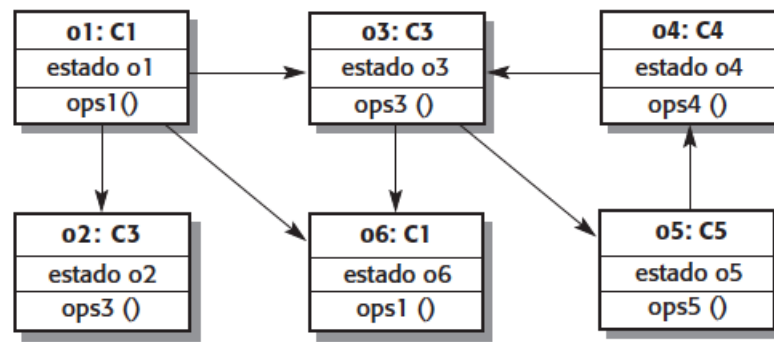


Figura 1 – Um sistema constituído de objetos que interagem entre si.

Fonte: [Sommerville 2003]

Como afirma Sommerville 2003, projeto orientado a objetos é parte do desenvolvimento orientado a objetos, em que uma estratégia orientada a objetos é utilizada em todo o processo de desenvolvimento:

- A **análise orientada a objetos** se dedica a desenvolver um modelo orientado a objetos do domínio de aplicação. Os objetos identificados refletem entidades e operações que estão associadas com o problema a ser resolvido.
- O **projeto orientado a objetos** se dedica a desenvolver um modelo orientado a objetos de um sistema de software para implementar os requisitos identificados. Os objetos em um projeto orientado a objetos estão relacionados à solução do problema que está sendo resolvido. É possível que haja relacionamentos próximos entre alguns objetos do problema e alguns objetos da solução, mas o projetista, inevitavelmente, precisa adicionar novos objetos e transformar objetos do problema, a fim de implementar a solução.
- A **programação orientada a objetos** se ocupa de realizar um projeto de software utilizando uma linguagem de programação orientada a objetos, por exemplo, a linguagem Java, que aceita a implementação direta de objetos e fornece recursos para definir as classes de objetos.

A transição entre esses estágios de desenvolvimento deve ser contínua e direta, com a mesma notação utilizada em cada estágio. Mover para o próximo estágio envolve aprimorar o estágio anterior, adicionando detalhes às classes existentes de objetos e inventando novas classes, a fim de oferecer funcionalidade adicional.

1.2 Classes e objeto

Para [Sommerville 2003] ‘objeto’ e ‘orientado a objetos’ são amplamente utilizados e aplicados a diferentes tipos de entidades, métodos de projeto, sistemas e linguagens de programação. Contudo, existe uma aceitação geral de que um objeto é um encapsulamento de informações, e isso se reflete na definição de um objeto e de uma classe de objeto a seguir:

- Um objeto é uma entidade que possui um estado e um conjunto definido de operações que operam nesse estado. O estado é representado por um conjunto de atributos de objeto. As operações associadas com o objeto fornecem serviços para outros objetos (clientes), que requisitam esses serviços quando alguma computação é necessária.
- Os objetos são criados de acordo com uma definição de classe de objetos que serve como um template para criar objetos. Essa classe apresenta declarações de todos os atributos e operações que devem ser associados a um objeto dessa classe.

Em outras palavras, podemos dizer que classe é uma descrição generalizada que descreve uma coleção de objetos similares, o qual, segundo Pressman e Maxim 2016 é um conceito orientado a objeto que encapsula dados e abstrações procedurais necessárias para descrever o conteúdo e comportamento de alguma entidade do mundo real.

Exemplos de objetos são: os objetos físicos (um livro, uma caneta), funções de pessoas para os sistemas (funcionário, cliente), eventos (uma compra, um telefonema), interações entre outros objetos (um item de uma nota fiscal é uma interação entre uma compra e um produto do estoque) e lugares (loja matriz, revenda nordeste).

Para fins de estudo, e com objetivo mais didático, usaremos um cachorro como nosso “objeto”:

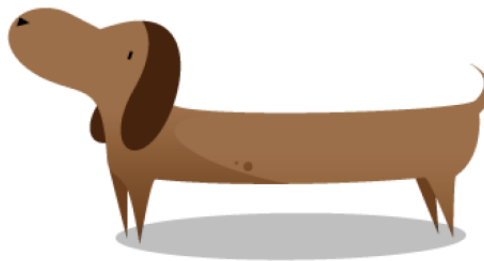


Figura 2 – Representação de um objeto

Ao analisar o objeto, deduz-se que há características pertencentes somente a ele. Tais como:

- Nome;

- Idade;
- Comprimento de pelos;
- Cor dos pelos;
- Cor dos olhos;
- Peso, entre outros;

Tais características que descrevem um objeto são chamadas na orientação a objeto de atributos.

1.3 Atributos

Os objetos do mundo real tem propriedades que por sua vez possuem valores. Estes valores determinam o estado do objeto. Assim, na orientação a objeto, essas propriedades são chamadas de atributos. Logo, podemos dizer que esses atributos dos objetos são semelhantes a variáveis ou campos que guardam os variados valores que os objetos podem receber como características. **O estado de um objeto é um grupo de valores que estão em seus atributos em um certo momento.**



Figura 3 – Representação de um objeto

Cachorro	
Nome:	Hubert
Idade:	2 anos
Tipo Pelo:	Curtos
Cor dos pelos:	Marrom
Cor dos olhos:	Castanhos
Peso	5kg

Tabela 1 – Atributos e valores

Outro objeto cachorro teria valores diferentes para estes mesmos atributos, como exemplo disto temos:

**Figura 4** – Representação de um objeto

Cachorro	
Nome:	Floks
Idade:	4 anos
Tipo pelo:	Curtos
Cor dos pelos:	Branco
Cor dos olhos:	Castanhos
Peso	5kg

Tabela 2 – Atributos e valores

Para que os atributos de um objeto mudem de valor isso deve ser feito exclusivamente por estímulos externos ou internos. Assim, a única maneira de alterar os atributos dos objetos é disparando eventos que geram a mudança desses estados no objeto.

1.4 Métodos

São procedimentos ou funções que executam as ações específicas do objeto. Dessa forma, os métodos são as ações que o objeto é capaz de realizar. É através dos métodos que o objetos faz tudo, inclusive se manifesta e interage com outros objetos.

Um objetos expõe algum comportamento (ou seja, executa uma operação) ao receber um estímulo de outro objeto. É através de mensagens que um objeto requisita uma ação a outro objeto. Sendo esta mensagem uma solicitação a um objeto para que sejam executadas as rotinas a qual são nomeadas de Método da classe. Então os métodos assumem a responsabilidade de acessar ou alterar os atributos de um objeto.

Anteriormente no estudo do objeto cachorro foi enumerado uma série de métodos (ações), tais como: latir, babar, comer, sentar, etc.

1.5 Herança

O conceito de herança é um dos conceitos fundamentais de POO. Herança, na prática, significa a possibilidade de construir objetos especializados que herdaram as características de objetos mais generalistas, ou ainda, a herança uma maneira de reutilizar código a medida que podemos aproveitar os atributos e métodos de classes já existentes para gerar novas classes mais específicas que aproveitarão os recursos da classe hierarquicamente superior [Ruiz 2008].

O conceito de herança mimetiza as características hierárquicas de vários sistemas reais, como por exemplo, os sistemas de classificação em biologia que, pode determinar como uma hierarquia o seguinte:

- animais;
- vertebrados e invertebrados;
- mamíferos e aves;
- entre outras características mais específicas

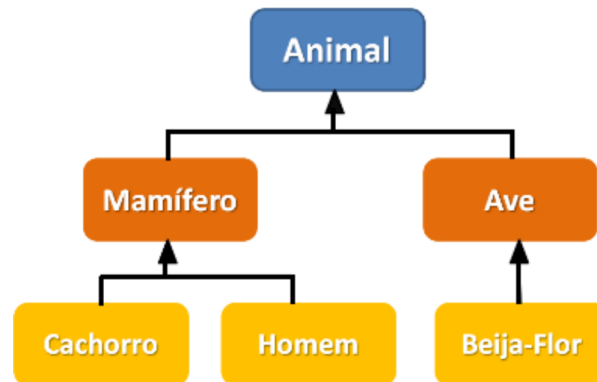


Figura 5 – Diagrama representando a herança entre as classes

1.5.1 Superclasses e subclasses

Em POO todo objeto de uma classe construída pelo usuário da linguagem é também um objeto de outra classe. Por exemplo, na hierarquia da área da saúde, podemos dizer que pessoa é uma superclasse e que empregados é uma subclasse de pessoa.

Outra nomenclatura que é utilizada para especificar superclasses ou subclasses é a Generalização ou Especialização. No exemplo acima, pessoa é a generalização de empregado, e empregado é a especialização de pessoa conforme representado na Figura 6

Uma máxima que podemos guardar é:

Uma subclasse guarda a relação é um com a superclasse.

**Figura 6** – Superclasses e Subclasses

Fonte: [Silva et al. 2009]

1.6 Exercícios

1. Para satisfazer as necessidades de informatização de uma biblioteca universitária um sistemas foi proposto para satisfazer algumas características:
 - Cadastro dos usuários da biblioteca com endereço completo. Usuário são classificados em três grupos: professores, alunos e funcionário.
 - Cadastro das obras da biblioteca são classificados em: livros científicos, periódicos científicos, periódicos informativos, periódicos diversos, entretenimento, etc.
 - Linguagem usada no exemplar da obra.
 - Mídia que armazena o exemplar da obra.
 - Autores da obra com o controle da nacionalidade dos mesmos.
 - Editoras dos exemplares com ano de edição referente a cada exemplar.

Identifique os possíveis objetos com seus atributos e métodos respectivos.

2. **Desafio:** Pesquise sobre os pontos negativos da orientação a objeto, principalmente sobre os conceitos de Coesão e Acoplamento.

Referências

- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016. Citado na página 4.
- RUIZ, E. E. S. *IBM1030 Programação Orientada a Objetos*. 2008. Disponível em: <http://dcm.ffclrp.usp.br/~evandro/ibm1030/constru/heranca.html>. Citado na página 6.
- SILVA, D. Lucas da et al. Ontologias e unified modeling language: uma abordagem para representação de domínios de conhecimento. v. 10, 10 2009. Citado na página 8.
- SOMMERVILLE, I. *Engenharia de Software. 6ª*. [S.l.: s.n.], 2003. Citado 3 vezes nas páginas 2, 3 e 4.