

CURSO: Ciência da Computação
PERÍODO: 4º.
DISCIPLINA: Técnicas Alternativas de Programação

DATA: ____/____/ 2013
PROFESSOR: Andrey
AULA: 04

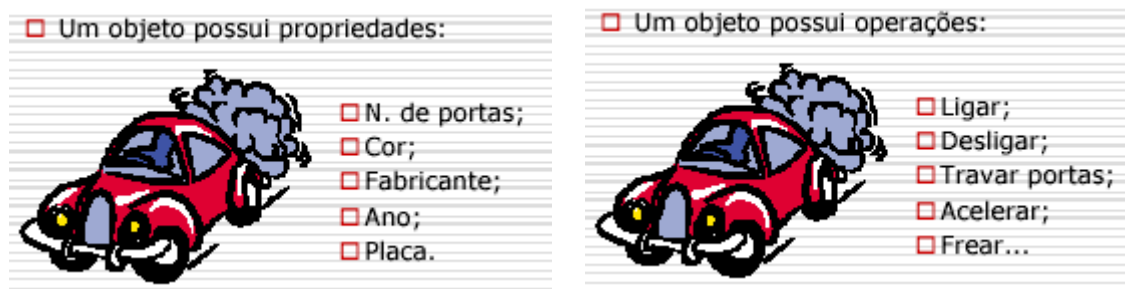
APRESENTAÇÃO:

Hoje vamos discutir os conceitos de Orientação a Objetos.

DESENVOLVIMENTO:

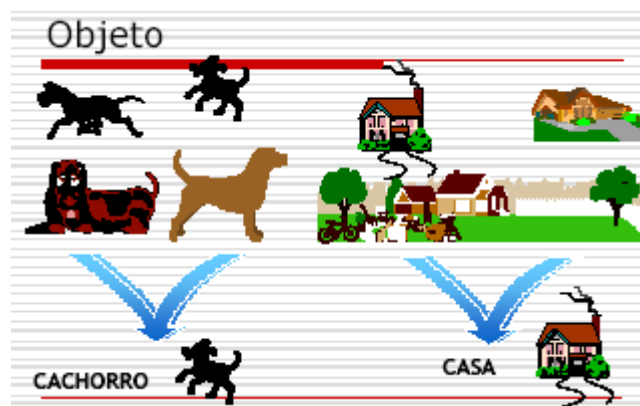
Orientação a Objetos (OO)

É uma forma de entender e representar sistemas complexos como estruturas hierárquicas de objetos correlatos.



Objeto

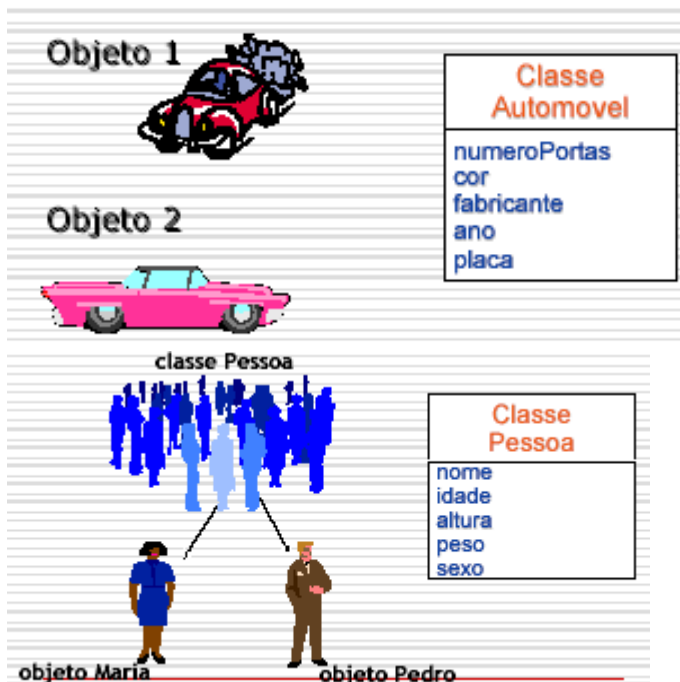
- Um objeto é uma construção de software que encapsula estado e comportamento, através de propriedades (atributos) e operações (métodos);
- Estado de um objeto: composto por suas propriedades e seus respectivos valores;
- Comportamento: a maneira como o objeto reage quando o seu estado é alterado ou quando uma mensagem é recebida.



CLASSES

- Conjunto de objetos similares.
 - Estrutura de dados similares (propriedades);
 - Comportamento similar (operações)
- Um objeto é uma instância de uma classe;

- Objetos de uma mesma classe diferenciam-se pelos valores de suas propriedades e de seus identificadores únicos.

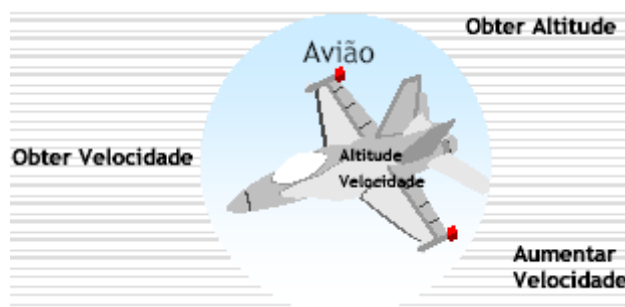


Troca de Mensagens

- Mecanismo através do qual os objetos se comunicam, invocando as operações desejadas;
- Um objeto (Emissor) envia uma mensagem a outro (Receptor) que executará uma tarefa;
- Operações (métodos) são invocados através de mensagens.

Encapsulamento

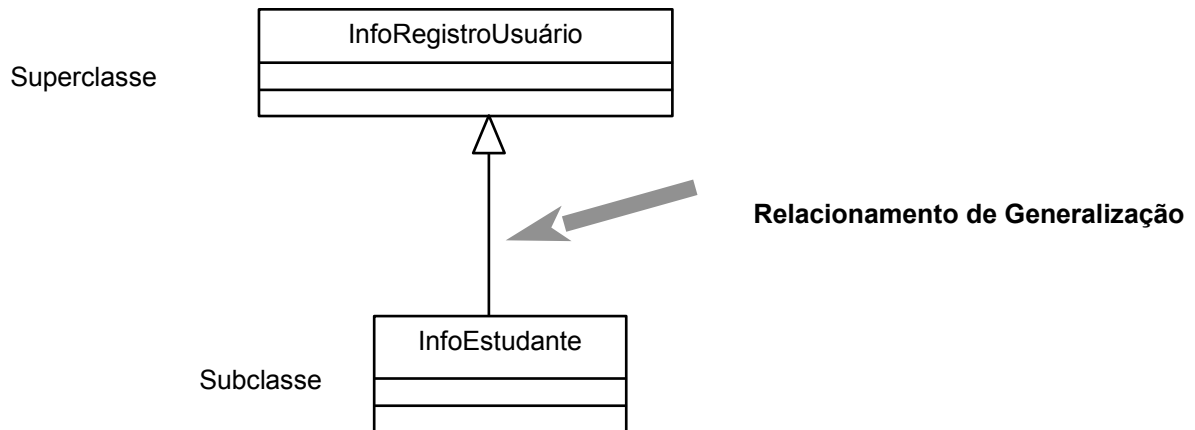
- “Não mostre as cartas de seu baralho”
- Objetivo: Ocultar do mundo externo ao objeto, os detalhes de implementação e restringir o acesso às propriedades e métodos.
- Vantagens:
 - Segurança no acesso ao objeto;
 - Melhor consistência no estado interno, pois evita alterações incorretas de valores das propriedades.



Herança

- **Generalização** define um relacionamento entre classes onde uma classe compartilha a estrutura e/ou comportamento de uma ou mais classes
- Generalização define uma hierarquia de abstrações na qual uma subclasse herda de uma ou mais superclasses
 - Com **herança simples**, a subclasse herda de apenas uma superclasse
 - Com **herança múltipla**, a subclasse herda de mais de uma superclasse
- Generalização é um relacionamento "é um" ou "tipo de"

Desenhando uma Hierarquia de Herança



Considerações sobre generalização

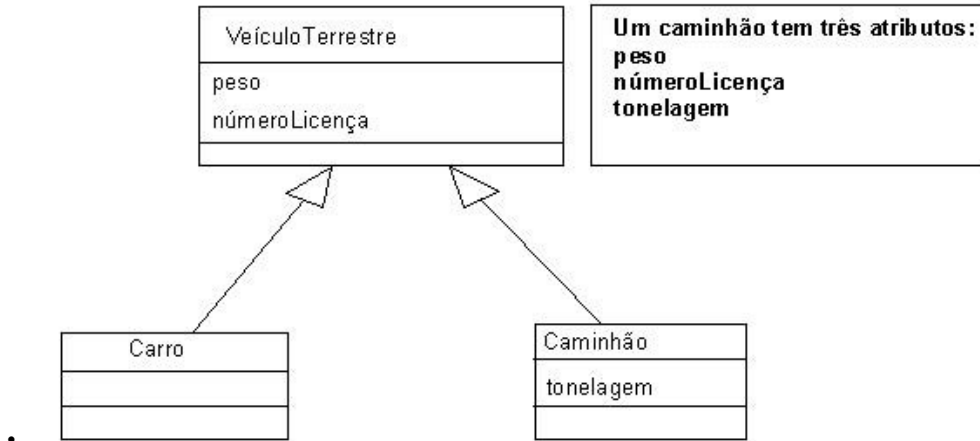
- Como um relacionamento de generalização não se refere a objetos individuais
 - O relacionamento não é nomeado
 - Multiplicidade não tem sentido
- Teoricamente, não há limite no número de níveis em uma hierarquia
 - Na prática, os níveis precisam ser bem limitados
 - Hierarquias típicas em C++ tem 3 a 5 níveis
 - Hierarquias em Smalltalk podem ser um pouco maiores

O que é Herdado?

- Uma subclasse herda de seus pais:
 - Atributos
 - Operações
 - Relacionamentos
- Uma subclasse pode:
 - Incluir atributos, operações e relacionamentos adicionais
 - Redefinir as operações herdadas (use com cautela!)

Herdando Atributos

- Atributos são definidos no nível mais alto da hierarquia de herança na qual eles são aplicáveis
- Subclasses de uma classe herdam todos os atributos
- Cada subclasse pode adicionar novos atributos



Herdando Operações

- Operações são definidas no nível mais alto da hierarquia de herança na qual elas são aplicáveis
- Subclasses de uma classe herdam todas as operações
- Cada subclasse pode aumentar ou redefinir operações herdadas

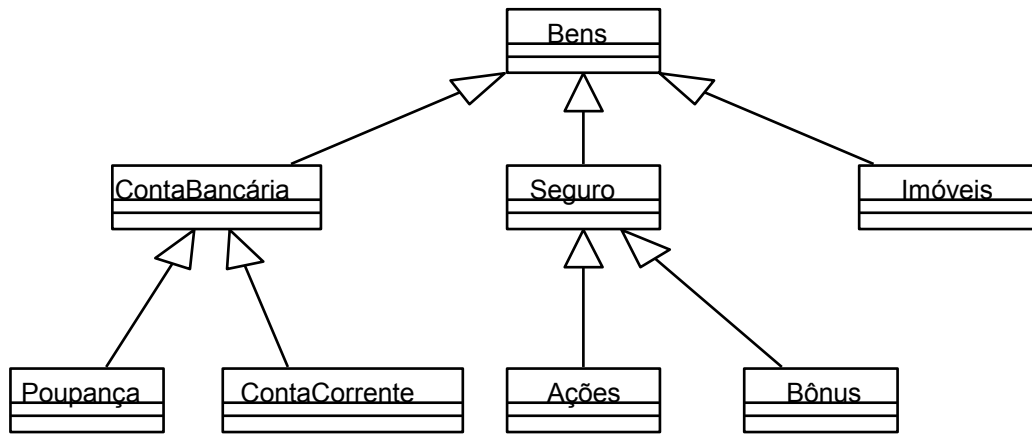
Herdando Relacionamentos

- Relacionamentos também são herdados e devem ser definidos no nível mais alto da hierarquia de herança na qual eles são aplicáveis
- Subclasses de uma classe herdam todos os relacionamentos
- Cada subclasse pode também possuir relacionamentos adicionais

Generalização de Classes

- Generalização proporciona a capacidade de criar superclasses que reúnem estrutura e/ou comportamento comum a várias subclasses
- Procedimento de generalização
 - Identificar similaridades de estrutura/comportamento entre várias classes
 - Criar uma superclasse para reunir a estrutura/comportamento comum
 - As classes originais passam a ser subclasses da nova superclasse
- Superclasses são mais abstratas que suas subclasses

Exemplo de Generalização



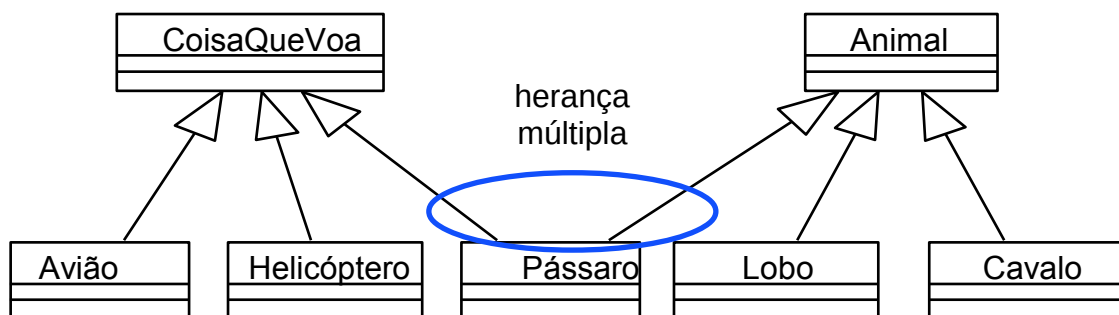
Especialização de Classes

- A especialização proporciona a capacidade de criar subclasses que representam refinamentos nos quais a estrutura e/ou comportamento da superclasse são adicionados ou modificados
- Procedimento de Especialização
 - Notar que algumas instâncias apresentam estrutura ou comportamento especializado
 - Criar subclasses para agrupar instâncias de acordo com sua especialização
- Subclasses são menos abstratas que suas superclasses

Hierarquias de Herança

- Tanto generalização quanto especialização são usadas no desenvolvimento de uma hierarquia de herança
- Durante a análise, são estabelecidas hierarquias de herança entre abstrações chaves (i.e., classes) se necessário
- Durante o projeto, as hierarquias de herança são refinadas para:
 - Aumentar reutilização
 - Incorporar classes de implementação
 - Incorporar bibliotecas de classes disponíveis

Herança Múltipla



Conceitos de Herança Múltipla

- Conceitualmente é necessário para modelar o mundo real de forma precisa
- Na prática, isto pode gerar dificuldades na implementação
 - Nem todas as linguagens de programação orientadas a objetos suportam herança múltipla diretamente

Cada ambiente/linguagem de programação escolhe maneiras de resolver estas dificuldades.

INTEGRAÇÃO:

BIBLIOGRAFIA:

FURGERI, SÉRGIO. *Java 2 Ensino Didático. Desenvolvendo e Implementando Aplicações*. ed. Érica. São Paulo, 2002.
DEITEL, H. M. e DEITEL, P. J.. *Java, como Programar*. Ed. Bookman. Porto Alegre. 2001.
ARNOLD Ken, GOSLING James: "The Java Programming Language Second Edition", Addison-Wesley, 1997.