Introdução à Orientação a Objetos

SSC 121 - Engenharia de Software I

Profa. Dra. Elisa Yumi Nakagawa

2º semestre de 2012

Conteúdo

- Histórico de OO
- Vantagens de OO
- Linguagens OO
- Conceitos Básicos de OO
- Conclusão

Histórico de OO

- A OO surgiu no final da década de **60**, quando dois cientistas dinamarqueses criaram a linguagem Simula (*Simulation Language*)
- **1967** - Linguagem de Programação Simula-67- *conceitos de classe e herança*
- O termo Programação Orientada a Objeto (POO) é introduzido com a linguagem Smalltalk (**1983**)
- FINS DOS ANOS **80** ⇨ Paradigma de Orientação a Objetos
  - abordagem poderosa e prática para o desenvolvimento de software

Histórico de OO

- Surgiram linguagens híbridas:
  - C++ (**1986**), Object-Pascal (**1986**)
- Surgiram diversos Métodos/Técnicas de Análise e Projeto OO
  - CRC (*Class Responsability Collaborator*, Beecke e Cunningham, **1989**)
  - OOA (*Object Oriented Analysis*, Coad e Yourdon, **1990**)
  - Booch (**1991**)
  - OMT (*Object Modeling Technique*, Rumbaugh, **1991**)
  - Objectory (Jacobson, **1992**)
  - Fusion (Coleman, **1994**)
  - UML (*Unified Modeling Language*, **1997**)

Vantagens de OO

- abstração de dados: os detalhes referentes às representações das classes serão visíveis apenas a seus atributos;
- compatibilidade: as heurísticas para a construção das classes e suas interfaces levam a componentes de software que são fáceis de se combinar;
- flexibilidade: as classes delimitam-se em unidades naturais para a alocação de tarefas de desenvolvimento de software;

Vantagens de OO

- reutilização: o encapsulamento dos métodos e representação dos dados para a construção de classes facilitam o desenvolvimento de software reutilizável, auxiliando na produtividade de sistemas;
- extensibilidade: facilidade de estender o software devido a duas razões:
  - herança: novas classes são construídas a partir das que já existem;
  - as classes formam uma estrutura fracamente acoplada o que facilita alterações;
- manutenibilidade: a modularização natural em classes facilita a realização de alterações no software.

Vantagens de OO

- melhora de comunicação entre desenvolvedores e clientes;
- redução da quantidade de erros no sistema, diminuindo o tempo nas etapas de codificação e teste;
- maior dedicação à fase de análise, preocupando-se com a essência do sistema;
- mesma notação é utilizada desde a fase de análise até a implementação.

Frente a essas vantagens, a tecnologia de OO tem provado ser "popular" e eficaz.

Linguagens OO

- Existem diversas linguagens OO, tais como:
  - Smalltalk (1972)
  - Ada (1983)
  - Eiffel (~1985)
  - Object Pascal (1986)
  - Common Lisp (1986)
  - C++ (~1989)
  - Java

Conceitos Básicos

- Orientação a Objetos (OO) é uma abordagem de programação que procura explorar nosso lado intuitivo. Os objetos da computação são análogos aos objetos existente no mundo real.
- No enfoque de OO, os átomos do processo de computação são os objetos que trocam mensagens entre si.
- Essas mensagens resultam na ativação de métodos, os quais realizam as ações necessárias.
- Os objetos que compartilham uma mesma interface, ou seja, respondem as mesmas mensagens, são agrupados em classes.

Conceitos Básicos

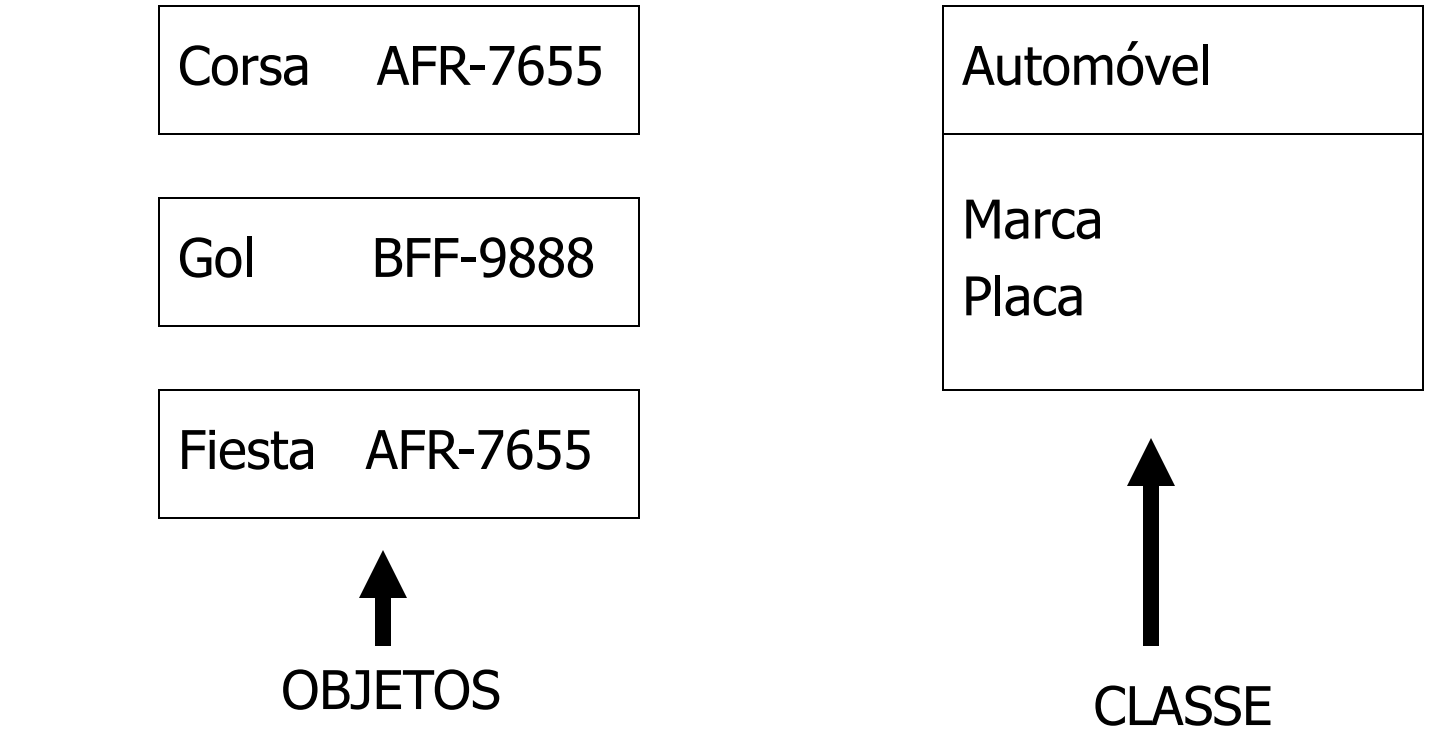
- Observe que:
  - sistema orientado a objetos (Estado **dinâmico** durante a execução)
  - sistema orientado a função (Estado **estático** durante a execução)

Conceitos Básicos

- Objeto é algo **DINÂMICO**: é criado por alguém, tem uma vida, e morre ou é morto por alguém. Assim, durante a execução do sistema, os objetos podem:
  - ser construídos
  - executar ações
  - ser destruídos
  - tornar inacessíveis

Conceitos Básicos

- Objetos e Classes



Conceitos Básicos

- Objetos:
  - Tudo em OO é OBJETO
  - Objeto, no mundo físico, é tipicamente um produtor e consumidor de itens de informação
  - Definição (mundo do sofware)
    - *"Qualquer coisa, real ou abstrata, a respeito da qual armazenamos dados e métodos que os manipulam"* Martin, Odell (1995)
    - Abstração de uma entidade do mundo real de modo que essa entidade possue várias características

Conceitos Básicos

- Classes:
  - Agrupamento de objetos similares.
  - Todo objeto é uma instância de uma Classe.
  - Os objetos representados por determinada classe diferenciam-se entre si pelos valores de seus atributos.
  - Conjunto de objetos que possuem propriedades semelhantes (ATRIBUTOS), o mesmo comportamento (MÉTODOS), os mesmos relacionamentos com outros objetos e a mesma semântica.

Conceitos Básicos

- Exemplo 1

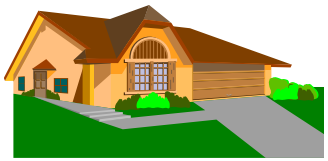
CASA

Portas
Quartos
Salas
Localização
Cozinha
Telhado

Reformar
Limpar
Pintar
Mobiliar

Casa do
Presidente Clinton

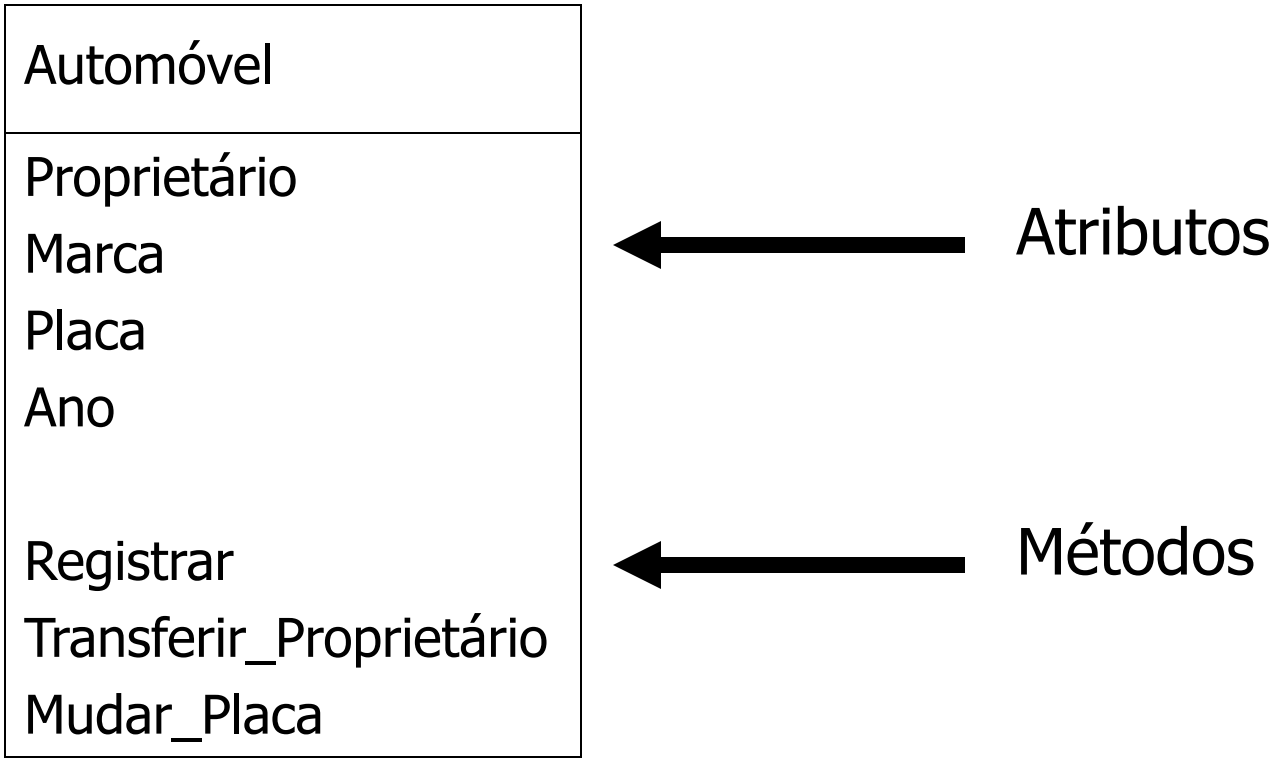
Casa do
Pedrinho



Casa Amarela
da Esquina

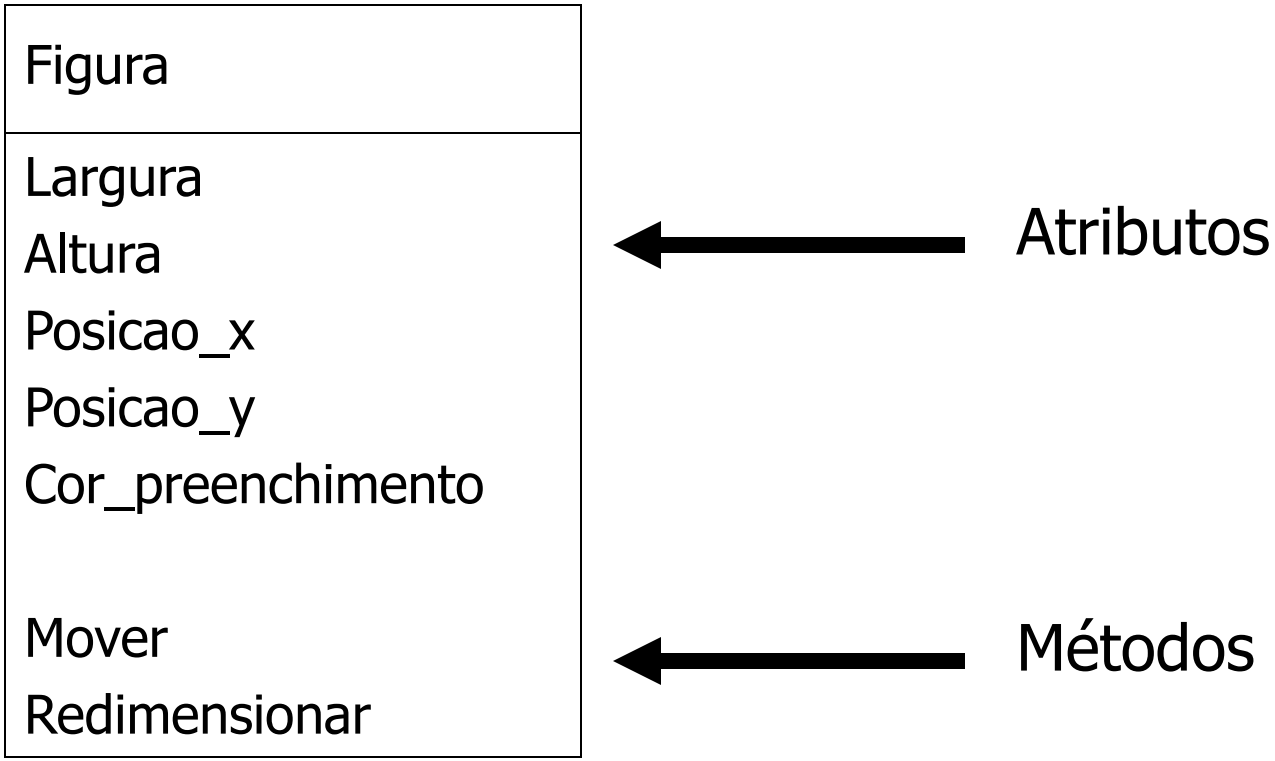
Conceitos Básicos

- Atributos e Métodos: Exemplo 2



Conceitos Básicos

- Atributos e Métodos: Exemplo 3



Conceitos Básicos

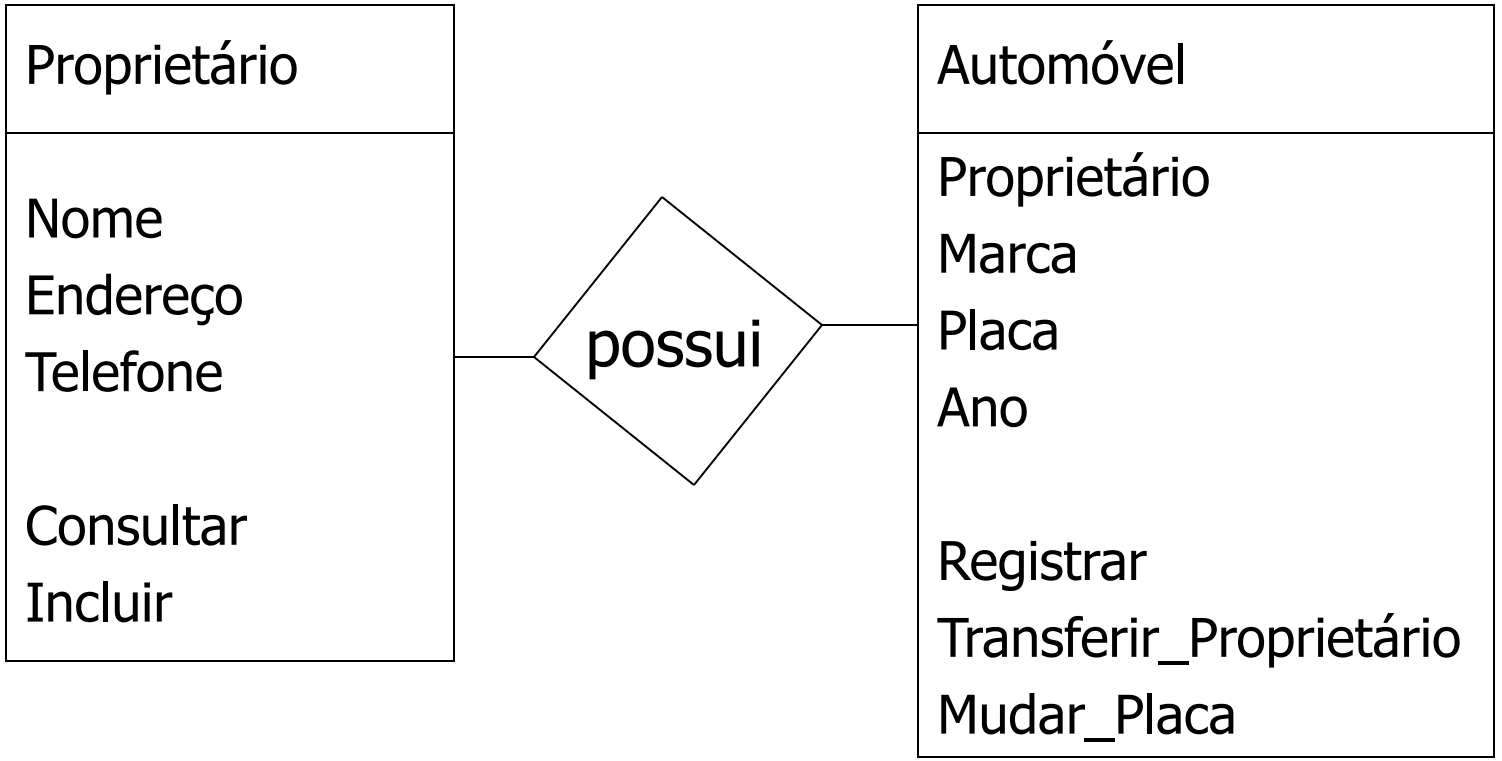
- Atributos:
  - Representam um conjunto de informações, ou seja, elementos de dados que caracterizam um objeto
  - Descrevem as informações que ficam escondidas em um objeto para serem exclusivamente manipulado pelas operações daquele objeto
  - São variáveis que definem o estado de um objeto, ou seja, são entidades que caracterizam os objetos
  - Cada objeto possue seu próprio conjunto de atributos

Conceitos Básicos

- Métodos:
  - Quando um objeto é mapeado dentro do domínio do software, os processos que podem mudar a sua estrutura de dados são denominados Operações ou Métodos
  - Métodos são invocados por Mensagens
  - Cada objeto possue seu próprio conjunto de métodos
  - Definições: São procedimentos definidos e declarados que atuam sobre um objeto ou sobre uma classe de objetos

Conceitos Básicos

- Relacionamento entre Classes



Conceitos Básicos

- Descrição da Classe Automóvel

```
classe Automóvel
    atributos proprietário: seqüência de caracteres
    atributos marca: seqüência de caracteres
    atributos placa: seqüência de caracteres
    atributo ano: inteiro
    método Registrar ()
    método Transferir_Proprietário ()
    método Mudar_Placa ()
fimclasse
```

Conceitos Básicos

- Três elementos chaves de OO são:
  - encapsulamento
  - herança
  - polimorfismo

Conceitos Básicos

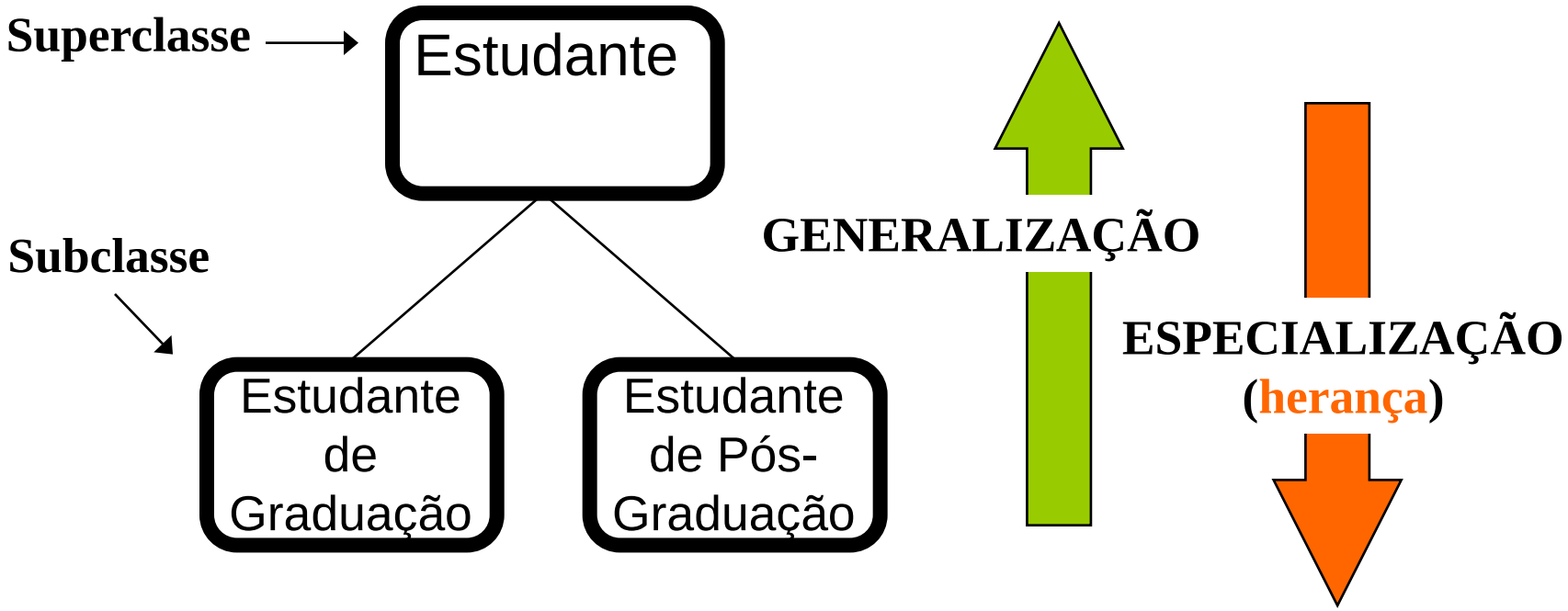
- Encapsulamento:
  - objetos encapsulam seus atributos;
  - propriedade segundo a qual os atributos de uma classe são acessíveis apenas pelos métodos da própria classe;
  - outras classes só podem acessar os atributos de uma classe invocando os métodos públicos;
  - restringe a visibilidade do objeto mas facilita o reuso
  - os DADOS e os MÉTODOS são empacotados sob um nome e podem ser reusados como uma especificação ou componente de programa.

Conceitos Básicos

- Herança:
  - A herança é o mecanismo pelo qual uma subclasse herda todas as propriedades da superclasse e acrescenta suas próprias e exclusivas características.
  - As propriedades da superclasse não precisam ser repetidas em cada subclasse.
  - Por exemplo, *JanelaRolante* e *JanelaFixa* são subclasses de *Janela*. Elas herdam as propriedades de Janela, como uma região visível na tela. *JanelaRolante* acrescenta uma barra de paginação e um afastamento.

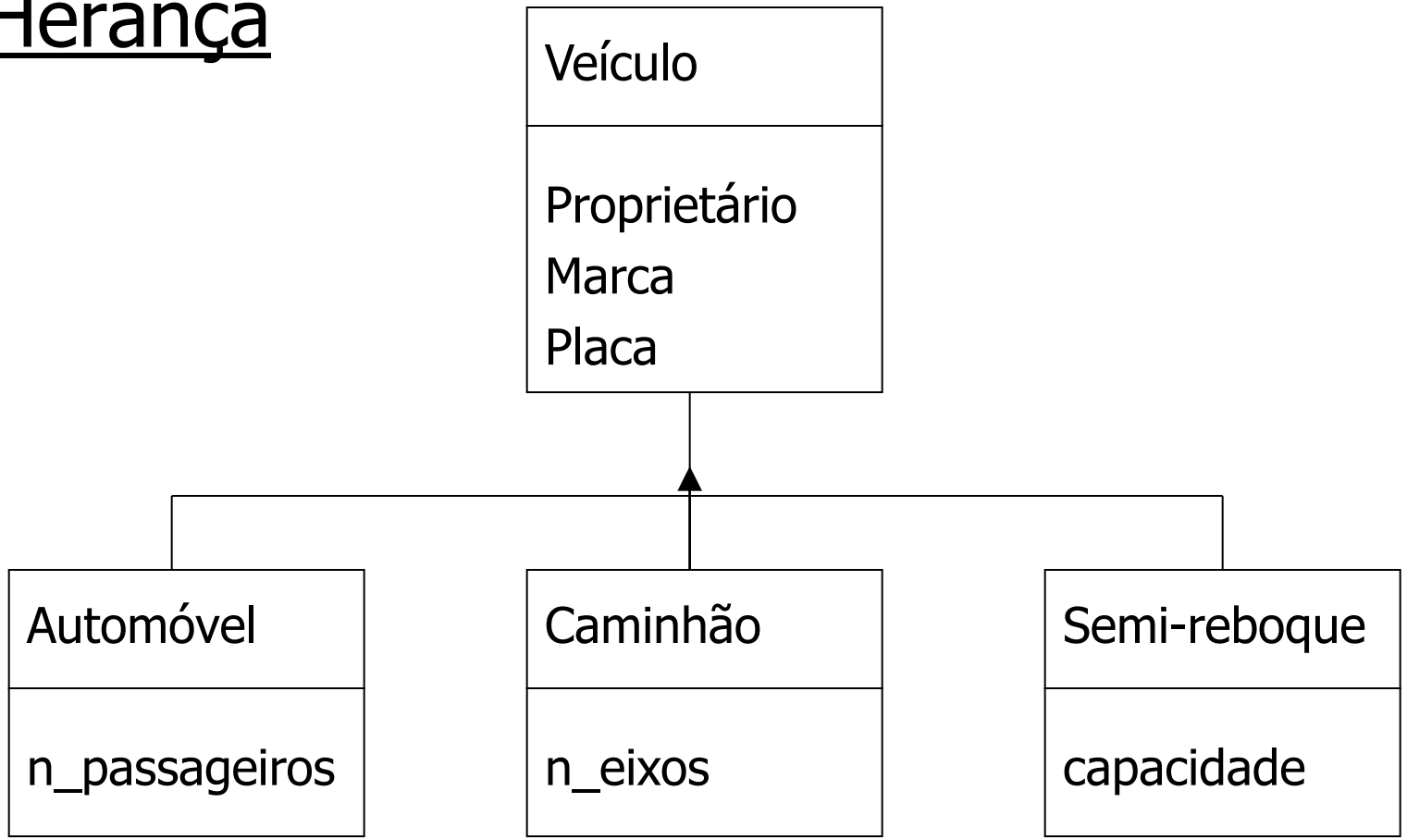
Conceitos Básicos

- Generalização/Especialização



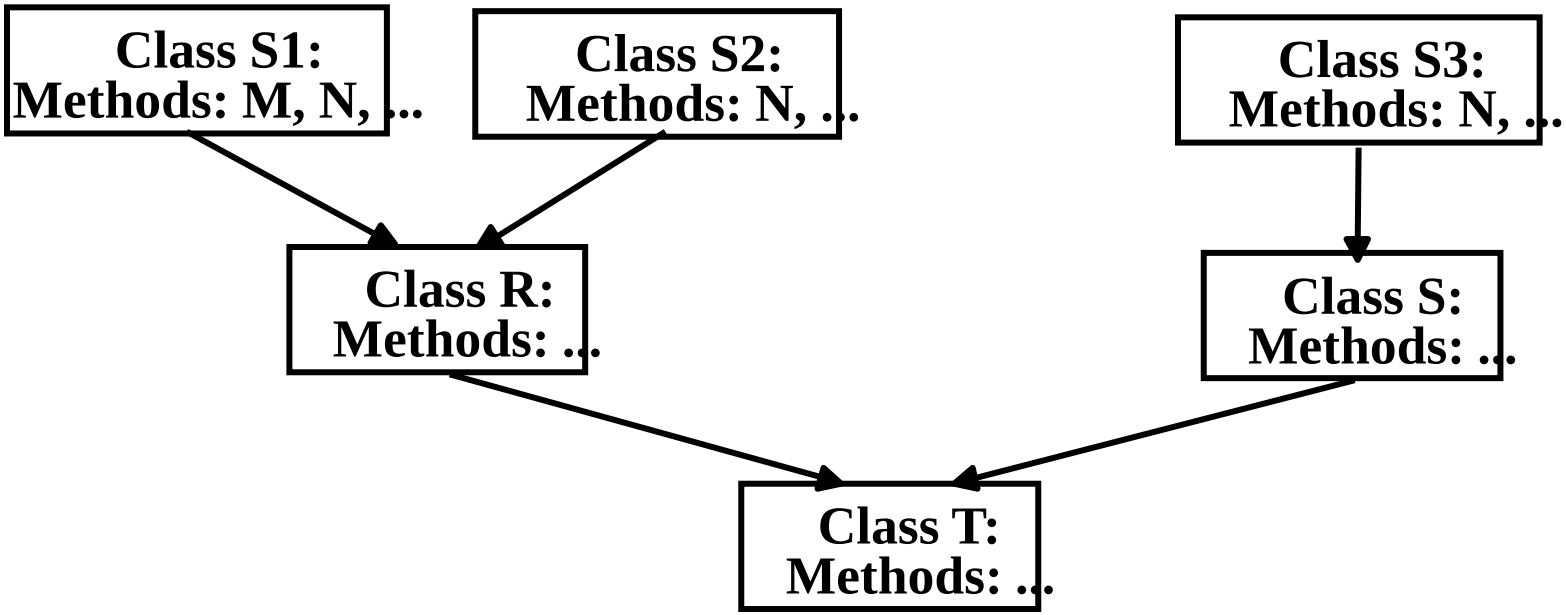
Conceitos Básicos

- Herança



Herança e Herança Múltipla

Cada classe é declarada como uma subclasse de uma ou mais superclasses. Quando existe mais de uma superclasse, a relação de herança é denominada herança múltipla.



Polimorfismo

- O Polimorfismo geralmente representa a qualidade ou estado de um objeto ser capaz de assumir diferentes formas.
- Mais especificamente, propriedade segundo o qual vários métodos podem existir com o mesmo nome.
  - Ao receber uma mensagem para efetuar uma Operação, é o objeto quem determina como a operação deve ser efetuada;
  - Permite a criação de várias classes com interfaces idênticas, porém objetos e implementações diferentes.
- Exemplos:
  - O operador "+" pode ser usado com inteiros, pontos-flutuantes ou *strings*.
  - A operação *mover* pode atuar diferentemente nas classes Janela e PeçadeXadrez.

Conceitos Básicos

- Analogia dos conceitos principais no paradigma orientado a objeto e no paradigma tradicional de programação

Linguagens Orientadas a Objetos	Linguagens Tradicionais
Objeto	Valor
Classe	Tipo (TAD)
Mensagem	Chamada de Procedimento
Método	Procedimento ou Função
Interface	Conjunto de nomes e funções para um fim específico

Conclusão

- tecnologia de OO é bastante recente e "veio para ficar"
- OO impõe qualidade, produtividade e profissionalismo na construção de sistemas
- existem métodos, técnicas e ferramentas de software OO que acompanham o processo de desenvolvimento do software desde a análise até a implementação