Programação Orientada a Objetos

Turma 3

Carlos Henrique Bughi, MSc

UNIVALI
Escola Politécnica

Onde estamos?
(e para onde vamos)

- Paradigmas de programação
- Introdução à Orientação a Objetos
  - Pilares do POO
- UML

UNIVALI

Escola Politécnica

Aula 3

Pilares da Programação Orientada a Objetos

UNIVALI

Escola Politécnica

Os 4 Pilares da Programção Orientada a Objetos

UNIVALI

Escola Politécnica

A abstração

Abstração

- Construção de um modelo para representação de uma realidade
- Foco em aspectos essenciais de uma aplicação enquanto ignora os detalhes
- Preserva a simplicidade do projeto

Abstração

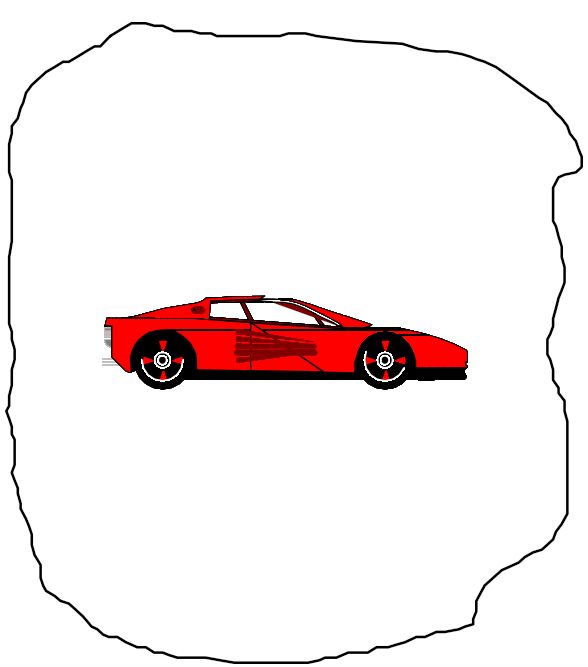
- Características essenciais de uma entidade que a distingue de todos os outros tipos de entidade.
- Define uma fronteira relativa à perspectiva do observador.
- Não é uma manifestação concreta, denota a essência ideal de alguma coisa.

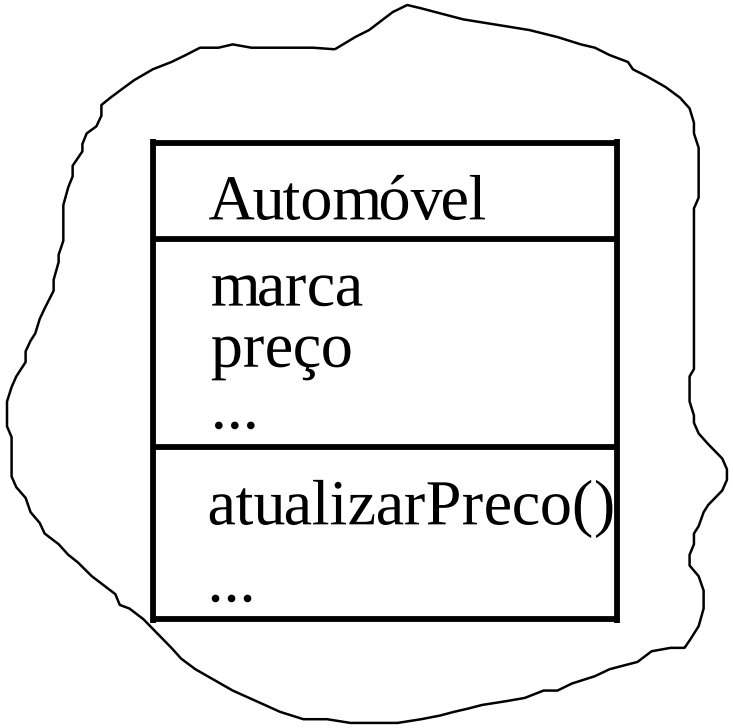
Processo de abstração

- Depende do ponto de vista de quem tem o problema para resolver.
- Um objeto pode ser abstraído de várias formas, dependendo do contexto do problema:
  - **Técnico em eletrônica**:
    - TV = transistores, resistores, capacitores, circuitos integrados.
  - **Telespectador**:
    - TV = ligar, desligar, trocar de canal, assistir.

Processo de abstração

- A representação do modelo pode ser feita através de uma linguagem de computador, diagramas ou até mesmo textos descritivos

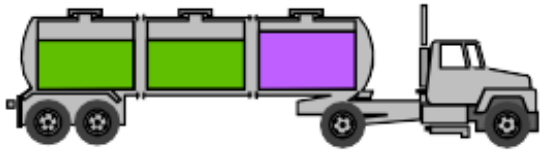




Construindo modelos (modelagem)

- Informalmente, um objeto representa um entidade que pode ser física, conceitual ou de software.

Entidade física

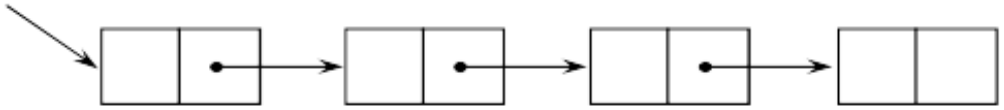


Caminhão

Entidade conceitual

Processo químico

Entidade de software



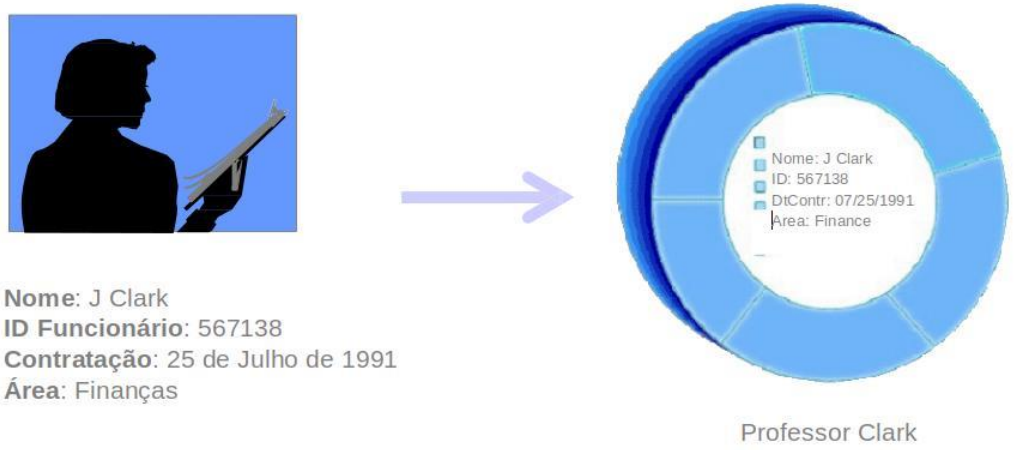
Lista ligada

Construindo modelos (modelagem)

- Em uma definição mais formal:
  - Um objeto é uma entidade com a fronteira bem definida e uma identidade que encapsula estado e comportamento.
    - Estado é representado por atributos e relacionamentos;
    - Comportamento é representado por métodos.

Construindo modelos (modelagem)

- Objetos possuem estado:
  - O estado de um objeto é a condição ou situação durante o ciclo de vida de um objeto o qual satisfaz algumas condições, executa alguma atividade ou aguarda um evento.
    - O estado de um objeto normalmente muda ao longo do tempo.



UNIVALI Escola Politécnica

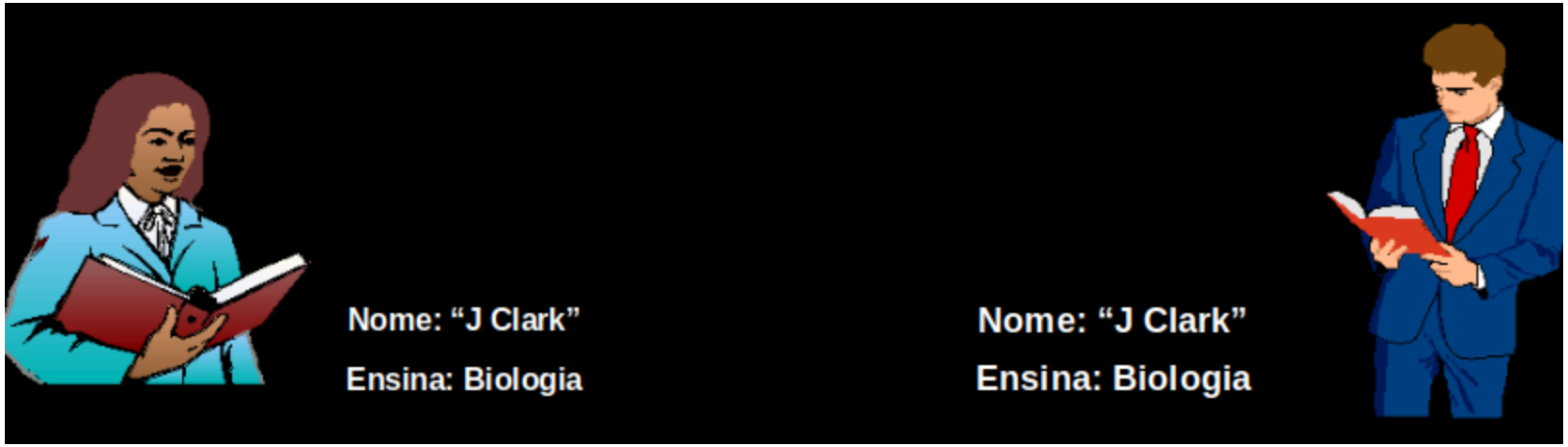
Construindo modelos (modelagem)

- Objetos possuem comportamento:
  - O comportamento determina como um objeto age e reage.
  - O comportamento observável de um objeto é modelado por um conjunto de mensagens que ele pode responder (ações que um objeto executa).

Comportamento do Professor Clark
Publicar notas finais
Aceitar oferta de curso
Corrigir notas

Construindo modelos (modelagem)

- Objetos possuem identidade:
  - Cada objeto possui uma identidade única, mesmo que o estado do objeto seja idêntico ao de outro objeto.



A Herança

UNIVALI

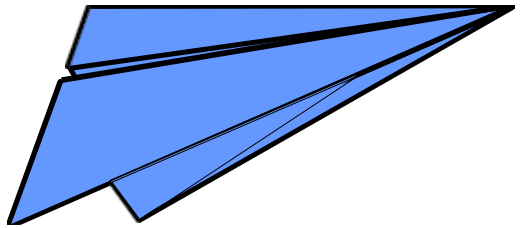
Escola Politécnica

Hierarquia

- Níveis de abstração (Pilar: Herança)

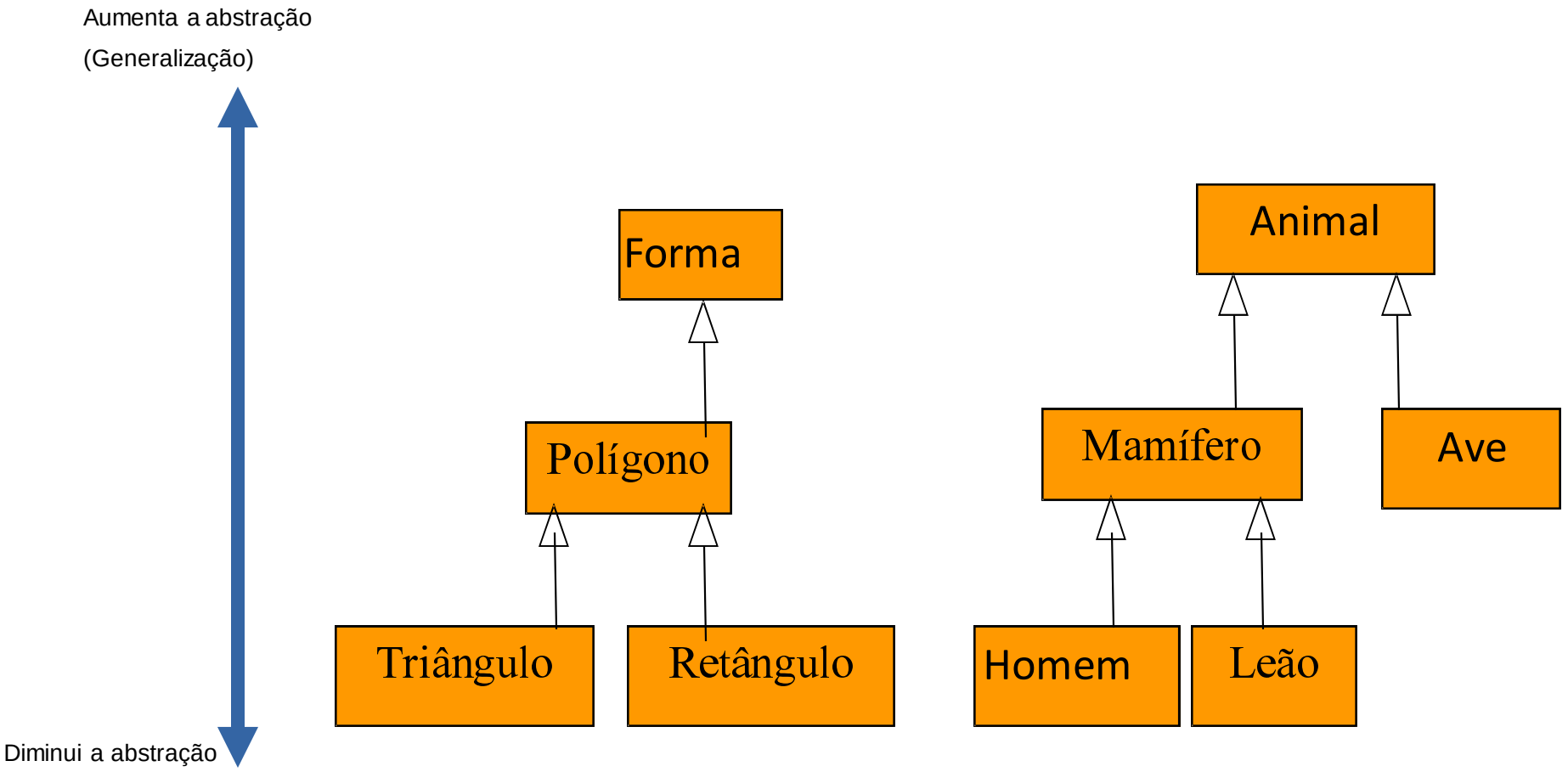
Mais abstrato
Mais genérico

Mais especializado
Mais detalhado



Hierarquia

- Níveis de abstração (Pilar: Herança)

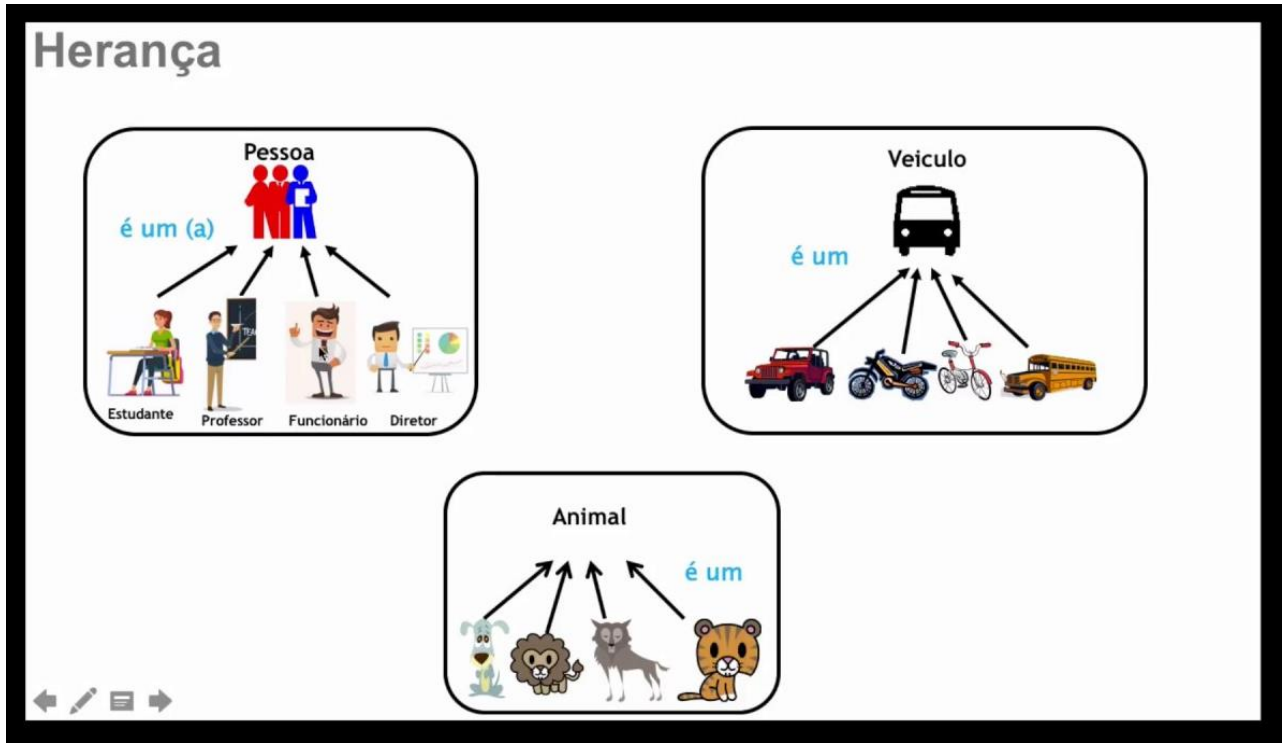


UNIVALI

Escola Politécnica

Hierarquia

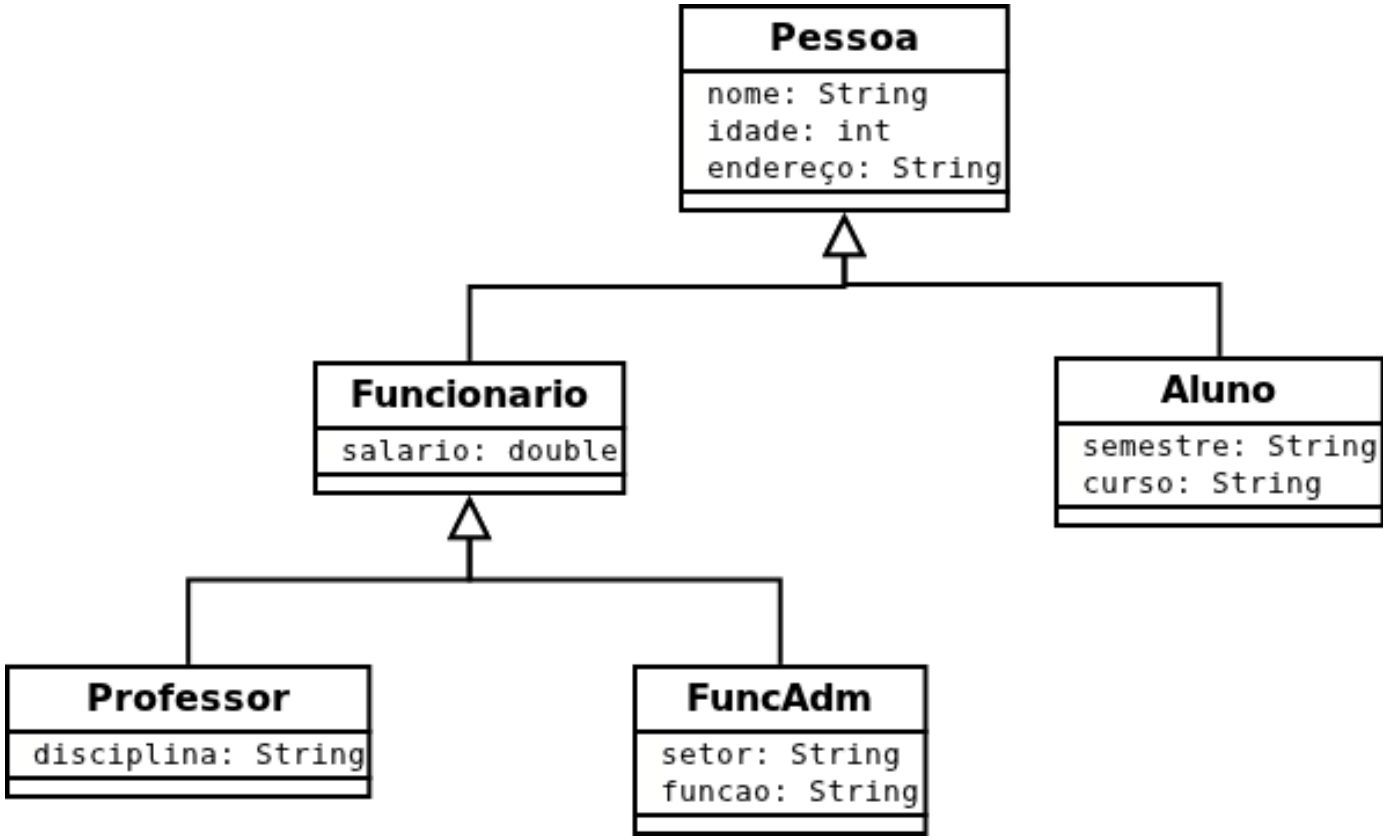
- Criação de níveis de abstração.
- Base conceitual para permitir a extensibilidade do software;
- Reuso de código e comportamento



Relação entre Hierarquia e Herança

- Herança permite modelar uma hierarquia entre classes.
  - Classes mais especializadas (subclasses ou classes filhas) herdam propriedades e métodos da classe mais geral (superclasse ou classe pai);
  - Assim, pode-se criar uma nova classe programando somente as diferenças desta para a classe pai (reuso).

Relação entre Hierarquia e Herança



O Encapsulamento

UNIVALI

Escola Politécnica

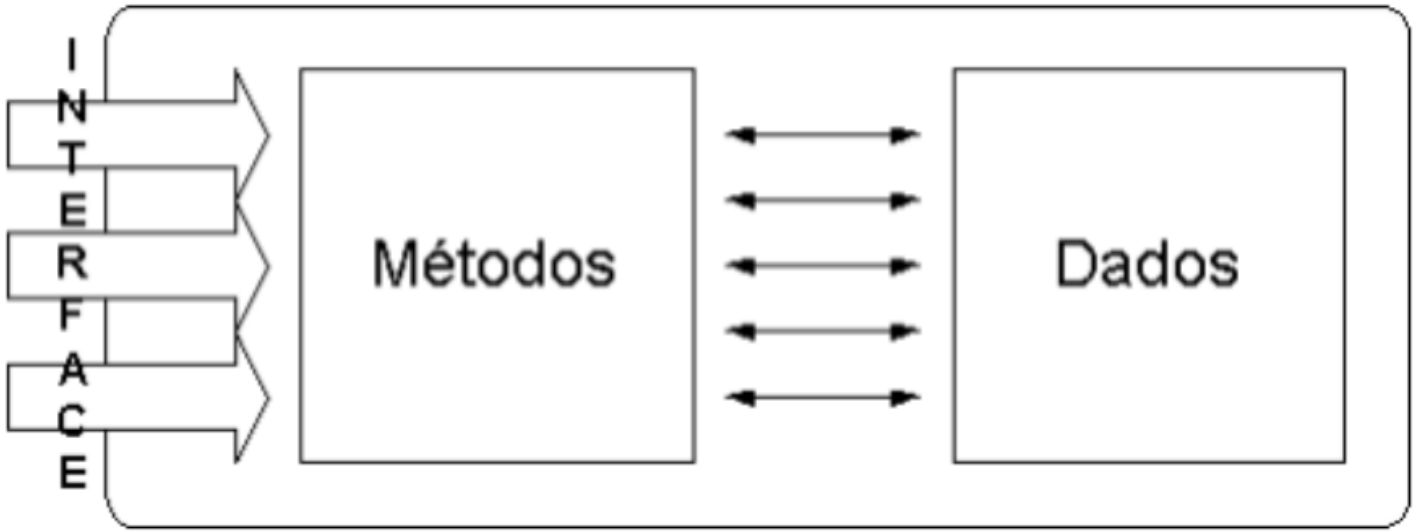
Encapsulamento

- O encapsulamento trata de dar alguma segurança aos objetos
- Esconde a implementação interna da especificação externa
  - Clientes conhecem somente a interface
  - Clientes dependem da interface e não da implementação

"Não posso quebrar o que não posso acessar"

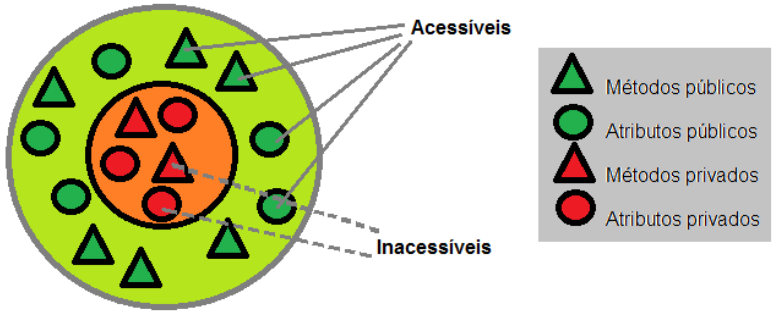
Encapsulamento

- Oferece os seguintes tipos de proteção
  - O estado interno de um objeto não pode ser corrompido por um cliente.
  - Mudanças internas não tem impacto sobre os clientes



Encapsulamento

- Proteger atributos através de métodos
  - Os atributos devem ser privados (ocultos)
  - Os métodos manipuladores devem ser públicos.



O Polimorfismo

UNIVALI

Escola Politécnica

Polimorfismo

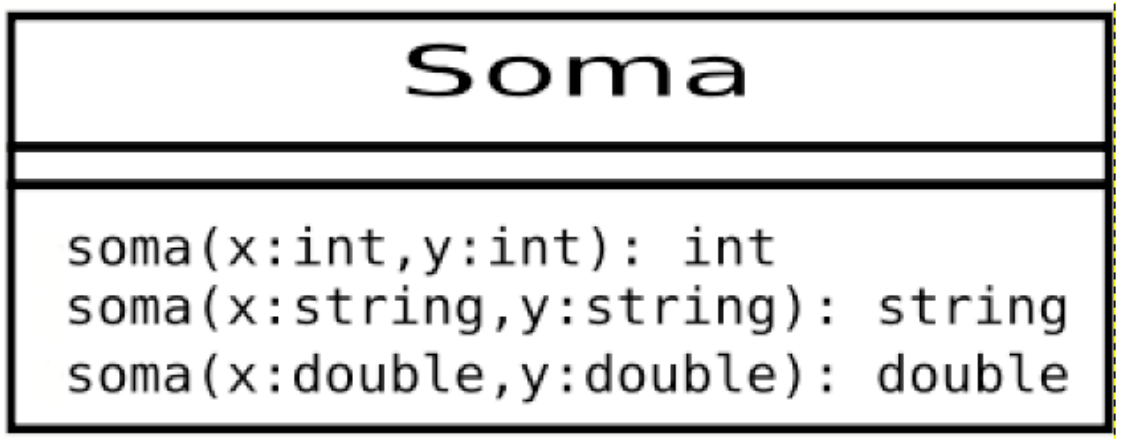
- De forma genérica, polimorfismo significa “várias formas”
- O polimorfismo não é um pensamento novo para nós, afinal, ele está contido em nosso dia a dia, principalmente na linguagem. Veja os exemplos:
  - Ontem saí para **dançar** com uns amigos, mas acabamos **dançando** porque não conseguimos encontrar um lugar que nos agradasse.
  - José **cantou** a noite inteira no Karaoke e João **cantou** a noite inteira a namorada de José.

Polimorfismo

- Numa linguagem de programação, isso significa que pode haver várias formas de fazer uma “certa coisa”
- Aí vem a primeira coisa importante: que “certa coisa” é essa? A resposta mais comum é que estamos falando de chamadas de métodos, ou seja:
  - A forma mais conhecida de Polimorfismo significa que uma chamada de método pode ser executada de várias formas (ou polimorficamente). Quem decide “a forma” é o objeto que recebe a chamada.

Polimorfismo

- Sobrecarga de métodos
  - Sobrecarga (overload) é a capacidade de poder definir dois, ou mais métodos, numa mesma classe com o mesmo nome;
  - Embora os métodos possam ter o mesmo nome, eles tem obrigatoriamente que ter uma assinatura diferente (diferentes parâmetros).



Polimorfismo

- Sobreescrita de métodos
  - Sobreescrita (override) está diretamente ligada a herança.
  - Com a sobrescrita conseguimos especializar os métodos herdados das superclasses, alterando o seu comportamento nas subclasses por um mais específico.
  - A sobrescrita de métodos consiste basicamente em criar um novo método na classe filha contendo a mesma assinatura e o mesmo tipo de retorno do método sobrescrito.

Polimorfismo

- Sobreescrita de métodos