

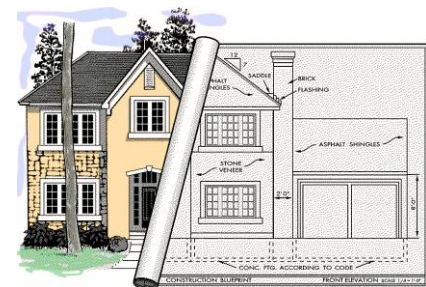
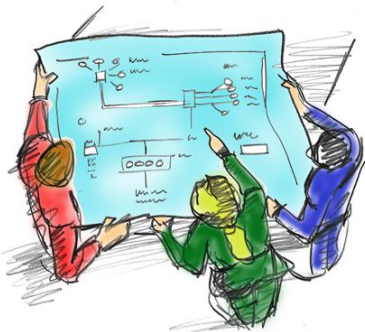
MODELAGEM DE SISTEMAS DE SOFTWARE



Professora: Aline Rosa
aline.sr@gmail.com

MODELOS DE SOFTWARE

- ✓ Na construção de sistemas de software, assim como na construção de sistemas habitacionais, também há uma gradação de complexidade.
 - ✓ A construção desses sistemas necessita de um planejamento inicial.
- ✓ Um modelo pode ser visto como uma representação idealizada de um sistema que se planeja construir.
- ✓ Maquetes de edifícios e de aviões e plantas de circuitos eletrônicos são apenas alguns exemplos de modelos.



RAZÕES PARA CONSTRUÇÃO DE MODELOS

- A princípio, podemos ver a construção de modelos como uma atividade que atrasa o desenvolvimento do software propriamente dito.
- Mas essa atividade propicia...
 - O gerenciamento da complexidade inerente ao desenvolvimento de software.
 - A comunicação entre as pessoas envolvidas.
 - A redução dos custos no desenvolvimento.
 - A predição do comportamento futuro do sistema.
- Entretanto, note o fator complexidade como condicionante dessas vantagens.

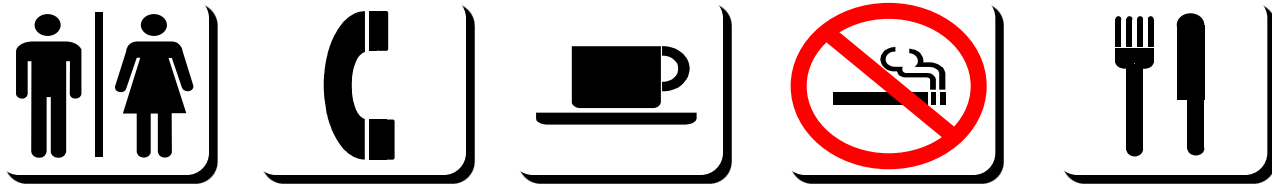
DIAGRAMAS E DOCUMENTAÇÃO

- No contexto de desenvolvimento de software, correspondem a desenhos gráficos que seguem algum padrão lógico.
- Podemos também dizer que um diagrama é uma apresentação de uma coleção de elementos gráficos que possuem um significado predefinido.
- Diagramas normalmente são construídos de acordo com regras de notação bem definidas.
 - Ou seja, cada forma gráfica utilizada em um diagrama de modelagem tem um significado específico.



DIAGRAMAS E DOCUMENTAÇÃO

- Diagramas permitem a construção de uma representação concisa de um sistema a ser construído.
 - “uma figura vale por mil palavras”



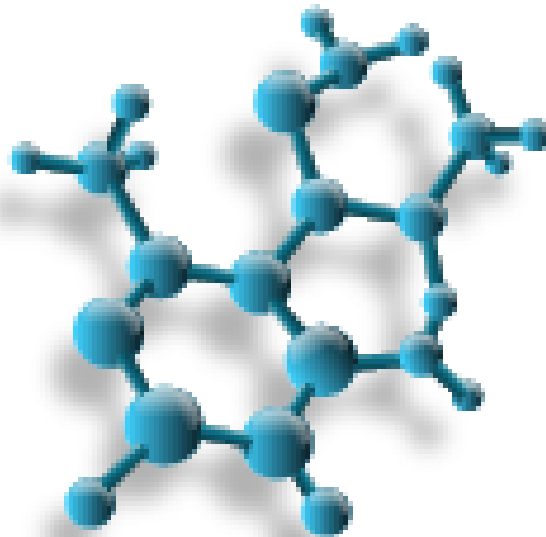
- No entanto, modelos também são compostos de informações textuais.
- Dado um modelo de uma das perspectivas de um sistema, diz-se que o seu diagrama, juntamente com a informação textual associada, formam a **documentação** deste modelo.

MODELAGEM DE SOFTWARE

A modelagem de sistemas de software consiste na utilização de notações gráficas e textuais com o objetivo de construir modelos que representam as partes essenciais de um sistema, considerando-se diversas perspectivas diferentes e complementares.



O PARADIGMA DA ORIENTAÇÃO A OBJETOS



PARADIGMA?

- *Um paradigma é uma forma de abordar um problema.*
- No contexto da modelagem de um sistema de software, um paradigma tem a ver com a forma pela qual esse sistema é entendido e construído.
- A primeira abordagem usada para modelagem de sistemas de software foi o ***paradigma estruturado***.
 - Uso da técnica de *decomposição funcional*
 - “divida sucessivamente um problema complexo em subproblemas”
- Hoje em dia, praticamente suplantou o paradigma anterior, o ***paradigma da orientação a objetos...***

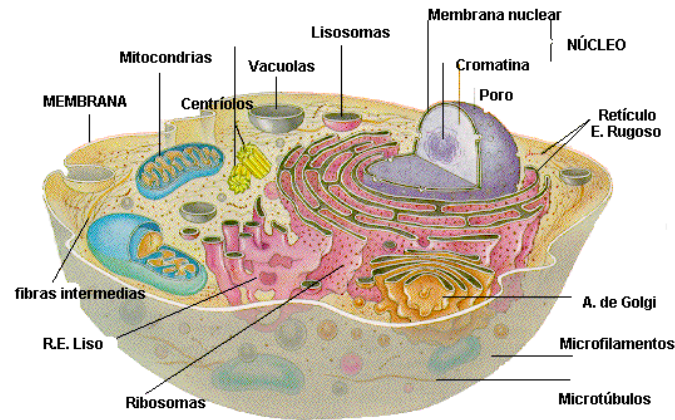
O PARADIGMA DA ORIENTAÇÃO A OBJETOS

- O paradigma da OO surgiu no fim dos anos 60.
- Alan Kay, um dos pais desse paradigma, formulou a chamada **analogia biológica**.
- *“Como seria um sistema de software que funcionasse como um ser vivo?”*



ANALOGIA BIOLÓGICA

- Cada “célula” interagiria com outras células através do envio de mensagens para realizar um objetivo comum.
- Adicionalmente, cada célula se comportaria como uma unidade autônoma.
- De uma forma mais geral, Kay pensou em como construir um sistema de software a partir de agentes autônomos que interagem entre si.



FUNDAMENTOS DA ORIENTAÇÃO A OBJETOS

Através de sua analogia biológica, Alan Kay definiu os fundamentos da orientação a objetos.

1. Qualquer coisa é um objeto.
2. Objetos realizam tarefas através da requisição de serviços a outros objetos.
3. Cada objeto pertence a uma determinada *classe*. Uma classe agrupa objetos similares.
4. A classe é um repositório para comportamento associado ao objeto.
5. Classes são organizadas em hierarquias.

O paradigma da orientação a objetos visualiza um sistema de software como uma coleção de agentes interconectados chamados ***objetos***. Cada objeto é responsável por realizar tarefas específicas. É através da interação entre objetos que uma tarefa computacional é realizada.

Um sistema de software orientado a objetos consiste de objetos em colaboração com o objetivo de realizar as funcionalidades deste sistema. **Cada objeto é responsável por tarefas específicas.** É através da cooperação entre objetos que a computação do sistema se desenvolve.

CONCEITOS E PRINCÍPIOS DA OO

❖ Conceitos

- ✓ Classe
- ✓ Objeto
- ✓ Mensagem

❖ Princípios

- ✓ Encapsulamento
- ✓ Polimorfismo
- ✓ Generalização (Herança)
- ✓ Composição

CLASSES, OBJETOS E MENSAGENS

- O mundo real é formado de coisas.
- Na terminologia de orientação a objetos, estas coisas do mundo real são denominadas *objetos*.
- Seres humanos costumam agrupar os objetos para entendê-los.
- A descrição de um grupo de objetos é denominada ***classe de objetos***, ou simplesmente de ***classe***.

O QUE É UMA CLASSE?

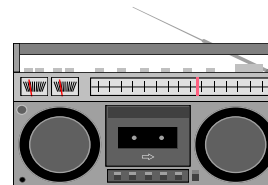
- Uma classe é um molde para objetos. Diz-se que um objeto é uma instância de uma classe.
- Uma classe é uma **abstração** das características **relevantes** de um grupo de coisas do mundo real.
 - Na maioria das vezes, um grupo de objetos do mundo real é muito complexo para que *todas* as suas características e comportamentos sejam representados em uma classe.



Representante



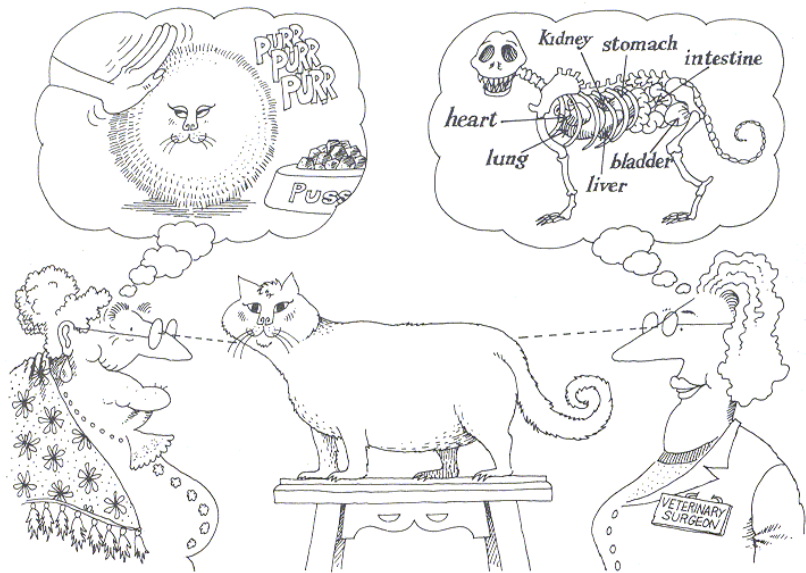
Cliente



Produto

ABSTRAÇÃO

- Uma abstração é qualquer modelo que inclui os aspectos relevantes de alguma coisa, ao mesmo tempo em que ignora os menos importantes. *Abstração depende do observador.*



Abstraction focuses upon the essential characteristics of some object, relative to the perspective of the viewer.

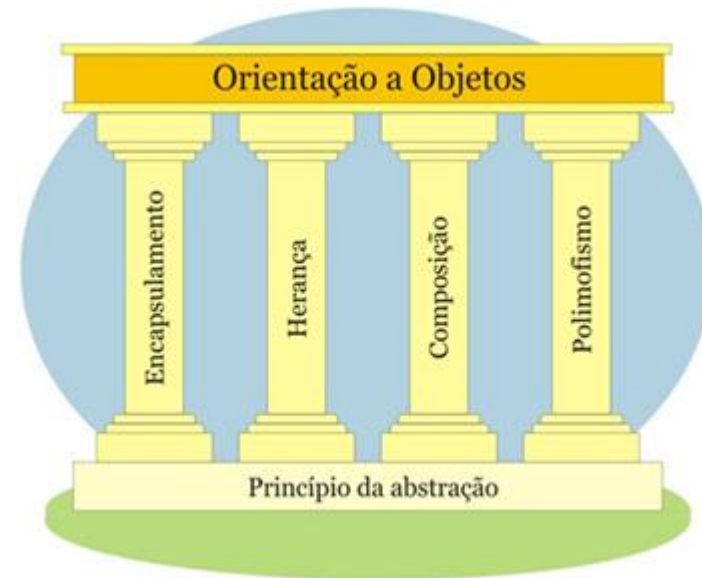


ABSTRAÇÃO NA ORIENTAÇÃO A OBJETOS

- A orientação a objetos faz uso intenso de abstrações.
 - Os princípios da OO podem ser vistos como aplicações da abstração.
- **Princípios da OO:** encapsulamento, polimorfismo, composição e herança.

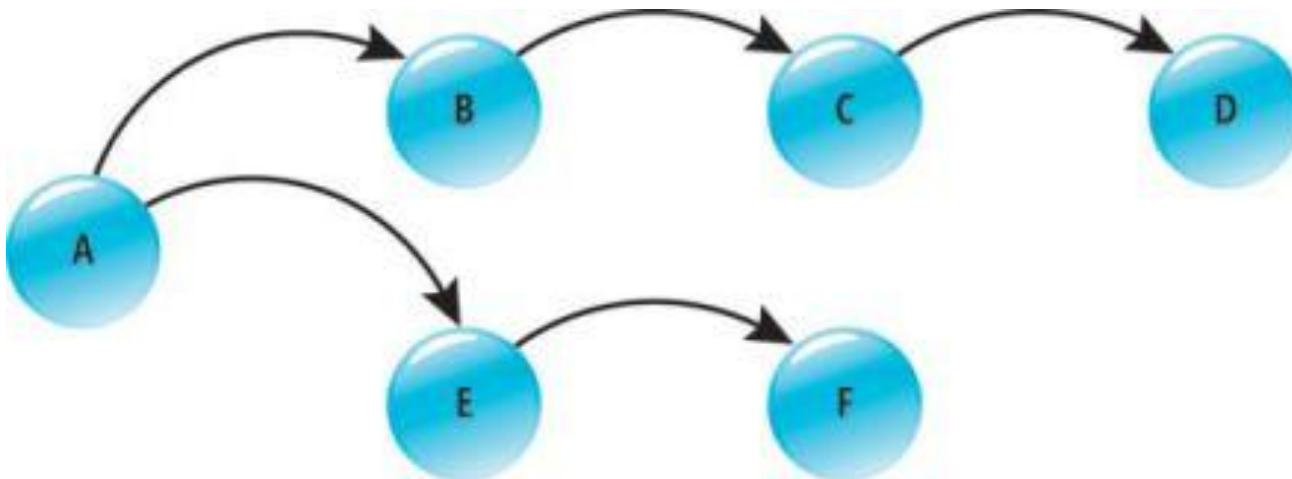
OBJETOS COMO ABSTRAÇÕES

- Uma abstração é uma representação das características e do comportamento relevantes de um conceito do mundo real para um determinado problema.
- Dependendo do contexto, um mesmo conceito do mundo real pode ser representado por diferentes abstrações.
 - Carro (para uma transportadora de cargas)
 - Carro (para uma fábrica de automóveis)
 - Carro (para um colecionador)
 - Carro (para uma empresa de kart)
 - Carro (para um mecânico)



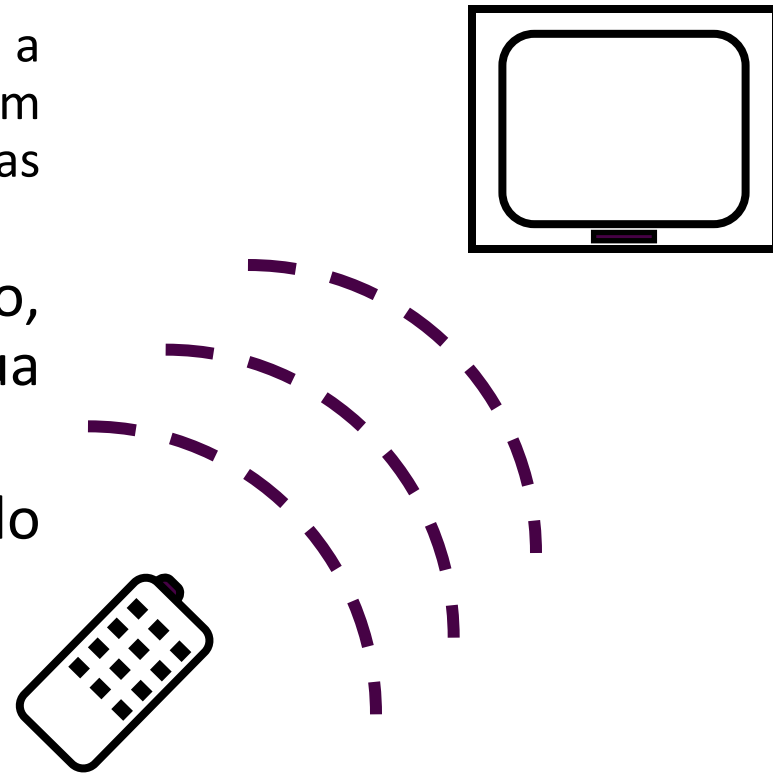
MENSAGENS

- Para que um objeto realize alguma tarefa, deve haver um estímulo enviado a este objeto.
- *Objetos de um sistema trocam mensagens*
 - isto significa que estes objetos estão enviando mensagens uns aos outros com o objetivo de realizar alguma tarefa dentro do sistema no qual eles estão inseridos.



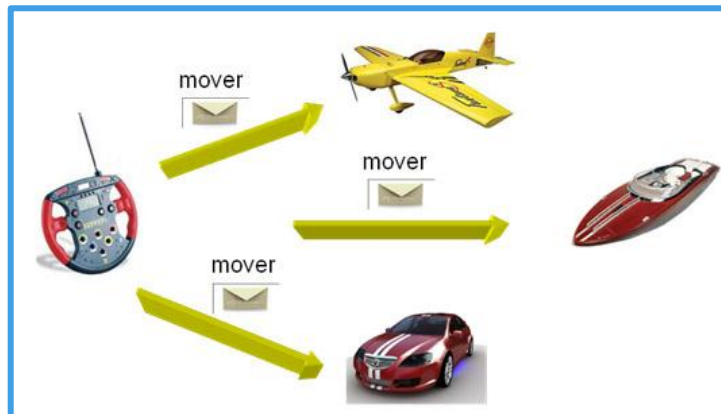
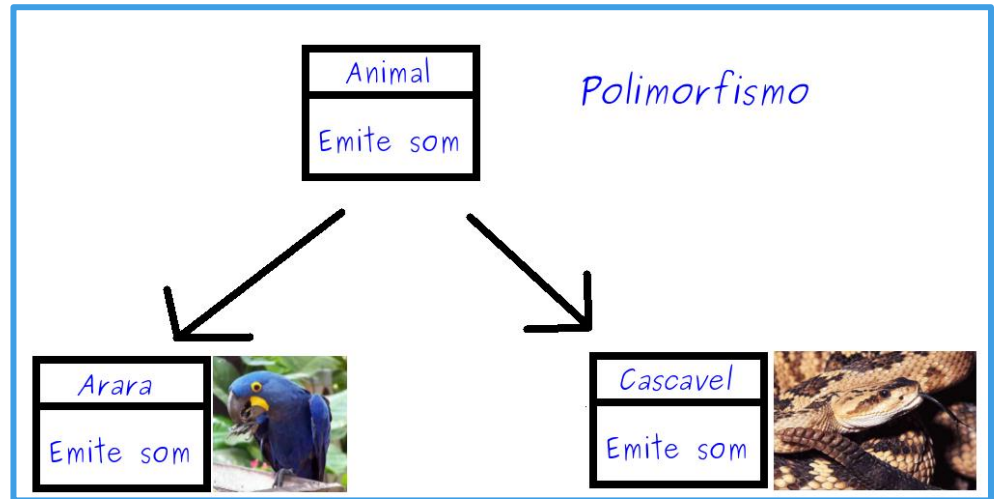
ENCAPSULAMENTO

- Objetos possuem ***comportamento***.
 - O termo comportamento diz respeito a que operações são realizadas por um objeto e também de que modo estas operações são executadas.
- De acordo com o encapsulamento, objetos devem “esconder” a sua complexidade...
- Esse princípio aumenta qualidade do SSOO, em termos de:
 - Legibilidade
 - Clareza
 - Reuso



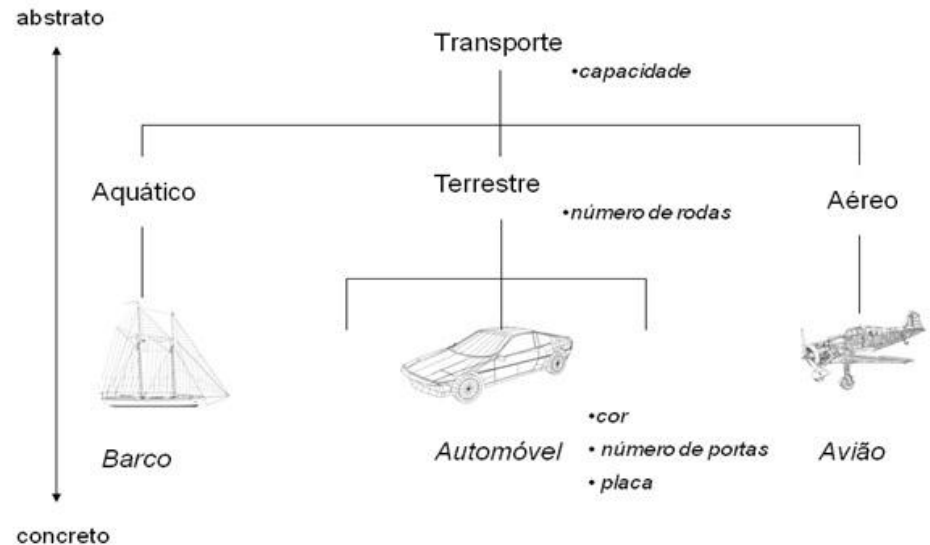
POLIMORFISMO

- É a habilidade de objetos de classes diferentes responderem a mesma mensagem de diferentes maneiras.



HERANÇA

- A herança facilita o compartilhamento de comportamento entre classes semelhantes.
- As diferenças ou variações de uma classe em particular podem ser organizadas de forma mais clara.



COMPOSIÇÃO

- É a capacidade de fazer com que uma classe seja composta de vários objetos de outras classes. A Composição oferece a possibilidade de fazer uso do comportamento das classes que ela agrega (como atributos) de maneira implícita ou explícita, dependendo da maneira que se escolhe para instanciar as classes componentes.

