



## Análise Orientada a Objetos

Breve Histórico:

- **Fim da década de 80:** amadurecimento da Orientação a Objeto
- **Década de 1990:** diversas proposições a partir de diversos autores, como Booch, Rumbaugh e Jacobson.
- **Atual:** Fortalecimento da OOA (Object Oriented Analysis) como método de análise e uma certa padronização através da utilização extensiva da UML.

*A OOA baseia-se em conceitos simples que o homem adquire desde a infância, como objetos e atributos, classes e membros, todo e partes do todo.*

**O que muda? Muda o enfoque!!**



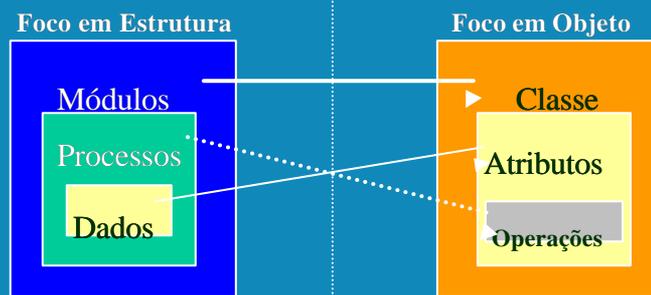
Prof. Dr. Alexandre Cardoso



## Análise Orientada a Objetos - o Enfoque

**Enfoque Tradicional:** compreensão do sistema como um *conjunto de programas* que executam processos sobre os dados.

**Enfoque OOA:** O sistema é uma *coletânea de objetos* que interagem entre si, com características próprias, representadas por atributos (dados) e operações (processos).



Prof. Dr. Alexandre Cardoso



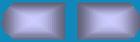
## Análise Orientada a Objetos - Fundamentos

---

- ✓ Os métodos de Orientação a Objetos apresentam uma visão mais integrada das funções e dados.
- ✓ O sistema Orientado a Objetos é estruturado através de objetos, que contemplam: funções + dados.

### Resultados:

- Produtos mais estáveis e de melhor qualidade
- Processo de desenvolvimento que permite:
  - melhor entendimento do sistema e do seu ambiente
  - melhor entendimento do domínio de aplicação
  - maior independência da implementação até estágios mais avançados



Prof. Dr. Alexandre Cardoso



## Análise Orientada a Objetos - Fundamentos

---

### Quando usar Análise OO?

- Projeto de grande porte (confinamento de informação)
- Requisitos não completamente fechados (modelos relativamente estáveis)
- Requisitos vagos, incompletos ou inconsistentes (recursos para identificar as informações)
- Novas aplicações (abordagem sistemática para melhor entendimento)
- Equipe com especialidades diversas (linguagem comum)
- Sistemas críticos (definição mais sistemática da lógica)



Prof. Dr. Alexandre Cardoso



## Sistemas Funções / Dados X Orientação a Objetos

### *A manutenção de sistemas é difícil de uma forma geral, Por que?*

- Todas as funções que manipulam os dados devem conhecer a forma de armazenamento dos dados.
- Uma modificação da estrutura de dados afeta todas as funções que manipulam esta estrutura.
- A decomposição do sistema em funções reformula os requisitos de funcionalidade para visão de implementação.
- A descrição do sistema em termos de *o que* fazer, acaba sendo transformado em *como* fazer.
- A mudança no comportamento do sistema se propaga através das funções.



Prof. Dr. Alexandre Cardoso



## Sistemas Funções / Dados X Orientação a Objetos

### *E em Sistemas Orientados a Objetos, em que este cenário se altera?*

- Um sistema OO (quando bem modelado) é estruturado através de objetos do domínio do problema.
- O sistema pode ser mantido o mais próximo possível de uma visão conceitual do mundo real.
- Há mais transparência na passagem da fase de modelagem para a fase de construção, não exigindo uma reorganização do modelo.
- Os objetos do domínio do problema são mais estáveis e as modificações acabam ficando confinadas somente a alterações nestes objetos, sem uma ampla gama de alterações em funções e dados.

*Vale ressaltar que uma boa modelagem do sistema através de objetos exige um bom conhecimento do domínio do problema.*



Prof. Dr. Alexandre Cardoso



## Características da AOO

1. Objetos são abstrações de entidades do mundo real (ou de algum sistema) que se auto-gerenciam.
2. Objetos são independentes e encapsulam suas representações de estado e de informações
3. A funcionalidade de um sistema é expressa em termos de serviços que os objetos prestam.
4. Áreas de dados compartilhadas são eliminadas.
5. Objetos se comunicam através do envio de mensagens.
6. Objetos podem ser distribuídos.
7. Objetos podem ser executados seqüencialmente ou de forma paralela.



Prof. Dr. Alexandre Cardoso



## Desenvolvimento voltado a Objetos

### O Projeto, a Análise e a Programação são atividades distintas:

- 📄 Análise OO se preocupa com a modelagem dos objetos para o domínio de aplicação.
- 📄 Projeto OO se preocupa com o desenvolvimento de um modelo de sistema que implemente os requisitos definidos pela AOO.
- 📄 Programação OO se preocupa com a implementação do Projeto OO usando uma linguagem de programação OO (como *C++* ou *Java*).



Prof. Dr. Alexandre Cardoso



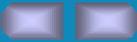
## Conceitos Básicos

### *Objetos:*

Objetos são entidades em um sistema de software que representam instâncias de entidades do mundo real e de algum sistema.

### Características:

- Objeto é um conceito, uma abstração ou uma coisa, com limites e significados bem definidos, em relação ao problema considerado.
- Um objeto é geralmente identificado por um substantivo.
- Um objeto contém estrutura e comportamento.



Prof. Dr. Alexandre Cardoso

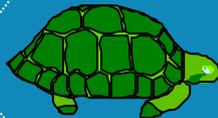


## Objetos

### Exemplos:



Um automóvel



Um animal



Uma pessoa



Um documento



Prof. Dr. Alexandre Cardoso



## Objetos

Objeto	Parte Estrutural	Parte Comportamental
um parágrafo do texto	página a que pertence número de linhas	pode ser lido
uma pessoa	nome idade peso	pode mudar de emprego pode mudar de residência
uma janela da tela	dimensões cor posição na tela	pode ser aberto pode ser fechado pode ser lido pode ser alterado
um polígono	vértices cor do contorno cor do fundo	pode ser apresentado pode ser apagado pode ser movido pode ser rodado



Prof. Dr. Alexandre Cardoso



## Objetos

### Conceitos sobre objetos:

1. **Abstração:** ressaltar os aspectos essenciais de um dado objetos, descartando aspectos desnecessários
2. **Encapsulamento de Informações:** um objeto contém informações que não são acessíveis a outros objetos
3. **Atributos:** uma propriedade do objeto identificada por um nome. Atributos são valores dos dados que caracterizam os objetos. Um atributo tem um valor único para um objeto. \*\*

\*\* . conceito que será melhorado ao longo do curso.

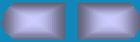


Prof. Dr. Alexandre Cardoso



## Objetos

4. **Operação**: um serviço que é requisitado a um objeto, e que é parte de seu comportamento em resposta à solicitações externas. Uma operação pode ter parâmetros. De forma simples, Operação é uma ação ou transformação realizada por um objeto ou sofrida por um objeto.
5. **Mensagens**: objetos se comunicam através de mensagens, isto é, sinais enviados de um objeto a outro, requisitante de um serviço, através da execução de uma operação pelo objeto requisitado.
6. **Polimorfismo**: “várias formas”: refere-se aos vários comportamentos que uma mesma operação pode assumir, assim como a capacidade de uma variável referir-se a diferentes objetos, que preenchem certas responsabilidades em função da mensagem que lhes é passada.

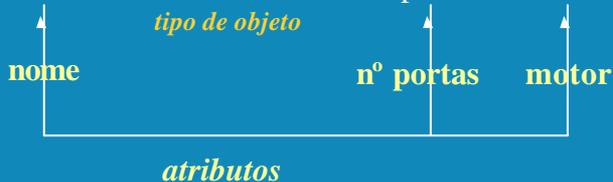


Prof. Dr. Alexandre Cardoso



## Atributos e Operações

Corsa Sedã 1.0 é um automóvel de 5 portas e motor 1.0



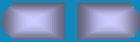
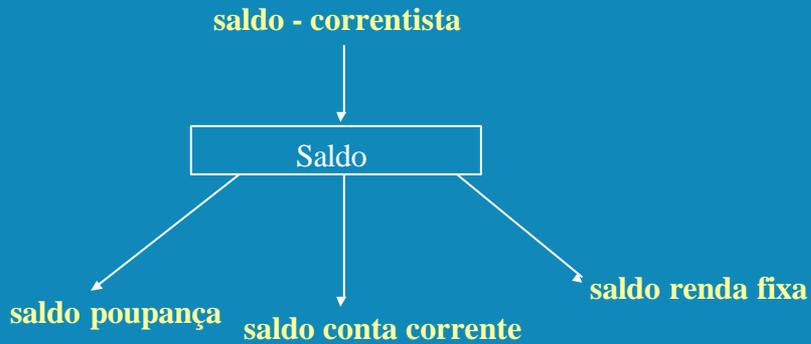
Corsa Sedã 1.0 é um automóvel de 5 portas e motor 1.0, que foi lançado em 1996



Prof. Dr. Alexandre Cardoso



## Polimorfismo



## Classes

**Classes** são *templates* (formas) de objetos. Elas são usadas para criar objetos. (Classes podem herdar atributos e serviços de outras classes).

Classe é uma descrição de:

- um grupo de objetos, que tem:
  - propriedades (atributos) semelhantes,
  - comportamento (operações) comum e
  - os mesmos tipos de relacionamentos com outros objetos.

Obs: A escolha de uma classe é arbitrária e depende da aplicação em desenvolvimento.





## Classes - Exemplos



Canário

### Classe Aves



Koala

Coioote

### Classe Mamíferos



Prof. Dr. Alexandre Cardoso



## Classes

### Classes      Atributos      Operações

Classes	Atributos	Operações
Parágrafo	página a que pertence número de linhas	ler
Pessoa	nome idade peso	mudar de emprego mudar de residência
Janela	dimensão cor posição na tela	abrir fechar ler alterar
Polígono	vértices cor do contorno cor do fundo	apresentar apagar mover rodar



Prof. Dr. Alexandre Cardoso



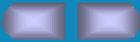
## Classes - conceitos

---

1. **Método**: implementação de uma dada operação para uma classe.

### 2. Superclasse / Subclasse

- Uma classe pode ser refinada em subclasses (especialização).
- Ao colocar em evidência os atributos e as operações de um conjunto de classes, identifica-se uma superclasse (generalização).
- Uma subclasse herda as propriedades da superclasse e tem as suas propriedades específicas.



Prof. Dr. Alexandre Cardoso



## Classes - conceitos

---

3. **Herança**: capacidade de um novo objeto tomar atributos e operações de um objeto existente.

É o compartilhamento de atributos e operações entre as classes que possuem relacionamento de hierarquia.

- Exemplos:

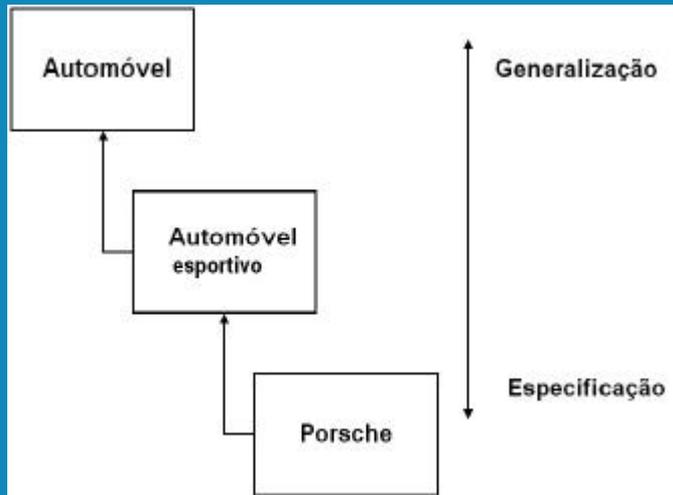
- uma pessoa pode ser um professor ou estudante
- um estudante pode ser de graduação ou pós-graduação



Prof. Dr. Alexandre Cardoso



# Herança



# Elementos Principais da AOO



## Objetos, Classes e Pacotes





## Objetos: Identificação

*Facilidade de obter objetos físicos: olhe ao seu redor!*

No domínio de solução de um problema de Eng. de Software: esta tarefa é mais difícil...

**Objetos são entidades em um sistema de software que representam instâncias de entidades do mundo real e de algum sistema.**

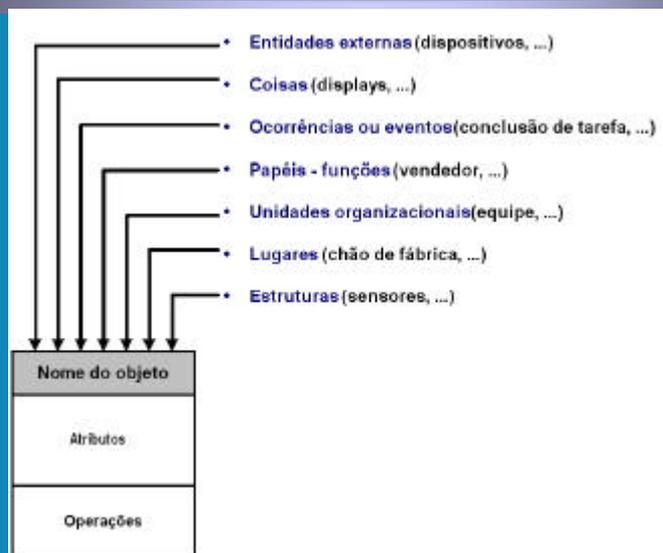
- Um **objeto** é uma entidade que possui um *estado* e define um conjunto de *operações* que modificam este estado.
  - O estado do objeto é representado por um conjunto de atributos
  - As operações associadas com o objeto provêm serviços para outros objetos (clientes), os quais requisitam serviço quando alguma computação é necessária. Tal requisição é feita através do envio de mensagens entre os objetos.



Prof. Dr. Alexandre Cardoso



## Objetos: Identificação



Prof. Dr. Alexandre Cardoso



## Objetos: Identificação

**Entidades Externas:** produzem ou consomem informações que são usadas por um sistema baseado em computador

**Coisas:** elementos que fazem parte do domínio de informação do problema, tais como relatórios, cartas, cartazes

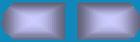
**Ocorrências ou Eventos:** ocorrem durante o contexto de operação do sistema

**Papéis:** desempenhados por elementos que interagem com o sistema, tais como: vendedor, gerente, engenheiro

**Unidades Organizacionais:** pertencem à organização

**Lugares:** auxiliam na definição do contexto - ex: piso da fábrica

**Estruturas:** definem classes relacionadas com o sistema, ex: sensores, veículos, computadores



Prof. Dr. Alexandre Cardoso



## Objetos: Identificação

**Regras para identificar quem não são objetos:**

- geralmente, objetos não têm nome procedimental imperativo - ex.: rotação de imagem.

**Lembre-se:** Objeto é um conceito, uma abstração ou uma coisa, com limites e significados bem definidos, em relação ao problema considerado.

- ☐ Um objeto é geralmente identificado por um substantivo.
- ☐ Um objeto contém estrutura e comportamento.
- ☐ Cada objeto tem sua identidade.
- ☐ Dois objetos são distintos, mesmo que eles apresentem as mesmas características.

Exemplo: 1 dezena de automóveis Astra

- cada automóvel é um objeto!
- todos tem a mesma característica!



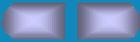
Prof. Dr. Alexandre Cardoso



## Objetos: Identificação

### Dicas:

- ↑ sublinhar cada nome ou cláusula nominal;
- ↑ separar objetos do domínio da solução de objetos do domínio do problema
- ↑ após isolar os nomes, tentar identificar as maneiras relacionadas com os objetos possíveis
- ↑ Uma definição de classe serve como forma (*template*) para os objetos: Ela inclui declarações para todos os atributos e serviços que devem estar associados a um objeto daquela classe.
- ↑ Classes são *fábricas* de objetos...



Prof. Dr. Alexandre Cardoso



### Ex.:



**Encapsulação:** O objeto encapsula os dados (valores dos atributos), as operações, outros objetos, constantes e outras informações necessárias.



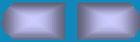
Prof. Dr. Alexandre Cardoso



## Classes e Objetos



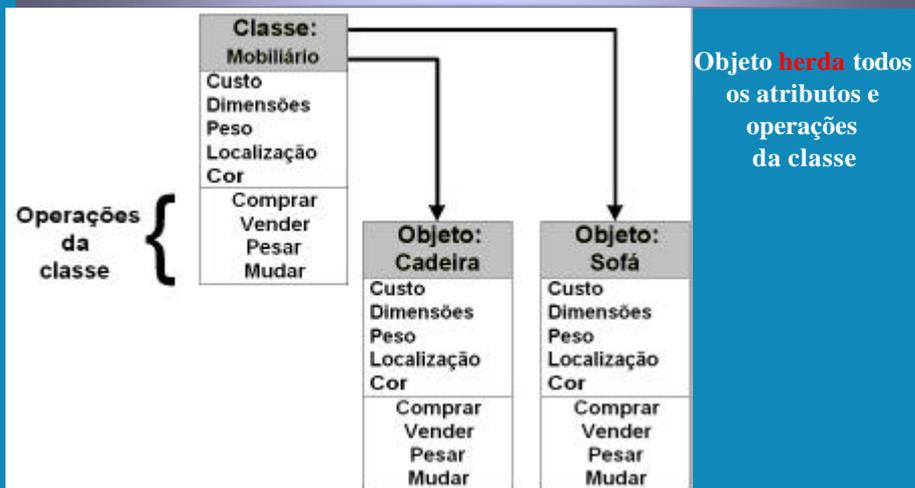
**Cadeira:** um **objeto** do mundo real que é **membro** (instância) de uma **classe** maior de objetos que podemos denominar **mobiliário**.



Prof. Dr. Alexandre Cardoso



## Classes e Objetos



As **operações** (serviços ou métodos) permitem que modifiquemos os atributos dos objetos da classe.



Prof. Dr. Alexandre Cardoso

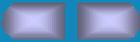


## Comunicação entre Objetos

### Mensagens:

- ✓ elemento usado para prover a comunicação entre objetos
- ✓ Definem:
  - O nome do serviço requisitado
  - A informação necessária para a execução do serviço
  - O nome do requisitante.
- ✓ Na prática, mensagens são implementadas como ativações de uma função definida no objeto chamado, onde:
  - Nome é o nome da função.
  - Informação é a lista de parâmetros.
  - Requisitante é o objeto que realizou a chamada.

**"ORIENTADO A OBJETO" = *Objetos* + *Classificação* + *Herança*  
+ *Comunicação***



Prof. Dr. Alexandre Cardoso



## Pacotes

*Pacote é o mecanismo de propósito geral usado para organizar elementos de modelo em grupos*

Pacote = item de agrupamento!!!

- Um pacote pode estar aninhado dentro de outro pacote!
- São porções resultantes das subdivisões em limites que permitam coesão lógica e funcional
- Em sistemas de grande porte, pacotes devem ser identificados o mais cedo possível - durante a fase de modelagem

*Notação: retângulo com aba superior:*



OU



Prof. Dr. Alexandre Cardoso

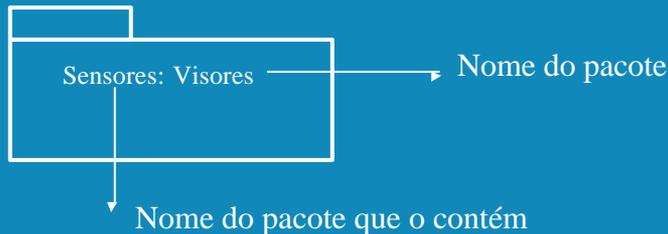


## Pacotes

- referem-se a agrupamentos que organizam um modelo
- um pacote permite controlar o acesso a seus conteúdos
- Nome\* do pacote:

- simples
- nome do caminho

*\*Obs: O nome do pacote deve ser uma expressão ou substantivo curto*



## Pacotes

### Dicas de Concepção:

- Um pacote pode conter outros elementos, incluindo classes, interfaces, componentes, nós, colaborações, casos de uso, diagramas ou outros pacotes
- Um elemento mantém uma relação únivoca com um pacote
- Se o pacote for destruído, todos os seus elementos também são destruídos
- Não há dois elementos de mesmo tipo com o mesmo nome no mesmo pacote
- Na prática, é saudável evitar o aninhamento de pacotes (ideal: dois ou três níveis de aninhamento)
- Pacotes são mecanismos importantes lidar com escala





# Pacotes

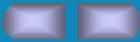
## Visibilidade:

- tipicamente, um elemento pertencente a um pacote tem visibilidade pública
- elementos protegidos somente podem ser vistos fora do pacote em que são declarados

+ : *elementos públicos*

# : *protegido*: podem ser vistos apenas por pacotes que herdam de um outro pacote

- : *privado*: não podem ser vistos de forma alguma fora do pacote



Prof. Dr. Alexandre Cardoso

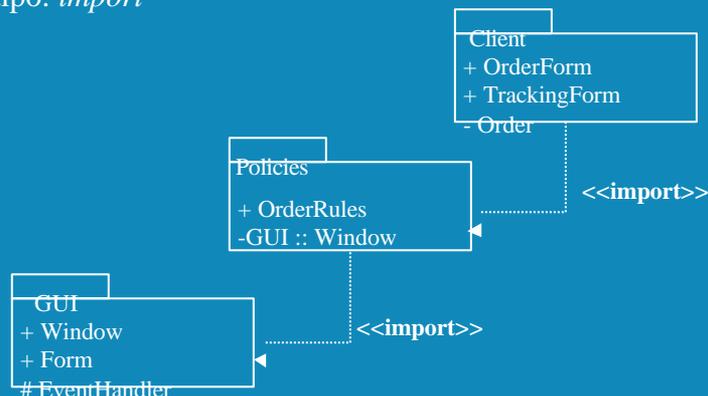


# Pacotes

## Importação e Exportação:

Importação: assegura uma permissão unilateral para que os elementos de um pacote tenham acesso a elementos pertencentes a outros pacotes

- estereótipo: *import*



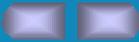
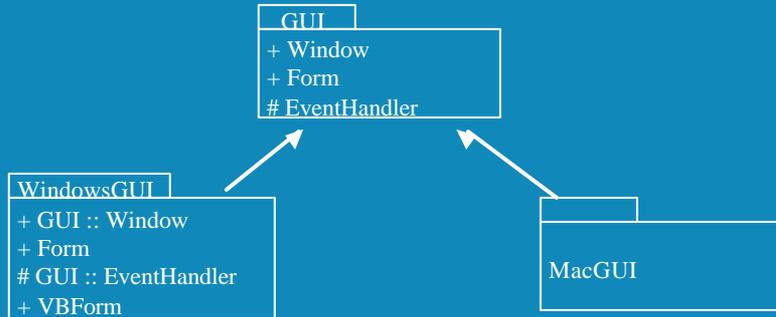
Prof. Dr. Alexandre Cardoso



# Pacotes

## Generalização:

- outra forma de relacionamento entre pacotes
- muito semelhante à generalização entre classes
- permite herança de elementos



# Pacotes

## Dependência:

- existe se houver dependência entre duas ou mais classes dentro desses pacotes
- dependências não são transitivas

