Elementos Principais da AOO

Objetos, Classes e Pacotes

Objetos: Identificação

Facilidade de obter objetos físicos: olhe ao seu redor!

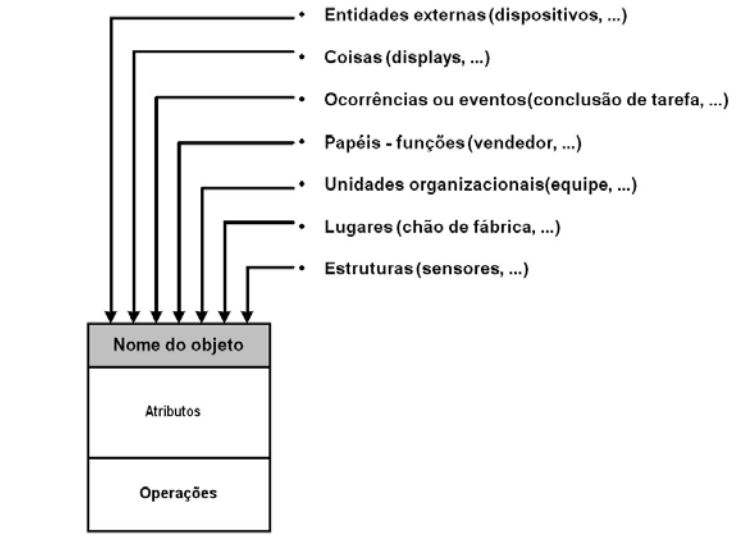
No domínio de solução de um problema de Eng. de Software: esta tarefa é mais difícil...

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.

Objetos: Identificação



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

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

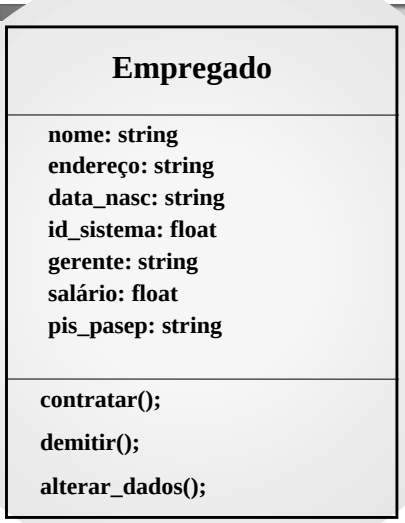
- **cada automóvel é um objeto!**
- **todos tem a mesma característica!**

Objetos: Identificação

Dicas:

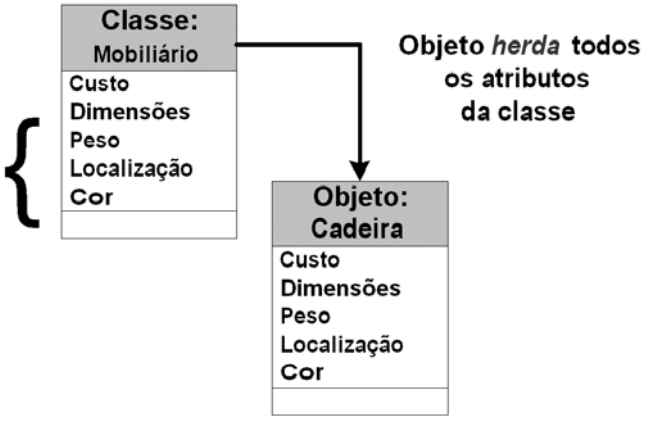
- sublinhar cada nome ou cláusula nominal;
- separar objetos do domínio da solução de objetos do domínio do problema
- após isolar os nomes, tentar identificar as maneiras relacionadas com os objetos possíveis
- Uma definição de classe serve como forma (*template*) para os objetos: Ela inclui declarações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

Ex.:



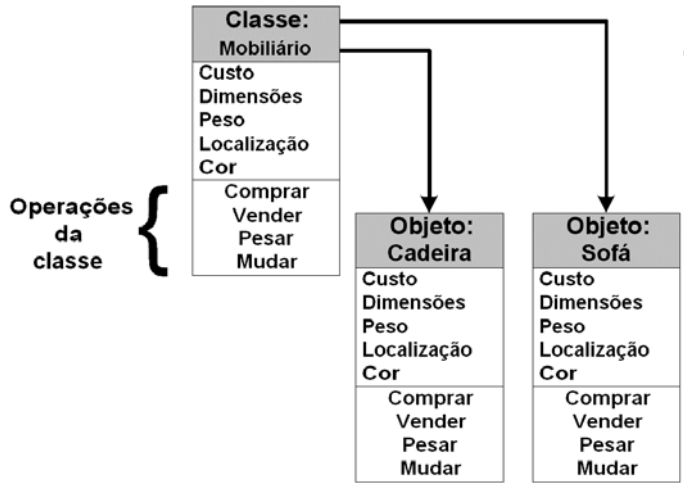
Encapsulação: O objeto encapsula os dados (valores dos atributos), as operações, outros objetos, constantes e outras informações necessárias.

Classes e Objetos



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

Classes e Objetos



Objeto herda todos os atributos e operações da classe

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

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

Pacotes

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

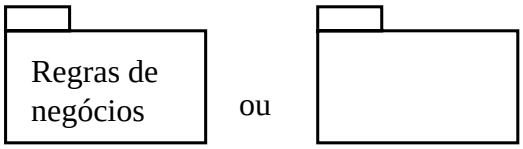
Pacote = item de agrupamento!!!

• 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

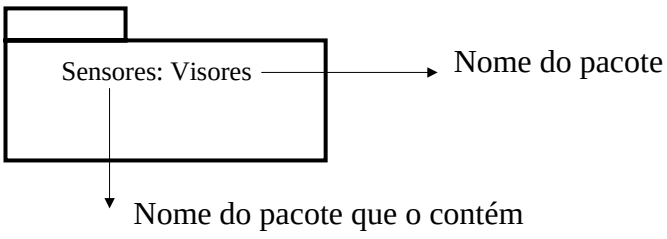
Notação: retângulo com aba superior:



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

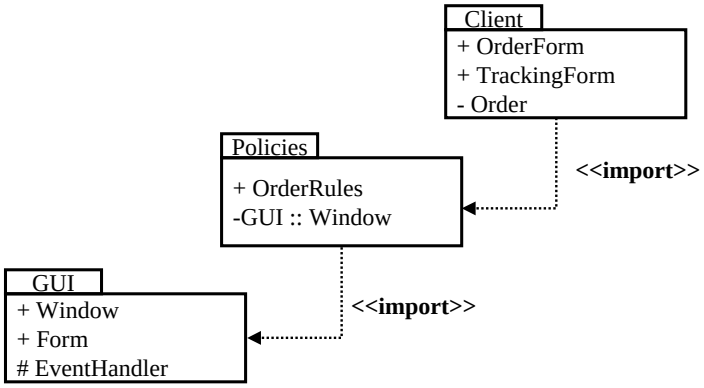
-: *privado:* não podem ser vistos de forma alguma fora do pacote

Pacotes

Importação e Exportação:

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

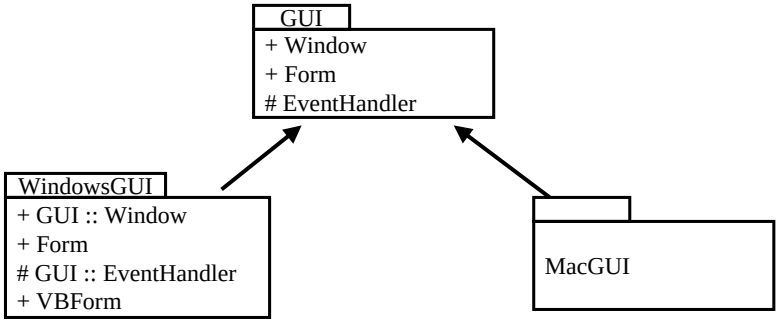
•estereótipo: *import*



Pacotes

Generalização:

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

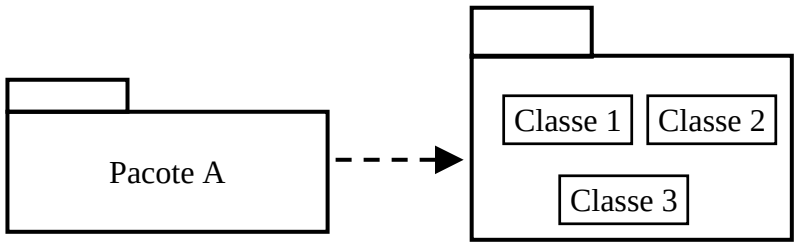


Pacotes

Dependência:

• existe se houver dependência entre duas ou mais classes dentro desses pacotes

• dependências não são transitivas



Fornecedor ***Cliente***