

*Engenharia de Software*

**Tema da Aula**  
**Análise de Requisitos - II**  
**Métodos de Análise de Requisitos**

*Prof. Cristiano R R Portella*  
portella@widesoft.com.br



**Análise de Requisitos**  
**Métodos Consagrados**

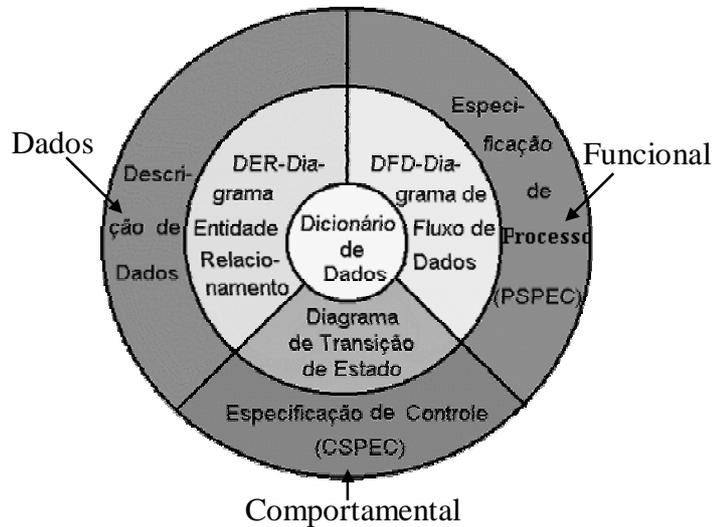
- ✓ **Análise Estruturada**
  - Baseada em Processos e Fluxos de Dados
- ✓ **Análise Orientada a Objetos**
  - Baseada em Classes, Objetos Casos de uso etc
- ✓ Análise Orientada a Estruturas de Dados
- ✓ Método de “Jackson”
- ✓ Métodos Formais

## Análise Estruturada

## Análise de Requisitos Análise Estruturada

- ✓ Uma das maiores contribuições à Engenharia de Software;
- ✓ Foi Desenvolvida no transcorrer da década de 70 e início da década de 80;
- ✓ Ainda é o método mais difundido e popular;
- ✓ Notação Gráfica que representam os dados e como estes fluem através dos processos (funções) de um sistema;

## Os Elementos do Modelo de Análise



## Análise (Estruturada) de Requisitos

### Diagrama de Fluxo de Dados:

Fluxo da informação e suas transformações;

### Dicionário de Dados:

Repositório que contém a descrição de todos os objetos de dados alimentados ou produzidos pelo software.

### Diagrama Entidade-Relacionamento:

Mostra as relações (cardinalidade e modalidade) existentes entre os itens de dados.

### Diagrama de Transição de Estado:

Destaca o evento e seu impacto sobre o sistema (fluxo de controle).

DFD - Diagrama de Fluxo de Dados:

- ✓ Existem duas notações populares de DFD
  - Notação de Gane/Sarson
  - Notação Tom DeMarco (Diagrama de Bolhas)
- ✓ O DFD fornece um modelo visual, contendo:
  - abrangência e limites do sistema
  - Entidades que interagem com o sistema
  - Dados e seu fluxo através do sistema
  - Processos (funções) executados pelo sistema

Entidade

**Símbolo**

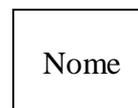
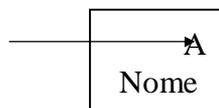
**Símbolo**

Externa

**GA**

**DeMarco**

Referência



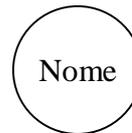
Representam as entidades que fornecem ou recebem dados do sistema

Processo  
 (Função)

**Símbolo**  
**GANE**

**Símbolo**  
**DeMarco**

Referência



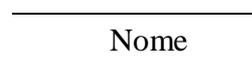
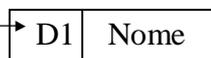
- ✓ Representam os processos que o sistema executa, ou seja, a transformação de dados de entrada em dados de saída

Depósito  
de Dados

**Símbolo**  
**GANE**

**Símbolo**  
**DeMarco**

Referência



- ✓ Dados que ficaram armazenados por algum período de tempo (representam a necessidade de sincronismo de tempo).

Fluxo de  
Dados

**Símbolo  
GANE**

**Símbolo  
DeMarco**

Nome do Fluxo  


Nome do Fluxo  


- ✓ Referenciam a Origem e o Destino (funções ou depósito de dados/funções).

- ✓ Modelo do Sistema em Níveis. Cada Nível agrega detalhes ao nível anterior;
- ✓ Quanto maior a complexidade do sistema, mais níveis deverão ser criados no modelo.
- ✓ DFD Nível 0 - Diagrama de Contexto
  - Limites e Abrangência do Sistema
  - Entidades que Interagem com o Sistema
  - Informações que entram e que são produzidas pelo sistema

## Diagrama de Fluxo de Dados

- ✓ DFD Nível 1
  - Expansão (Explosão) do Nível 0
  - Mostra os Processos que o sistema executa, os Depósitos de Dados e os Fluxos de Dados Internos ao Sistema
- ✓ DFD Nível 2
  - Expansão (Explosão) dos processos no Nível 1 consideradas complexas
  - Nem todos os processo de nível 1 serão expandidos no nível 2
- ✓ Níveis de Alta Ordem (3, 4, 5, etc...)

## Projeto Orientado a Fluxo de Dados

### Diagrama de Fluxo de Dados - Nível 0



Diagrama de Fluxo de Dados - Nível 1

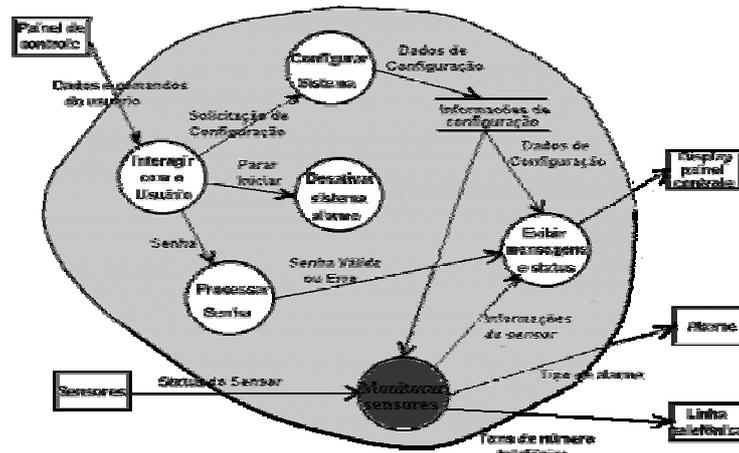


Diagrama de Fluxo de Dados - Nível 2

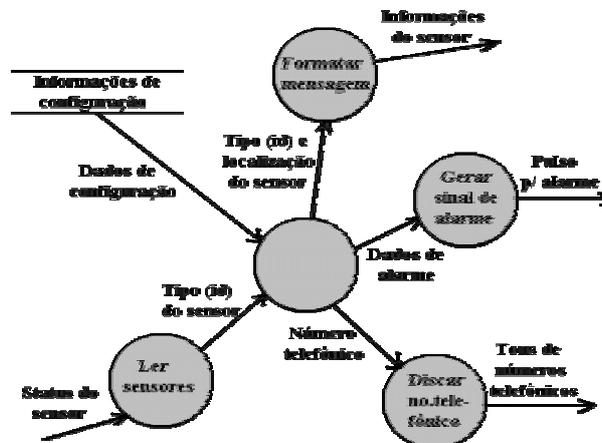
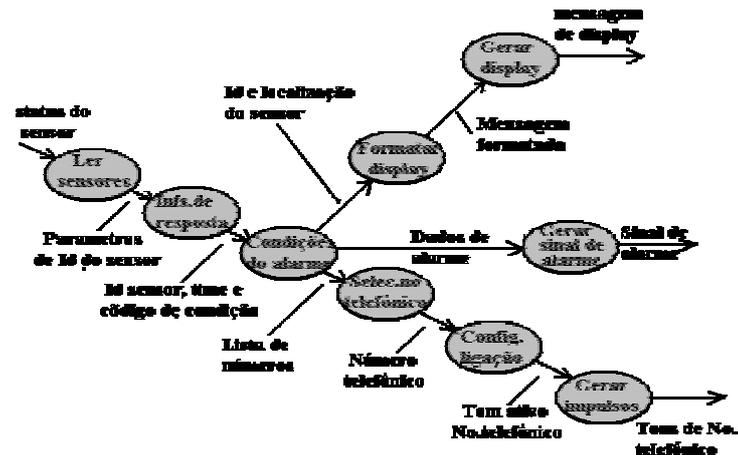


Diagrama de Fluxo de Dados - Nível 3



Análise (Estruturada) de Requisitos  
Diagrama Entidade-Relacionamento

Cardinalidade:

Número de ocorrências de um objeto (de dados) em relação às ocorrências de outro objeto.

Esposa  $\frac{1}{(tem)}$   $\frac{1}{(tem)}$  Esposa

Pai  $\frac{1}{(tem)}$   $\frac{0 : n}{(tem)}$  Filhos

Tio  $\frac{0 : n}{(tem)}$   $\frac{0 : m}{(tem)}$  Sobrinho(a)

Modalidade:

Obrigatoriedade (ou não) de ocorrer o relacionamento. Pode ser:

- 0 (zero) se não há necessidade explícita do relacionamento, ou ele é opcional.
- 1 se uma ocorrência do relacionamento é mandatória.



(p.s. ele pode ser autor de paper, monografia, artigos etc e não necessariamente livro)

### Dicionário de Dado: Modelo descritivo Padronizados dos itens componentes do DFD

- ✓ Itens que devem ser dicionarizados
  - Entidades Externas
  - Processos
  - Depósitos de Dados e Fluxos de Dados, com Descrição, Referências e Conteúdo
    - Estruturas de Dados
    - Elementos de Dados

### Dicionário de Dados: Itens

- Nome (do item)
- Alias
- Onde é usado
- Como é usado
- Descrição (notação)
- Outras informações

Dicionário de Dados: Notação

Notação	Significa	Operação
=	É composto de	Agregação
+	E	Sequência
[   ]	ou   ou	Seleção
( )	n repetições de	Repetição
* ..... *	Comentários	

Dicionário de Dados: Exemplo

- ✓ Nome: Número do telefone
- ✓ Aliás: Não tem
- ✓ Onde/como é usado: Entrada de dados dial-up
- ✓ Descrição:

no.telefone = prefixo + no. acesso

prefixo = \* número de 5 dígitos não iniciando por zero\*

no. acesso = \* string com qualquer número de dígitos \*

Nome do Fluxo: Dados do Usuário

Referências: B-3; 3-D1

Conteúdo:

Código de Identificação

Nome

Login

Tipo de Usuário

- ✓ Cada elemento e/ou estrutura de dados deve ser dicionarizado

Nome do Elemento: Código de Identificação

Descrição: Código que identifica os usuários

Aliases: RA (alunos); RU (professores e funcionários)

Tamanho: 8 posições Numéricas

Valores Válidos: Qualquer seqüência de 8 dígitos, sendo os dois primeiros o ano de ingresso na Universidade

Nome do Elemento: Tipo de Usuário

Descrição: Descreve qual a categoria a qual pertence o usuário do LABI

Aliases: Não há

Tamanho: 1 caracter

Valores Válidos:

‘A’ para categoria Aluno

‘P’ para categoria Professor

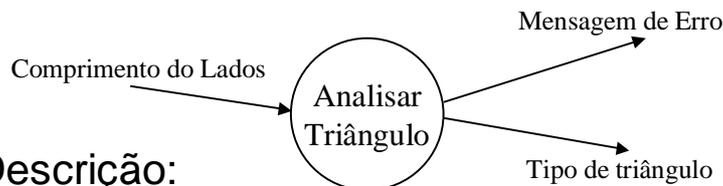
‘F’ para categoria Funcionário

✓ Objetivo: Descrever a finalidade do processo, suas entradas, saídas e como as informações são processadas

✓ Formas de Descrição

- Especificação Narrativa
- PDL/LDA (Program Design Language/ Linguagem de Descrição de Algoritmos ou Português Estruturado)
- Tabela de Decisão
- Árvores de Decisão
- Mistura das opções anteriores

✓ Nome: Analisar Triângulo



✓ Descrição:

O processo aceita três valores que representam as dimensões dos lados do triângulo. Testa os valores dimensionais para determinar se todos os lados são positivos; se um valor negativo for identificado é produzida uma mensagem de Erro. Se as dimensões definem um triângulo válido o mesmo será classificado como saída da função (isósceles, escaleno ou equilátero)

✓ Especificação em PDL/LDA (Port. Estrut.)

Ler as 3 dimensões do triângulo  
se qualquer dimensão for negativa  
então produzir Mensagem de Erro  
se a menor dimensão for menor que a soma das demais  
então  
| se 3 lados iguais      então Triângulo Equilátero  
| se dois lados iguais   então Triângulo Isósceles  
| se 3 lados diferentes   então Triângulo Escaleno  
| Produzir Resultado Tipo do triângulo  
senão produzir mensagem de erro

## Análise Orientada a Objetos

## Análise Orientada a Objetos

- ✓ Amadurecimento durante a década de 90
  
- ✓ Baseia-se nos seguintes conceitos:
  - Objetos e Classes (Categorias) de Objetos
  - Atributos e operações
  - Associação entre objetos
  - Todo e Parte
  - Herança
  - Troca de Mensagens

### **Objeto:**

Uma entidade **única**, identificável em um determinado nível de abstração

Uma unidade (real ou abstrata) **individual e identificável** que modela algum conceito da realidade humana, ocupando espaço físico (mundo real) ou lógico (na memória)

Exemplos de Objeto:

Um Carro, uma Nota Fiscal, uma Conta Corrente, uma Pessoa etc.

### **Exemplos de Objetos:**

- ✓ **Entidades Externas** ao Sistema (chamados em OO de Atores)
- ✓ **Coisas** que são manipuladas/geradas pela aplicação (Relatórios, cartas, cartões, recibos, etc..)
- ✓ **Papéis** Desempenhados no sistema (gerente, Vendedor, Alunos, etc..)
- ✓ **Ocorrência de Eventos** (Clique de um mouse em um botão, Conclusão de uma série de movimentos de um robô)

**Classe:**

Definição de um conjunto de objetos que **compartilham** os mesmo **atributos** e **comportamento**.

Representação de um conjunto de coisas reais ou abstratas que são reconhecidas como sendo do mesmo tipo por **compartilhar** as mesmas **características, atributos, relações e semântica**.

✓ **Classificação:**

Processo de agrupar objetos em classes.

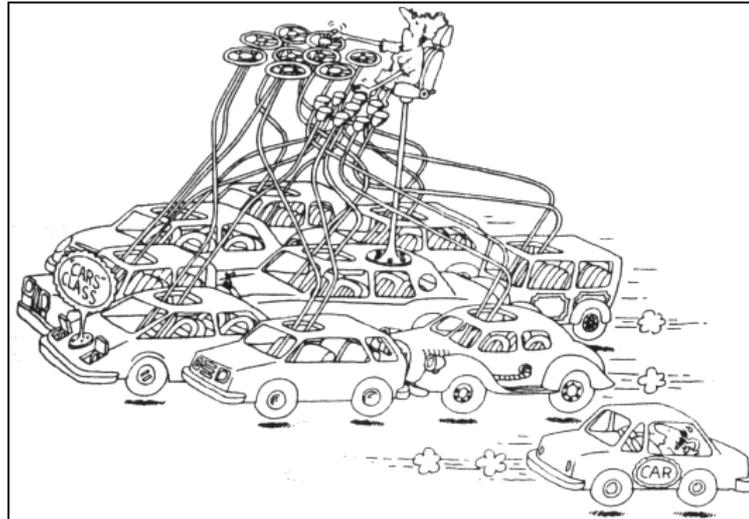
✓ **Exemplo de Classes:**

“Portella” é objeto (instância) da classe **Professor**

“Maria” é objeto da classe **Aluno**

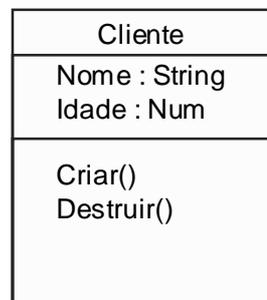
“Gol” é objeto da classe **Automóvel**

## Classes x Objetos



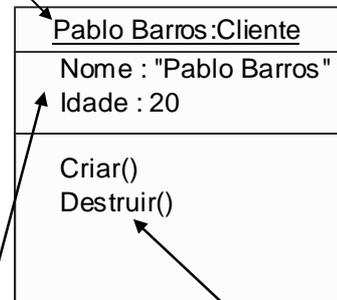
## Notação Classes e Objetos

### Classes



### Objetos

Identidade ➤ nome:classe



Atributos

Operações

**Atributo**

**Propriedade** associada ao objeto, que pode assumir valores dentro de um domínio

Menor unidade que em si possui significado próprio e interrelacionado com o conceito da classe a qual pertence.

**Exemplos de Atributos:**

✓ Atributos da Classe **Homem** (Ser Humano)  
Nome, Data\_de\_Nascimento, etc.

✓ Atributos da Classe **Arquivo**  
Nome, Tipo, Tamanho, Visibilidade, etc.

Os atributos são reconhecidos, definidos e usados conforme o **domínio do problema**

**Operação (Serviços):**

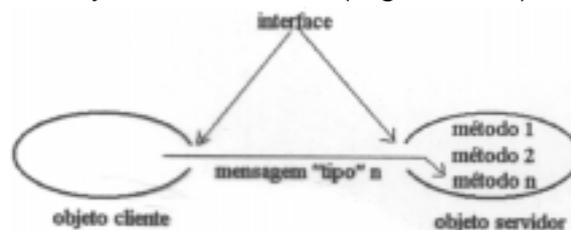
É um **serviço** provido por uma classe ou comportamento resultante de um procedimento algorítmico.

Componentes que precisam ser executados para prover um determinado serviço.

✓ Operações sempre são invocadas (chamadas à execução) por um objeto, através de uma **mensagem**.

- ✓ **Operação** é algo que é invocado por um objeto, com objetivo de prestar algum **serviço** definido pela classe desse objeto.
- ✓ **Operação** causa mudança em um objeto de alguma forma, mais especificamente em seus atributos .
- ✓ **Método** é a implementação de uma operação (serviço) para uma classe específica.

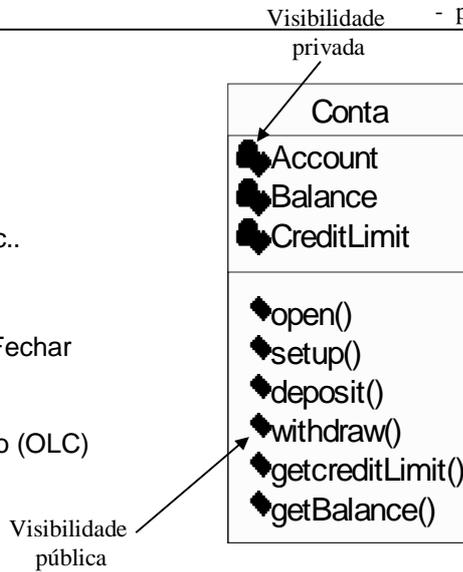
- ✓ **Troca de Mensagens**  
Mecanismo que permite a um objeto solicitar que outro objeto lhe preste algum serviço.
- ✓ **Mensagem**  
Solicitação de serviços entre objetos
- ✓ **Mensagem Típica:**  
saída=objeto-alvo.método (argumentos)



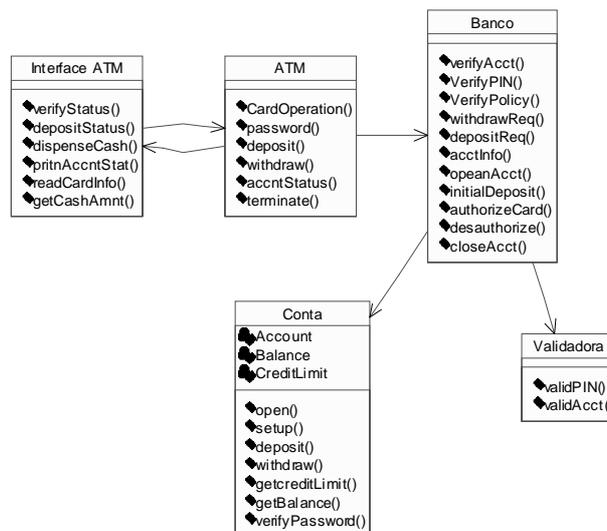
## Exemplo de Classe

+ publica  
 # protegida  
 - privada

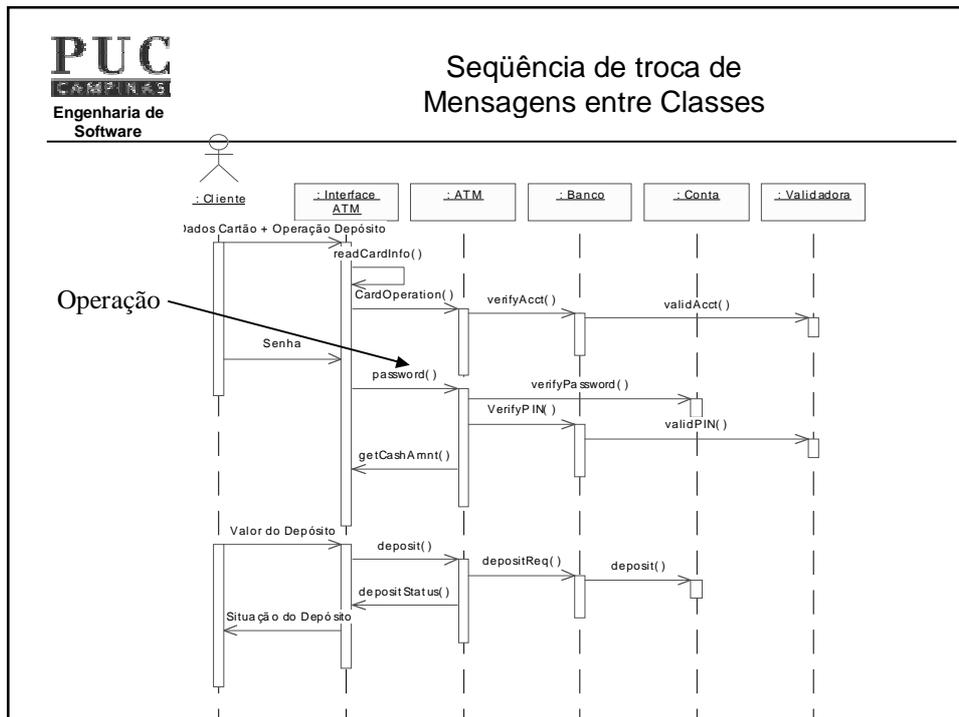
- ✓ Atributos:  
 #Conta, Saldo,  
 Limite\_Credito,  
 Data\_abertura, etc..
- ✓ Operações:  
 Abrir, Cadastrar, Fechar  
 Depositar, Sacar  
 ObterSaldo (OS),  
 ObterLimiteCrédito (OLC)



## Classes de um Sistema – Caixa Eletrônico



## Seqüência de troca de Mensagens entre Classes



## Conceitos Básicos de OO

### Herança:

Mecanismo que permite a definição de uma classe (sub-classe ou classe filha) que inclua atributos e comportamento de uma classe mais genérica (super-classe, ou classe mãe).

### Princípio da Herança:

Permitir que características comuns (operações e atributos) de uma classe mais abstrata (conceitual) sejam redefinidas nas classes mais concretas dela derivadas, ou seja, permitir que tais características possam ser adaptadas às particularidades.

**Herança:**



✓ **Mecanismos de abstração de Heranças**

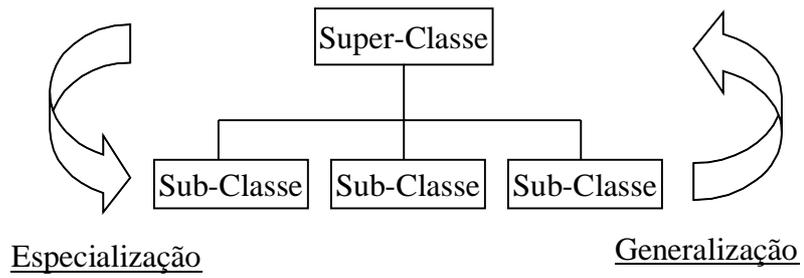
**Generalização:**

Classificação de um conjunto de objetos a partir de atributos e operações comuns

**Especialização:**

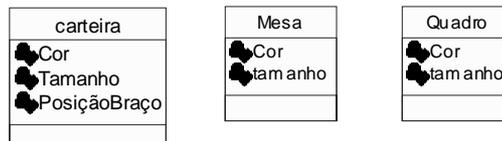
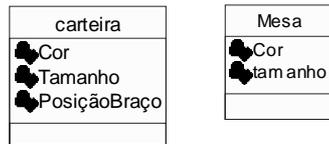
Separação de objetos de uma mesma classe a partir de atributos e operações mais específicas

## Exemplo

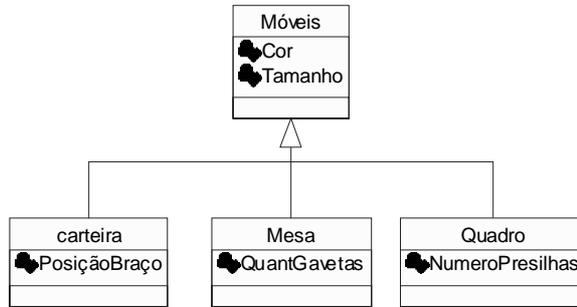


## Análise Orientada a Objetos Herança entre Classes

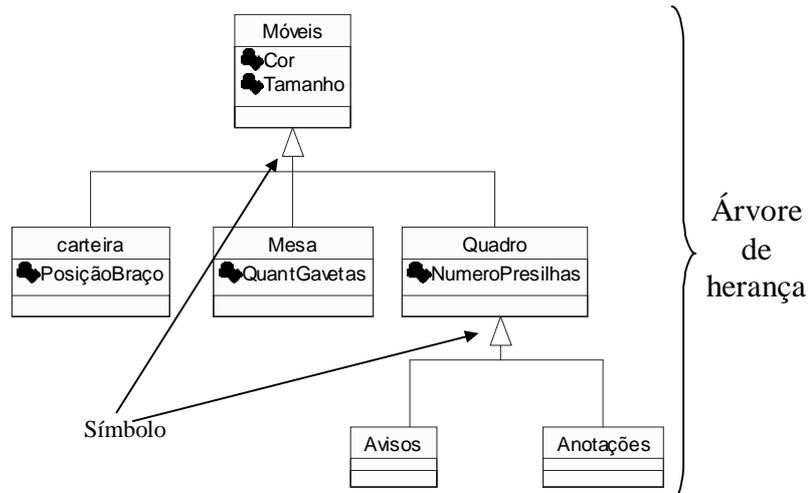




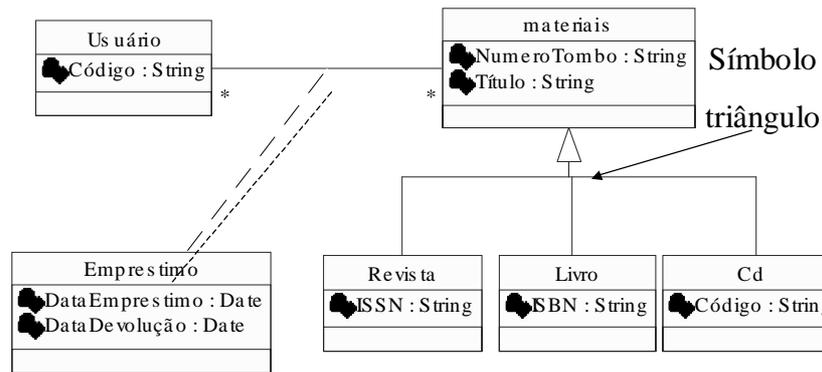
## Análise Orientada a Objetos Herança entre Classes



## Análise Orientada a Objetos Herança entre Classes



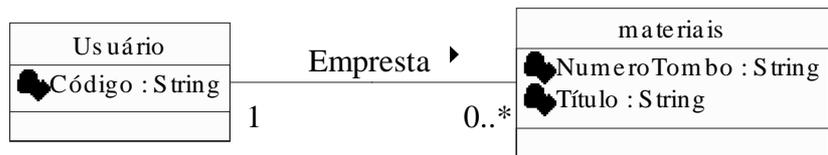
## Herança Outro Exemplo



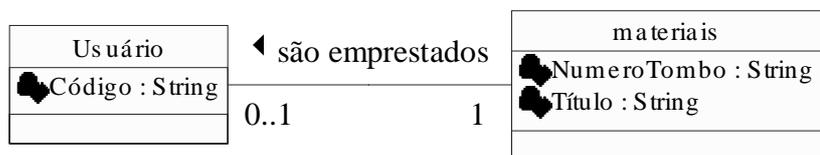
## Associação entre Classes

- ✓ Associação representa uma população variada de vínculos de relacionamento entre instâncias de uma classe.
- ✓ Exemplo: Classes usuário e materiais de um Biblioteca são associadas.
- ✓ Regras de Normalização podem ser aplicadas em Classes.
- ✓ Níveis de Multiplicidade Comuns:
  - 1 para 1
  - 1 para 1..N
  - 1 para Muitos
  - Muitos para Muitos, etc..

## Associação entre Classes Exemplo



## Associação entre Classes Exemplo

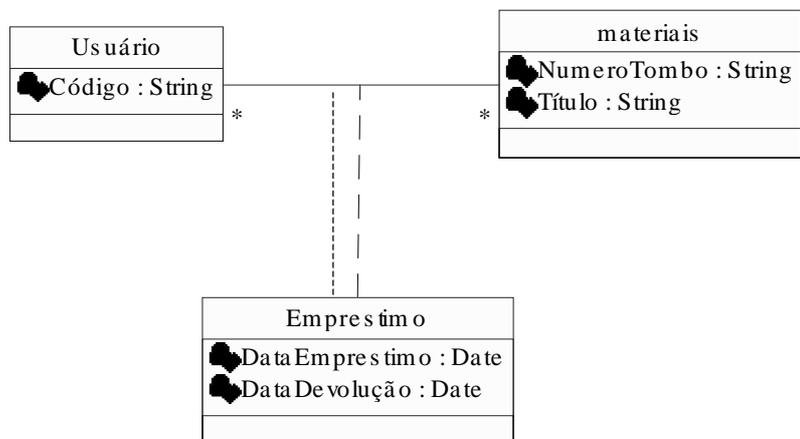


## Associação entre Classes Exemplo - Binárias

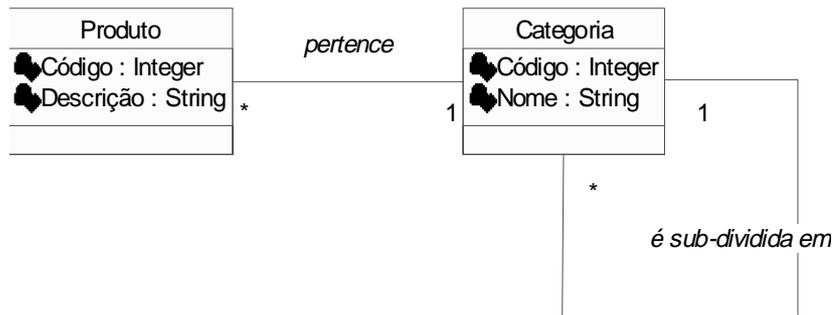


- Associação também pode ser feita através de uma classe
- Para isso, devem haver atributos da Associação entre as classes

## Associação entre Classes Exemplo (Ternárias)



## Associação entre a mesma Classes - Unárias



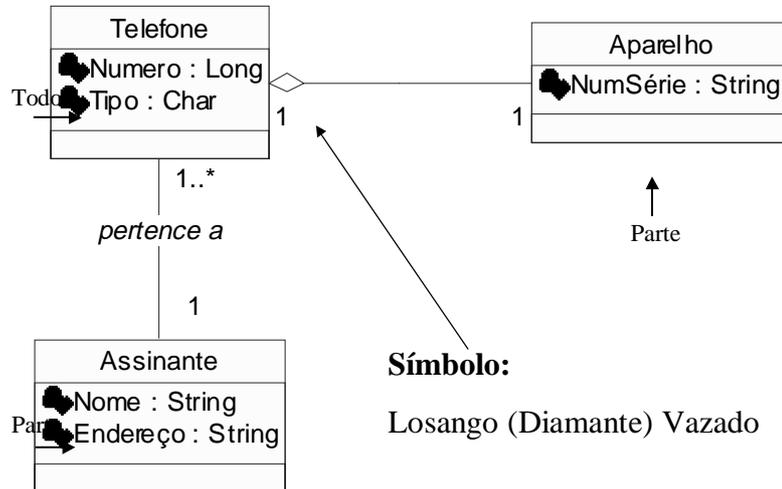
## Associação de Classes por Agregação

### Agregação:



- ✓ Agregação modela uma associação entre duas classes, do tipo todo-parte
- ✓ Quando juntas, as instâncias dessas classes têm um sentido orgânico, relacionado.
- ✓ Separadas, continuam existindo como unidades autônomas, podendo se associar com outras instâncias.
- ✓ Mantém características de multiplicidade

## Associação de Classes por Agregação



## Associação de Classes por Composição

Composição:



- ✓ Caso Particular de Agregação
- ✓ Representam duas classes que só possuem sentido quando estão associadas, ou seja, sempre cumprem um papel orgânico
- ✓ Se um instância da classes deixar de existir, todas as que estiverem agregadas por composição deixarão também de existir
- ✓ Algo similar com a “remoção” em cascata existente em Bancos de Dados Relacionais

## Associação de Classes por Composição

