

Tecnologia de Objetos

Conceitos

Princípios

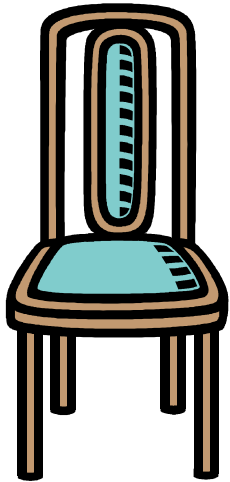
Modelagem

Copyright © 2020
Fábio Nogueira de Lucena
fabio@inf.ufg.br

O universo é orientado a objetos.

Cada objeto possui seu próprio estado

Custo
R\$70,00!

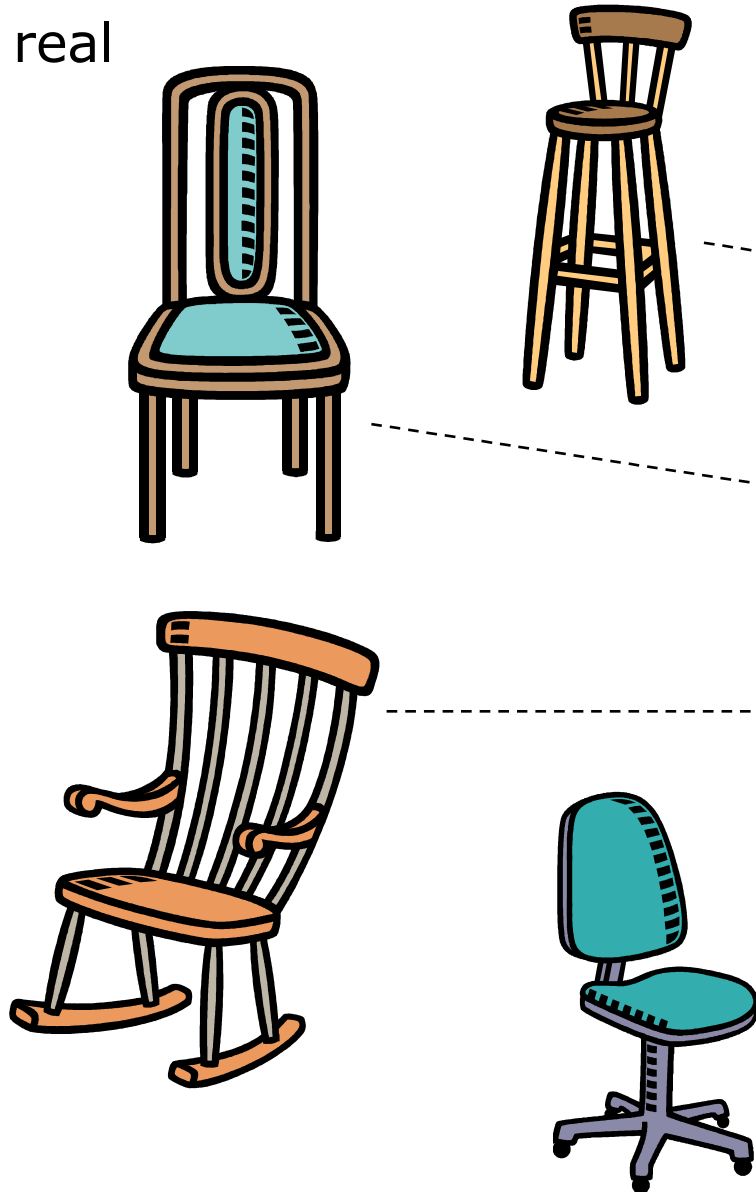


E quanto custo, será
que ele se interessa
por isso?



Noções de classe e objeto

Mundo real



Instâncias

c1 :
Cadeira

c2 :
Cadeira

c3 :
Cadeira

c4 :
Cadeira

Modelo de cadeiras do
do mundo real

Mundo virtual
(software)

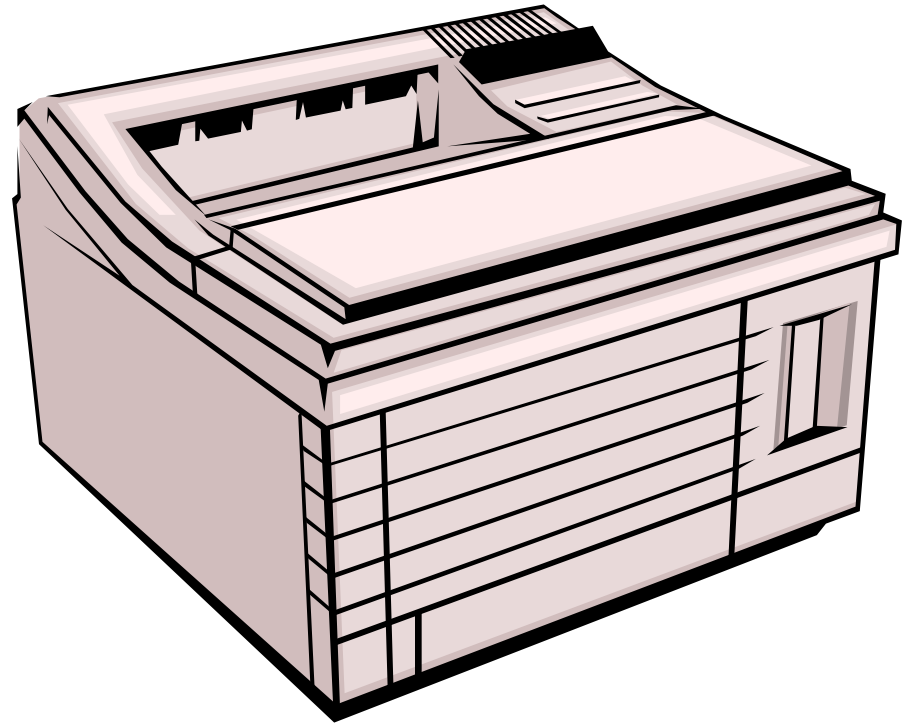
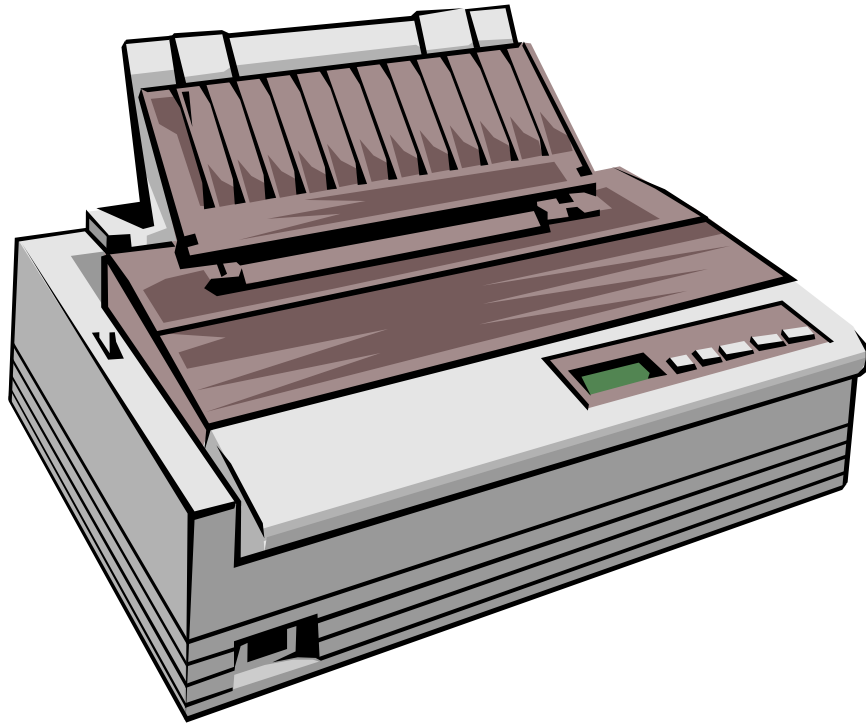
Classe

Cadeira

marca
fabricante
peso
carga
preço

Objetos compartilham comportamento

Não importa a marca, todas ...



Imprimem, geram página de teste, avançam página, ...

Métodos (descrevem o comportamento)

■ Classe **Impressora**

■ Métodos

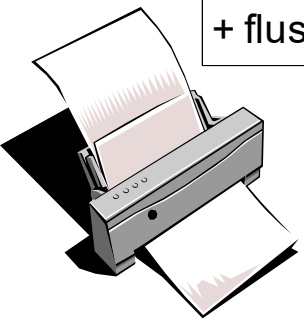
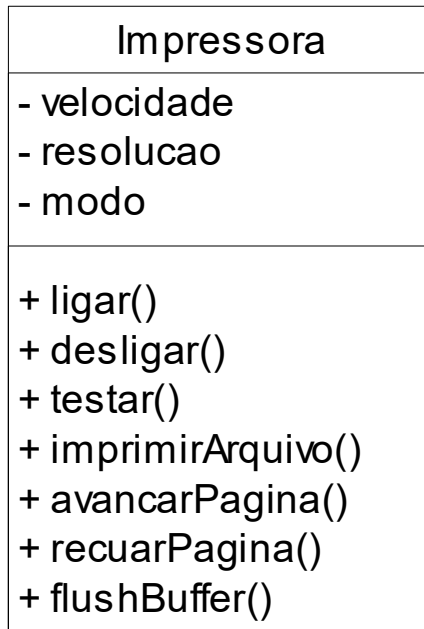
- Ligar/desligar
- Testar
- ImprimirArquivo
- AvancarPagina
- RecuparPagina
- FlushBuffer
- ...

Impressora	Nome da classe
- velocidade - resolucao - modo	Membros de dados
+ ligar() + desligar() + testar() + imprimirArquivo() + avancarPagina() + recuperarPagina() + flushBuffer()	Métodos

Representação na UML

Mensagem

Elemento inativo
(não recebe/envia mensagens)

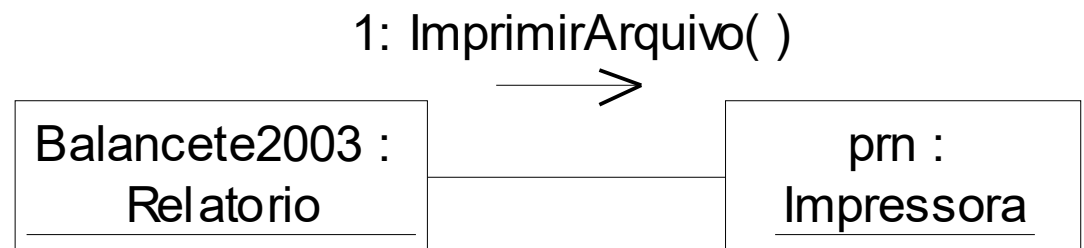


prn

Elemento dinâmico
(recebe/envia mensagens)

prn :
Impressora

Objeto **Balancete2003** envia
mensagem `imprimirArquivo`
para objeto **prn**



Representação na UML

Visão Orientada a Objetos

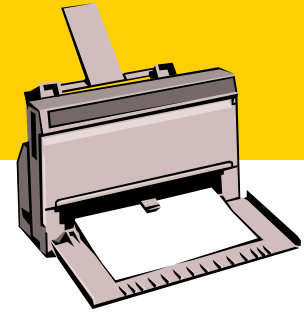
Conceitos básicos

- Objetos (instâncias de classes)
- Métodos (serviços oferecidos)
- Mensagens (requisição de serviços)

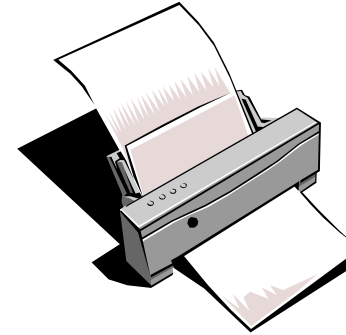
Métodos:
imprimir
teste
avancarPagina
...

Mensagem:
prn.imprimir(arquivoPDF)

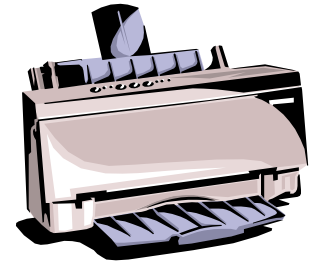
prn



Objetos



prn1

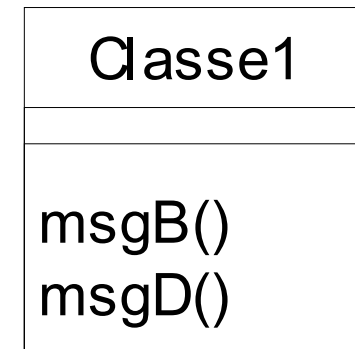
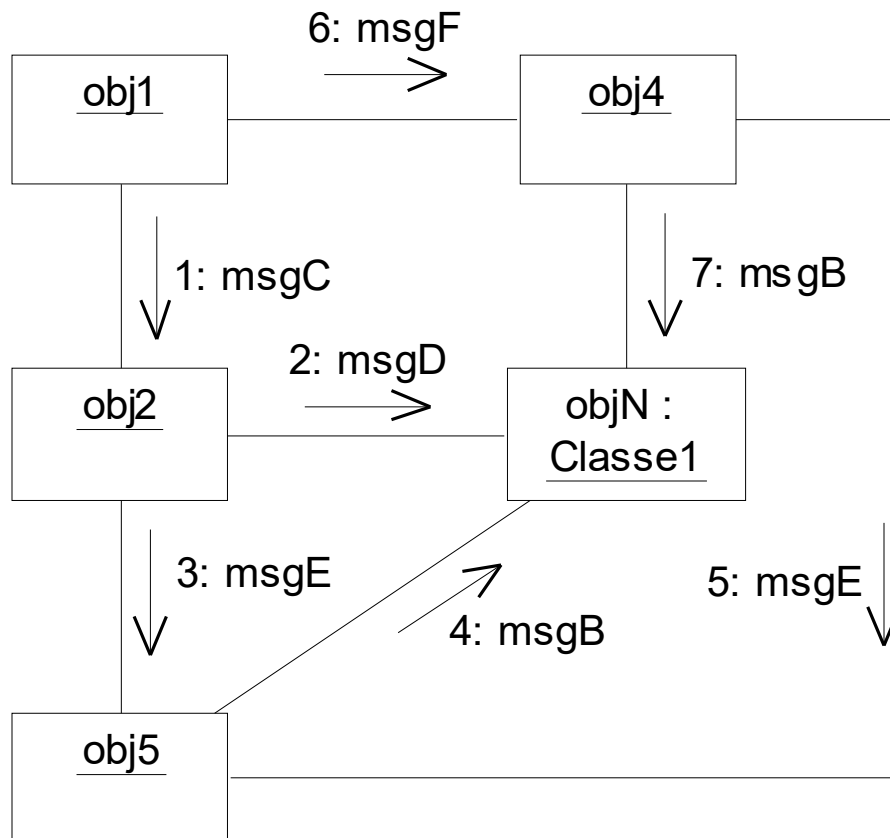


prn2

Classe:
Impressora

Aplicação orientada a objetos

- Aplicação OO é uma coleção de objetos que trocam mensagens entre eles



Classe do objeto **objN**

Cenário de troca de mensagens entre vários objetos

Desenvolvimento Orientado a Objetos

Como registrar?

Modelos precisam ser registrados

- Qual o material usado para imprimir fotografias no mundo OO?

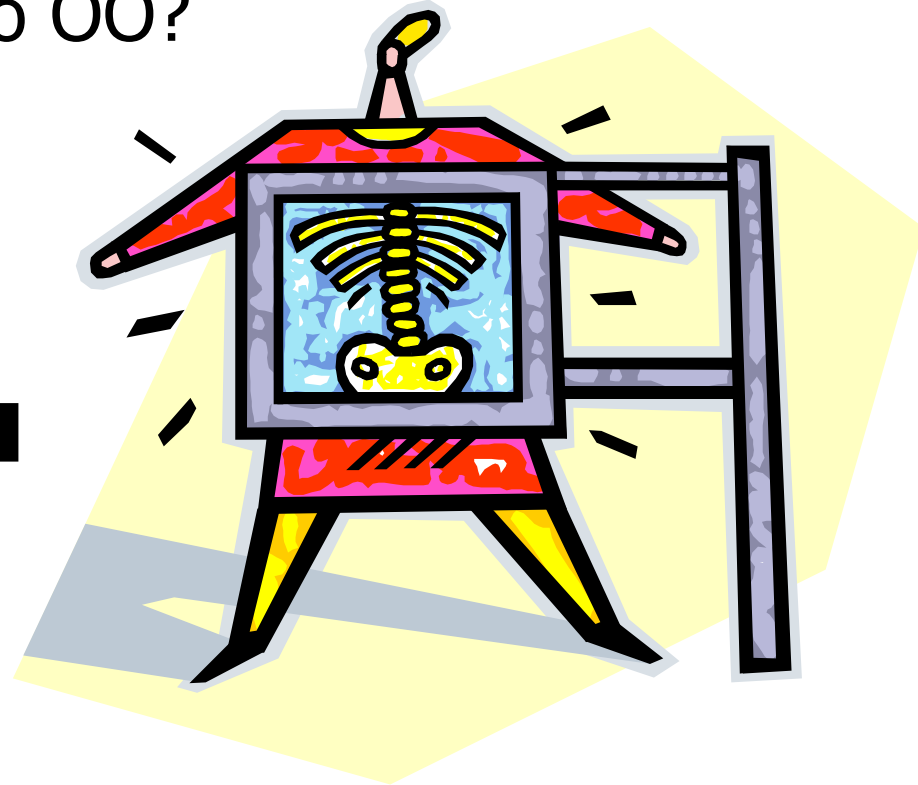
UML

Há outras propostas:

OML (Open Modeling Language)

Bon (Business Object Notation)

ROOM (Real-time object-oriented modeling), desenhos, ...



Diagramas “principais”

- **Diagrama de classes**
 - Classes
 - Relacionamento entre as classes
- **Diagrama de sequência**
 - Objetos
 - Troca de mensagens entre objetos

Interpretação de uma classe

- Pode se referir a um **Conceito** do mundo real




Amizade

- Entidade abstrata (software)

- **Especificação**

SGBD

- **Implementação** (representa uma classe em uma linguagem de programação)

Relógio
 alarme = 0
 setAlarme()  horaCorrente()

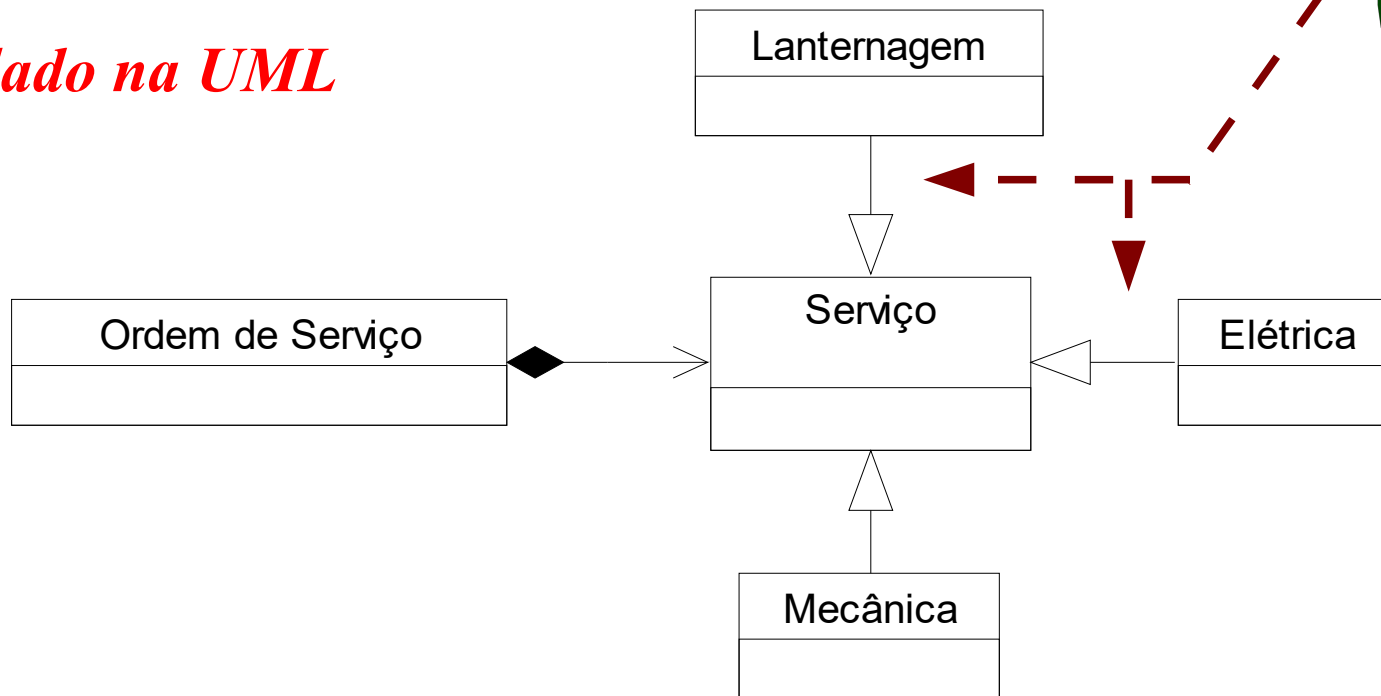
Exemplo (oficina mecânica)

Um pedaço do mundo ...

Uma ordem de serviço é uma composição de serviços.

Um serviço pode ser de Mecânica, Elétrica e Lanternagem.

Modelado na UML

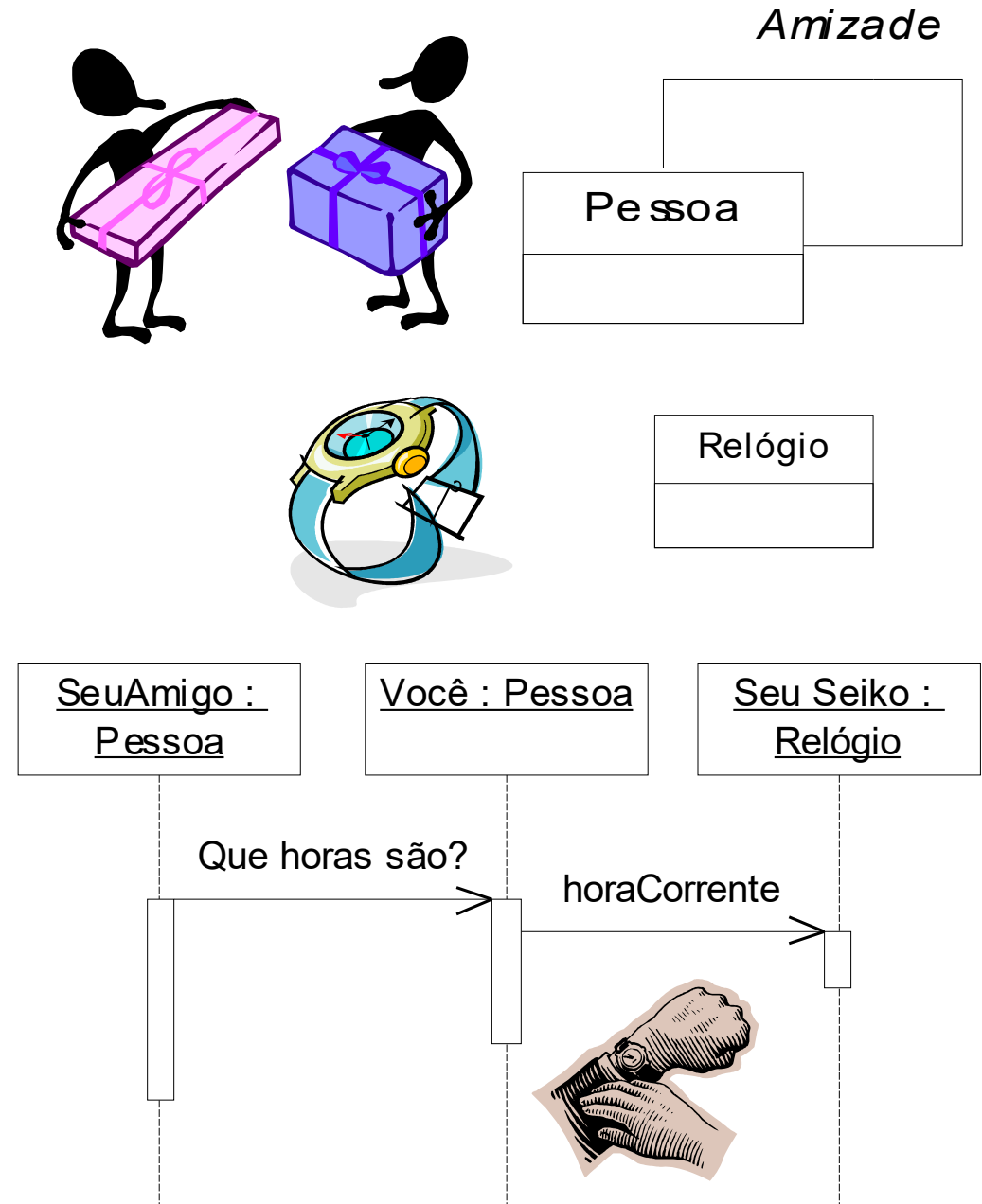


Herança

Forma de
relacionamento
entre classes.

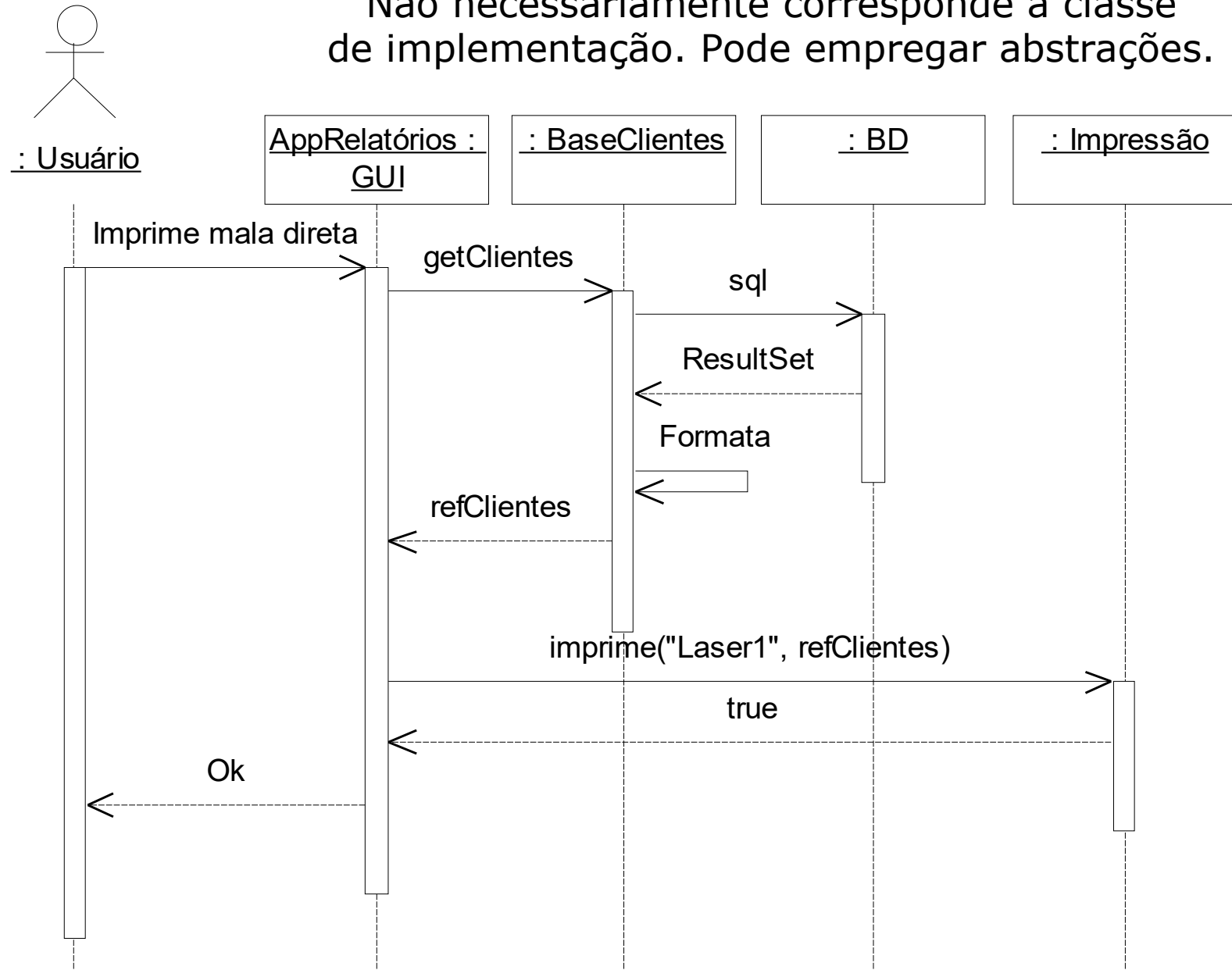
Exemplo (classes conceituais)

- **SeuAmigo** e **Você** são instâncias de `Pessoa`
- **Seu Seiko**: instância de `Relógio`
- Que horas são?: mensagem de **SeuAmigo** para **Você**
- `horaCorrente`: mensagem de **Você** para **Seu Seiko**



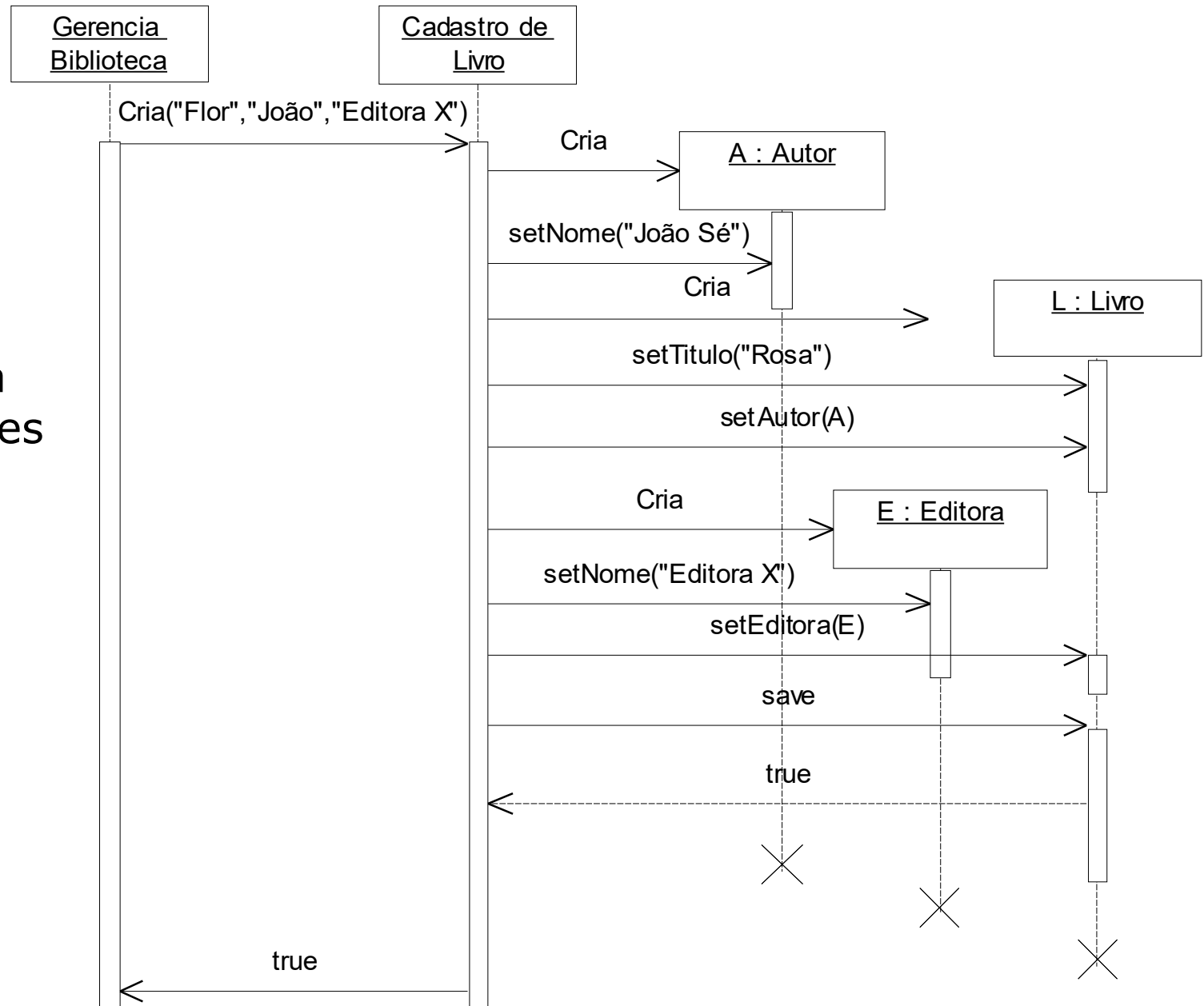
Exemplo (classes de especificação)

Não necessariamente corresponde à classe de implementação. Pode empregar abstrações.



Exemplo (classes de implementação)

Correspondência
direta com classes
em Java.



Tecnologia de Objetos

Solução dos problemas?

Problemas de desenvolvimento

Conhecimento
ingênuo de OO



Pressa em codificar



Falta de planejamento



Vantagens e desvantagens (OO)

■ Vantagens

- Melhor modularidade, manutenção e qualidade
- Modelos mais compreensíveis
- Maior estabilidade e flexibilidade
- **Transição mais suave da análise para o projeto**
- Reutilização de código mais efetiva
- Melhor comunicação

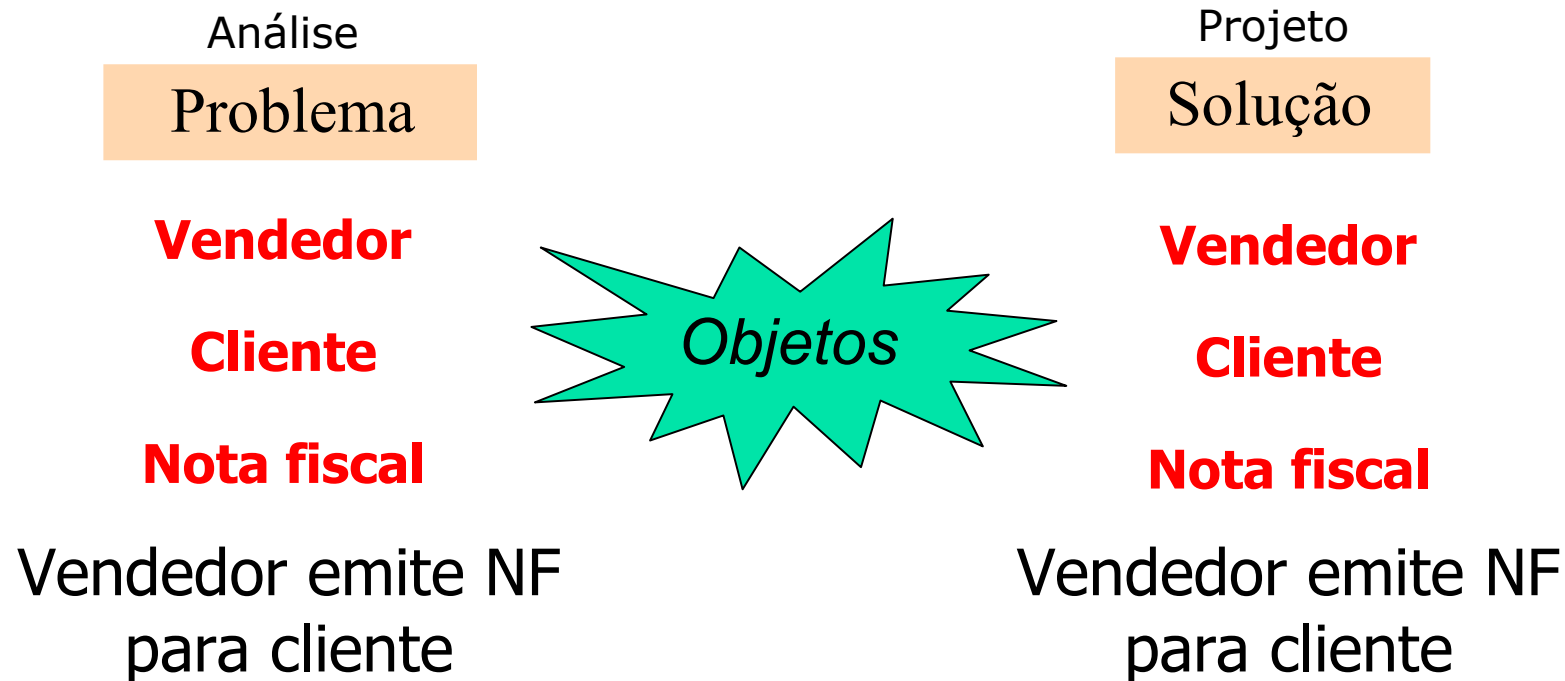
■ Desvantagens

- Desempenho inferior
- Não “decolaram” os SGDBs OO
- Maior tempo inicial de desenvolvimento
- Dificuldade em exhibir benefícios e de aprender OO

Crenças

Principal argumento favorável

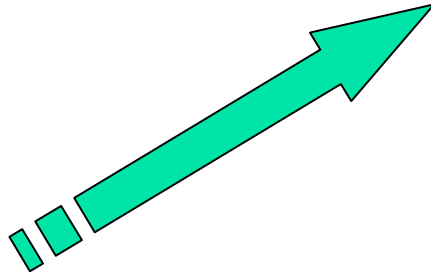
- Forma “mais natural e intuitiva”
- Modela tanto o problema quanto a solução como um coleção de objetos



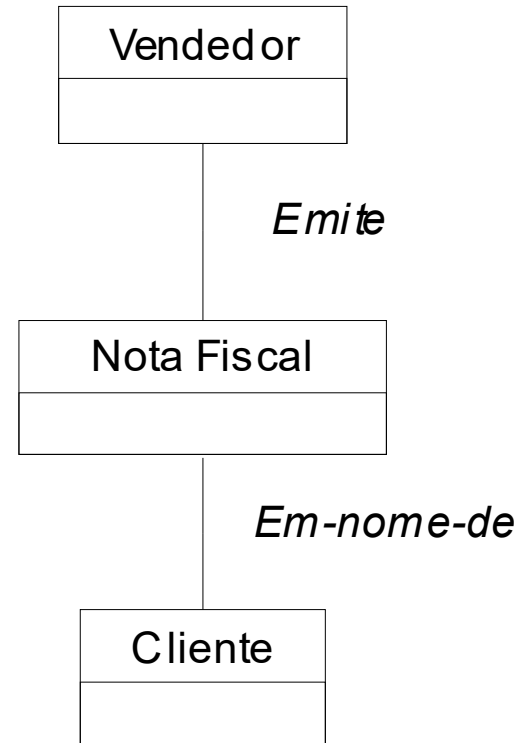
Representação mais conveniente

Problema
(negócio)

Vendedor emite NF
para cliente

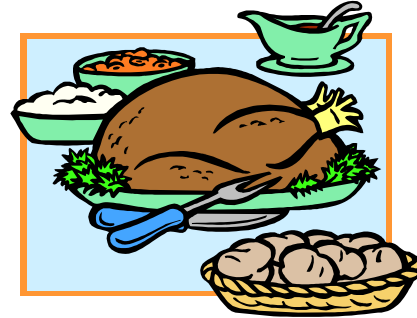


Visão OO
(UML)



Para pensar ...

Bom prato é obtido
com um bom fogão?



Bons instrumentos
fazem boa música?



**Emprego do
paradigma de objetos**



Não necessariamente

Sucesso



Neste ponto...

Talvez você não saiba muito sobre...
desenvolvimento de software
outras abordagens (p. ex.: estruturada)

Talvez você esteja confuso sobre...
os conceitos apresentados
sobre as afirmações do professor



E a disciplina oferece
os canais!!!!!!!!!!!!!!!!!!!!

<https://github.com/kyriosdata/oo>



Resumo

- Revisão da orientação a objetos
- “Principais recursos” da UML
 - Diagrama de classe e diagrama de seqüência
- Vantagens e dificuldades da OO
- Não perca...
 - O que é código orientado a objetos?
 - Como criar modelos orientados a objetos?
 - Como desenvolver software orientado a objetos?
 - Quais os relacionamentos entre classes?