

Tecnologia de Objetos

Conceitos

Princípios

Modelagem

1

Copyright © 2020

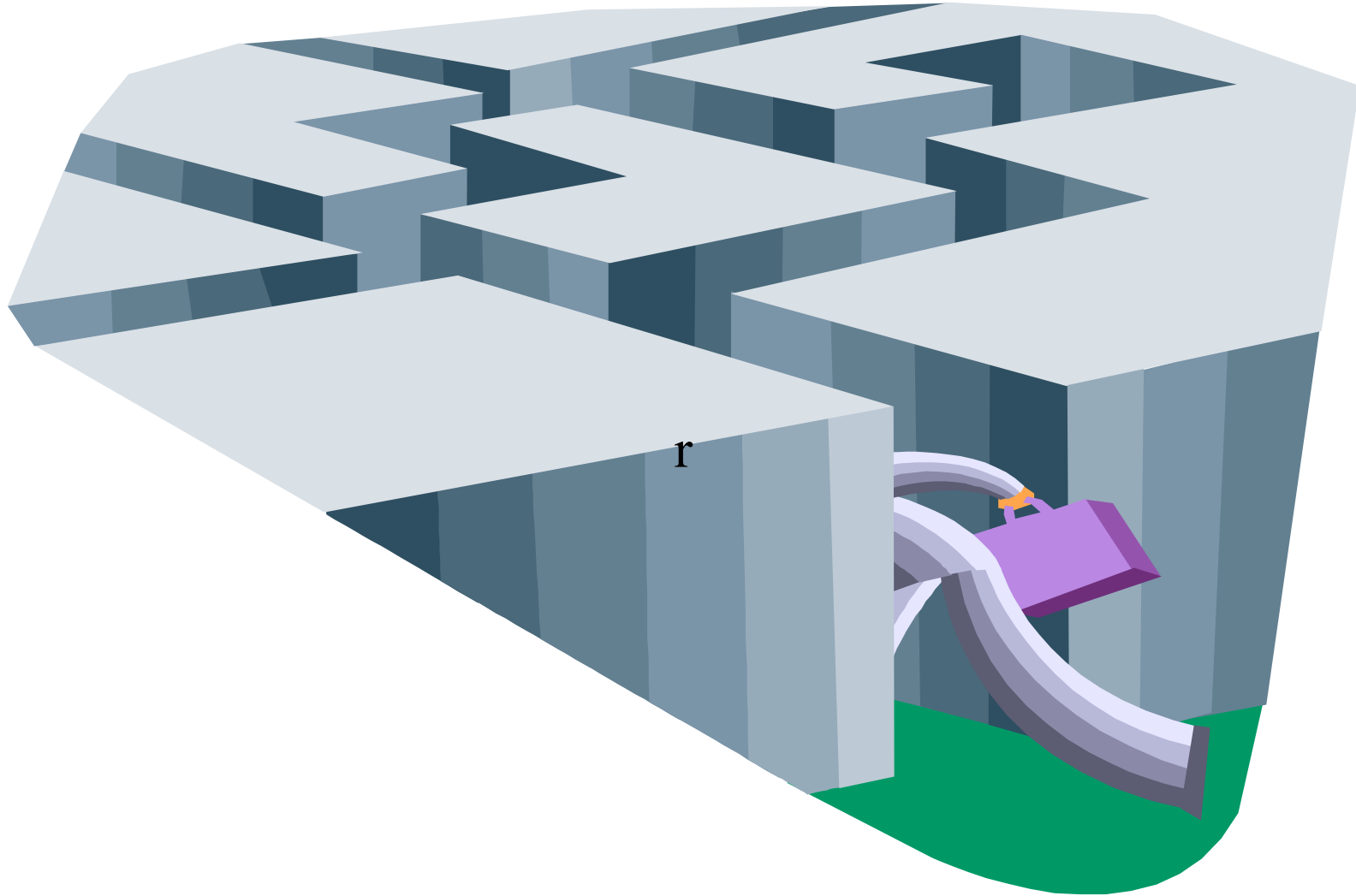
Fábio Nogueira de Lucena

fabio@inf.ufg.br

O universo é orientado a objetos.

Software é complexo

Naturalmente, nem todos. Considere aqueles que são!

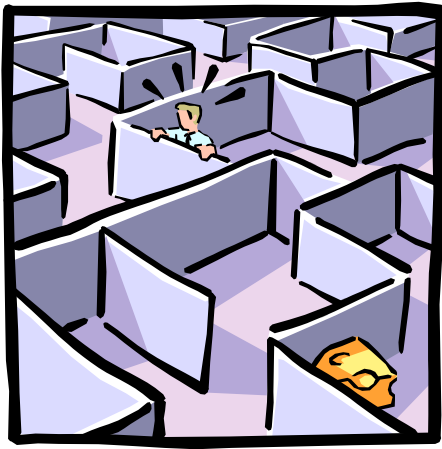


Como lidar com a complexidade?

Dividindo o problema em partes

Estruturado

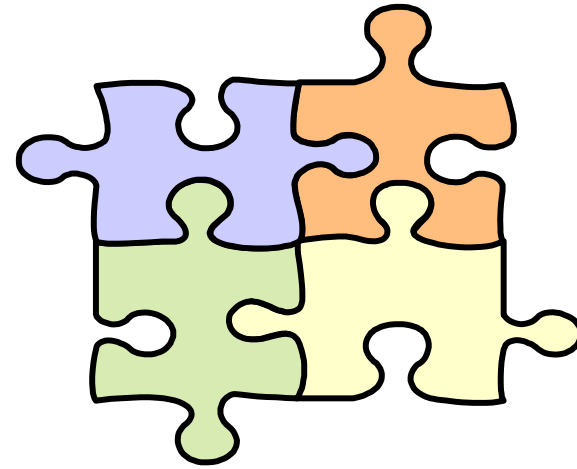
- Rotina é a unidade
- Muitas variáveis, funções



Ênfase em processos

Orientado a Objetos

- Objeto é unidade
- Une dados e funções

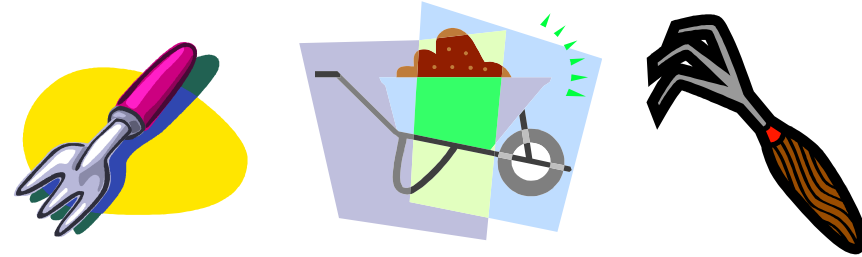


Ênfase em dados

Interpretação

Técnicas estruturas

Foco no código, na computação
Constante ênfase no software



Técnicas Orientadas a Objetos

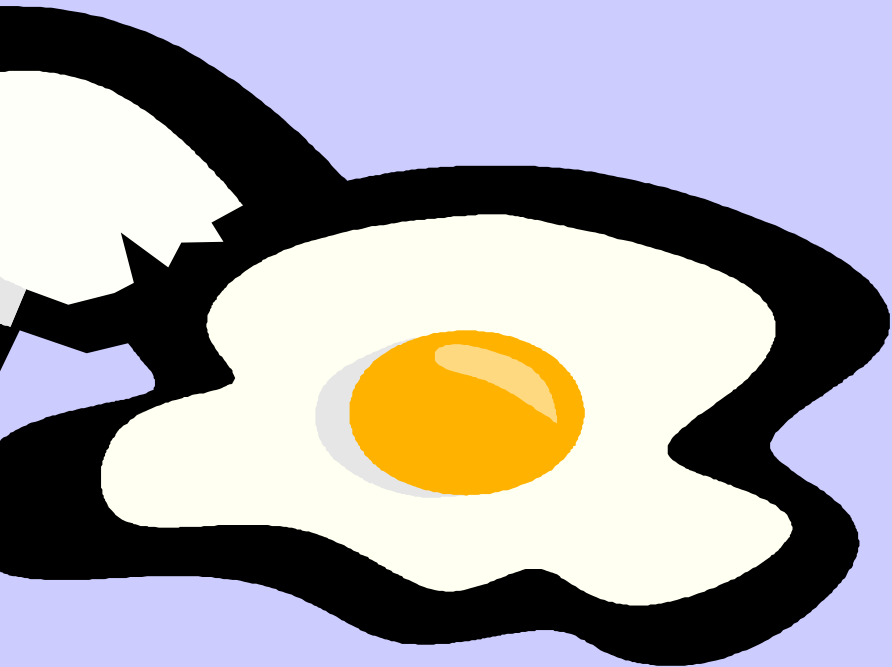
Foco no problema (mundo real)
Modelar o problema, especular, ...



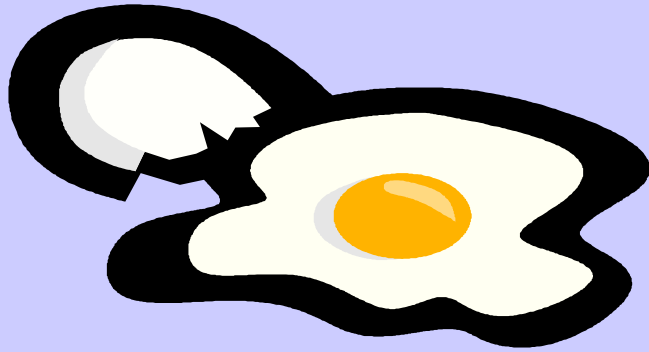
Object-Oriented Really Better Than Structured, Gary Warren King,
<http://eksl-www.cs.umass.edu/~gwking/whyoop.htm>

Dados são mais estáveis que processos
Facilita reutilização, maior abstração, ...

Como você frita um ovo?



Como você frita um ovo?



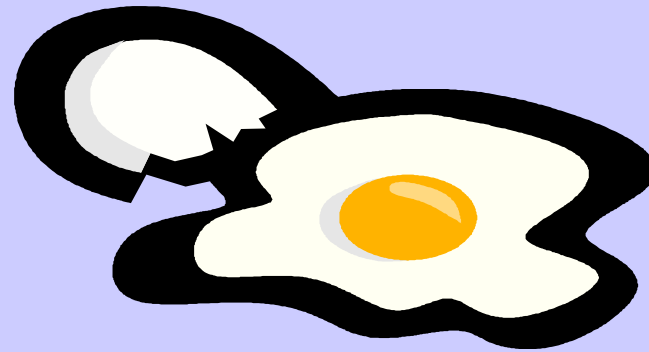
Se você respondeu algo como ...

1. Pego o ovo
2. Quebro o ovo e o deposito em uma panela
3. Levo a panela com óleo e o ovo ao fogo
4. Aguardo até que fique “bom”

**Então você frita ovos de forma
algorítmica, estruturada!**

Há algum problema?

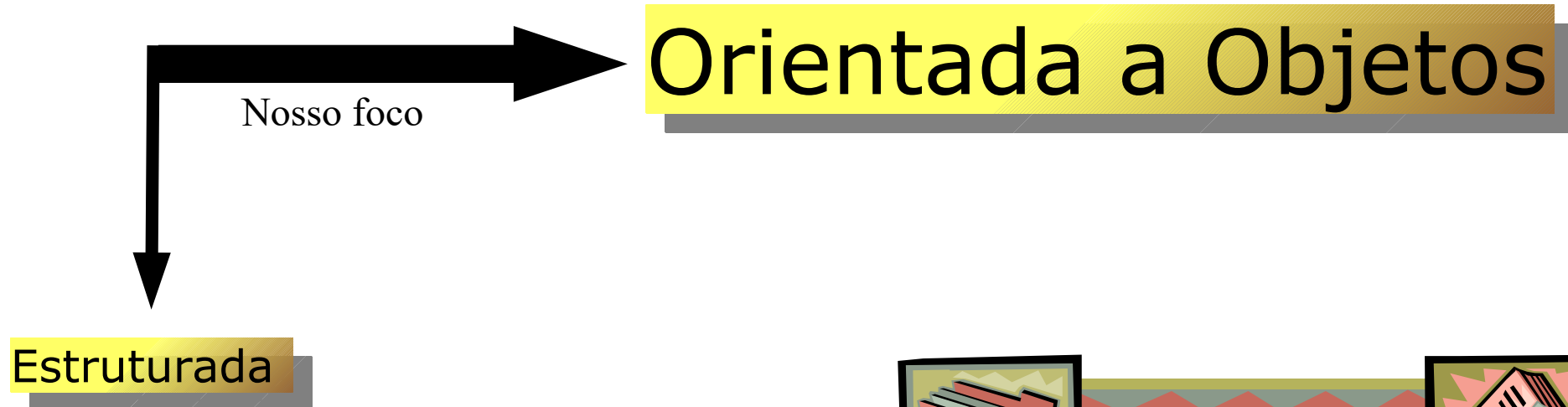
*Fritar um ovo de forma
orientada a objetos exige ...*



Domínio da tecnologia
Orientação a Objetos

Abordagem a ser empregada:

Problemas que não são complexos não precisam ser decompostos em partes.
Não precisam de análise e projeto estruturados ou orientados a objetos.



*Você ou sua empresa
provavelmente usa
ou já usou
esta abordagem!*

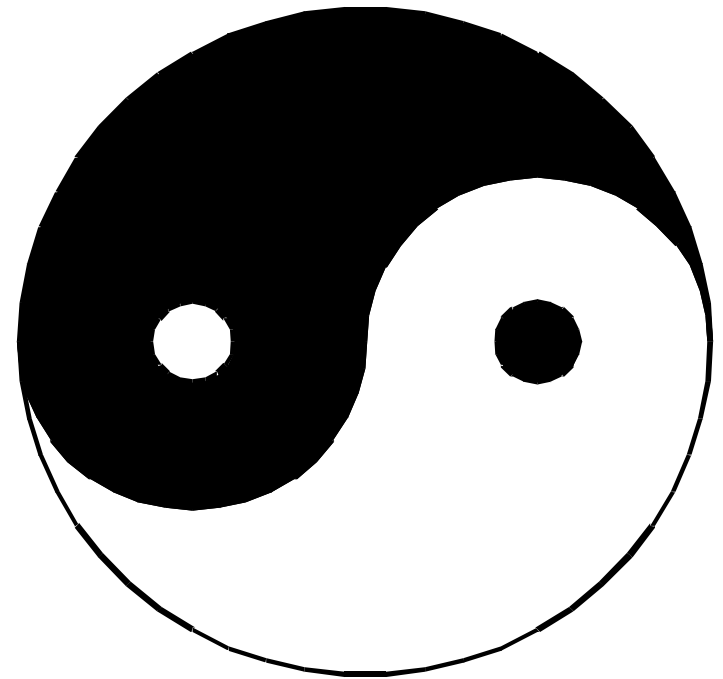


Objeto = dados + funções

*Não é apenas dados
Não é apenas funções*

***É uma unidade
formada pela união
de dados e funções***

Um objeto não é uma rotina!



Yin/Yang

Rotina em Visual Basic

```
Public Function Volume() As Double  
    Return x * y * z  
End Function
```

*Uma é boa,
20 é melhor ainda,
mas 300 é demais!*

Estruturada x Orientada a Objetos

- Ênfase em processos, funções, rotinas (ESTRUTURADO)
- Ênfase em dados e processos (ORIENTADO A OBJETOS)

Rotina, função, procedimento

```
Public Sub Volume() As Double
    Volume = x * y * z
End Sub
```

*Pense nos
processos!*

*Pense nas
“coisas” e nos
processos que as
manipulam!*

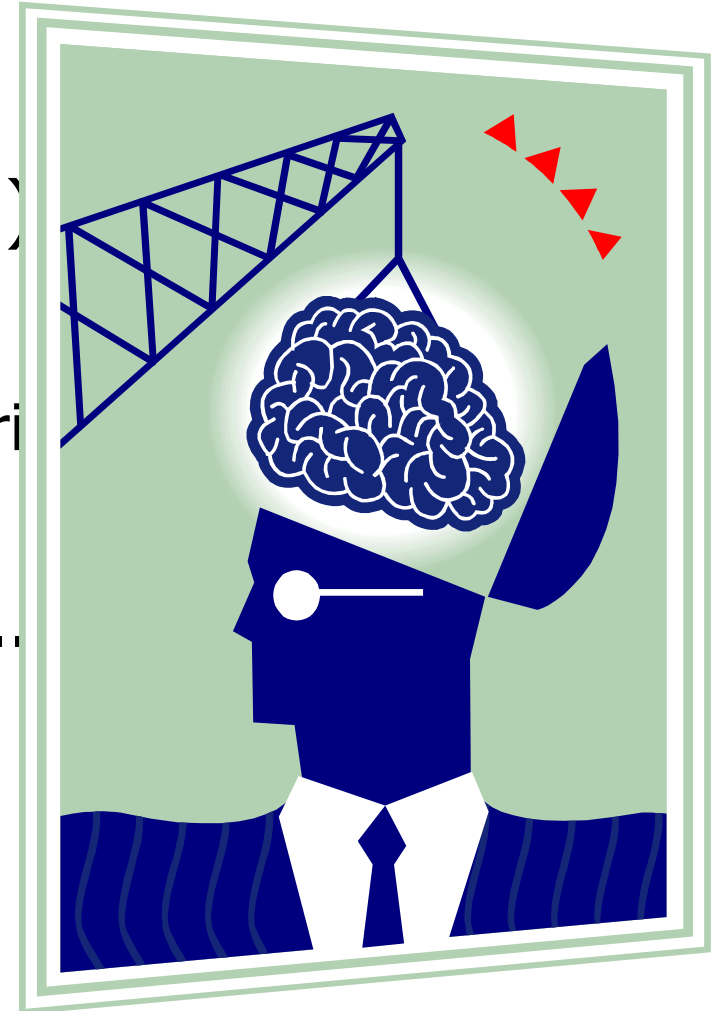
Orientado a Objeto

```
Public Class Solido
    Inherits FiguraGeometrica
    Private x As Double = 0
    Private y As Double = 0
    Private z As Double = 0

    Public Function Volume() As Double
        return x * y * z
    End Function
End Class
```

Abstração crescente

- Padrões de bits (01001111011...)
- Assembly (ADD %eax, %ebx)
- Procedimentos (Volume, slide anterior)
- **Orientação a objetos**
classes, herança, polimorfismo,...

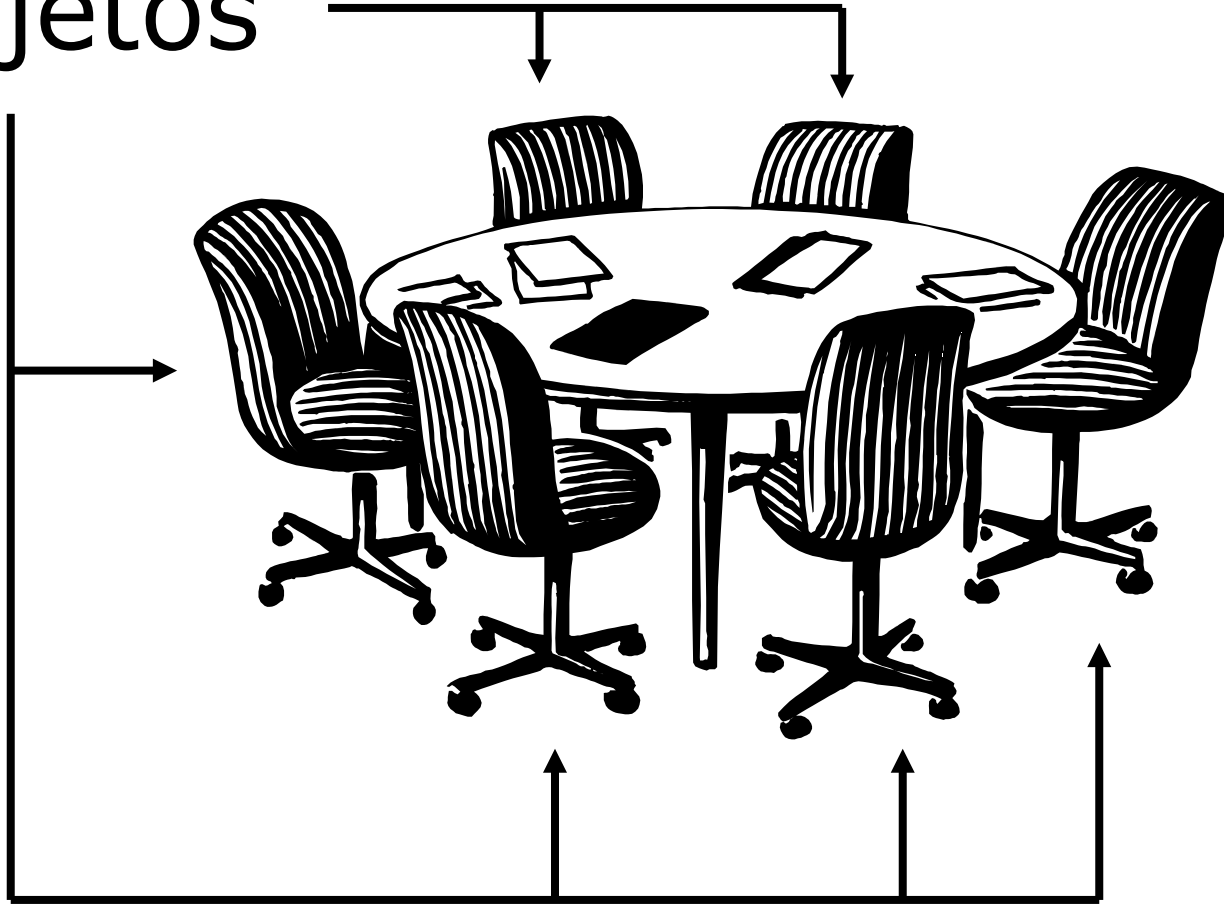


Visão orientada a objetos ("reunião")



Noções de classe e objeto

Objetos



Por simplicidade, consideremos apenas as cadeiras

Noções de classe e objeto

- Indústria X
Cadeiras CostaFlex
Custo: R\$190,00
Peso: 5,6 Kg
Carga: 150 Kg

Detalhes relevantes de cada uma das cadeiras

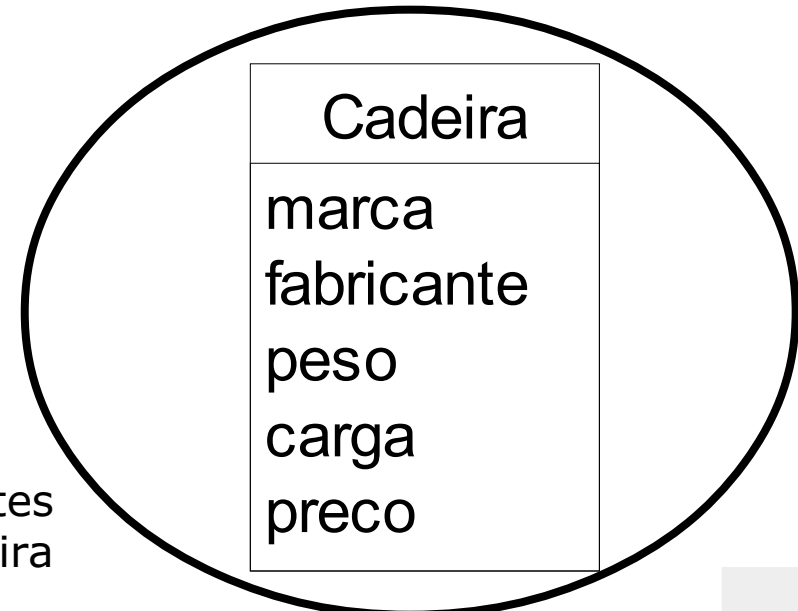


Mundo real

Mundo do software

- Software
Classe Cadeira

Características relevantes
a serem registradas para cada cadeira



Noções de classe e objeto

Caso 1



Caso 2



Mundo real

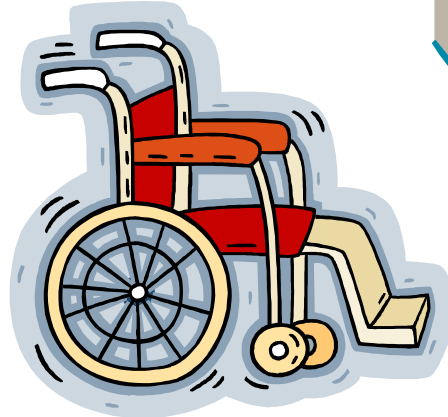
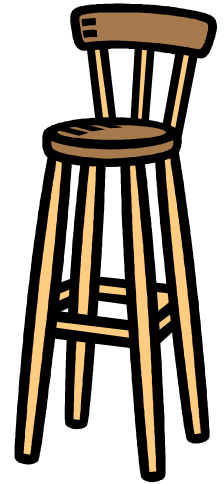
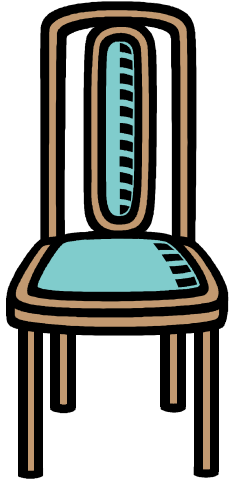
Mundo do software

Classe Cadeira satisfaz
os casos reais (ambos)!

Cadeira
marca
fabricante
peso
carga
preco

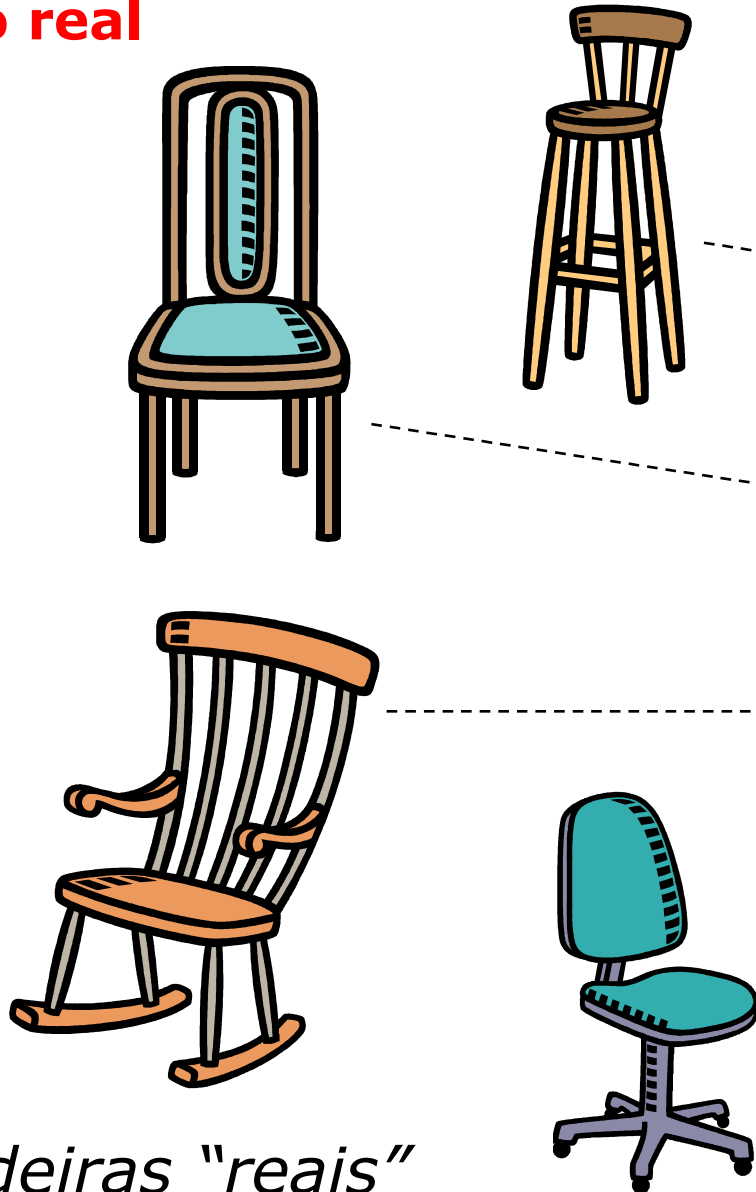
Noções de classe e objeto

Objetos do mundo real (cadeiras)



Noções de classe e objeto

Mundo real



Cadeiras "reais"

Instâncias (ou objetos)

c1 :
Cadeira

c2 :
Cadeira

c3 :
Cadeira

c4 :
Cadeira

Modelo de cadeiras

Mundo virtual (software)

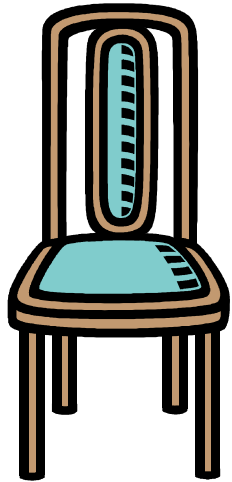
Classe

Cadeira

marca
fabricante
peso
carga
preço

Do real para o orientado a objeto...

Abaixo é o que você vê!
Seja isto o "mundo real"



*Ligação imaginária entre
o real e o que é software*

Só existe na nossa mente!

**Mundo virtual
(software)**

Instância
(objeto)

c2 :
Cadeira

Classe

Cadeira

marca
fabricante
peso
carga
preço

Modelo de cadeira

Noções de classe e objeto

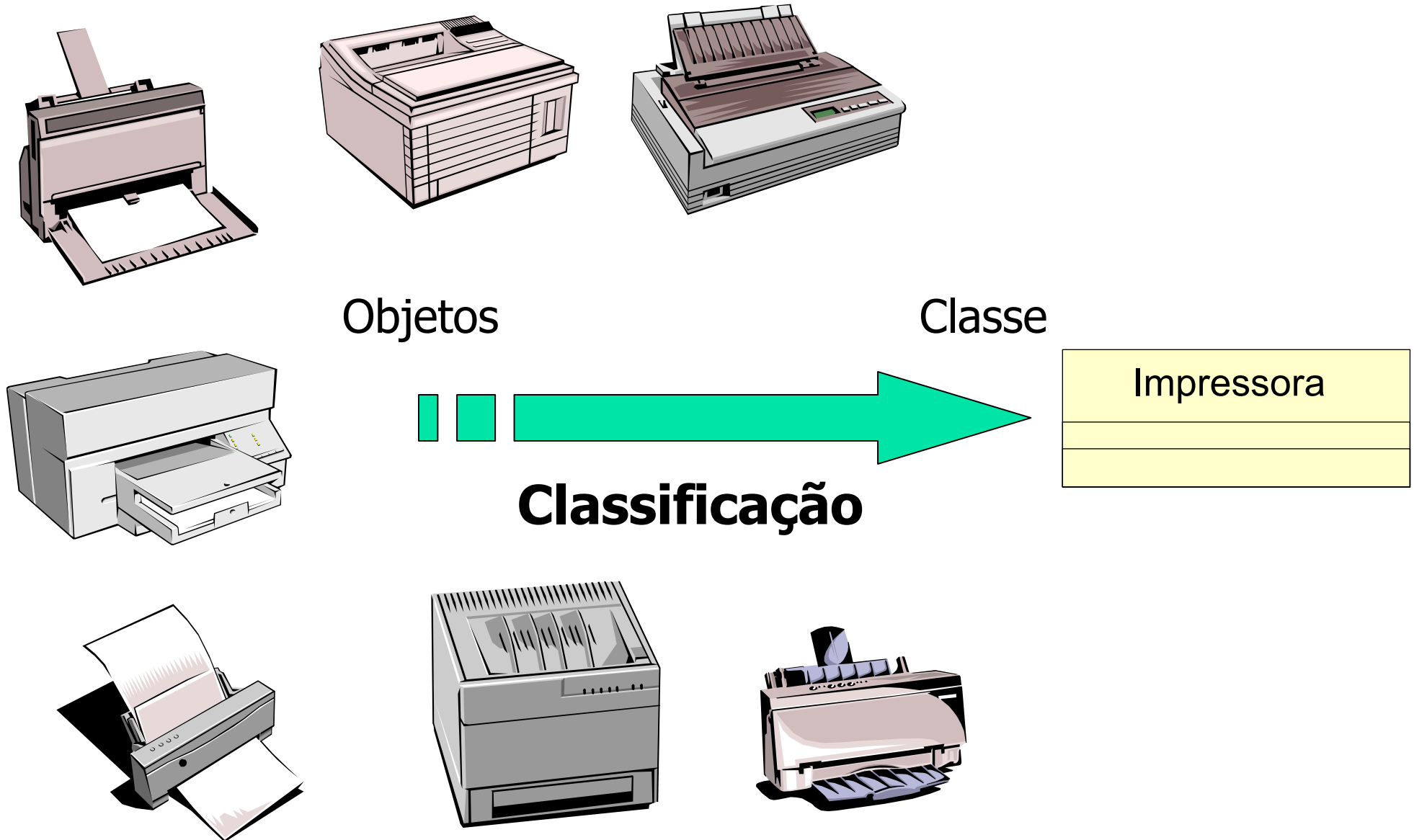


Brinquedos do mundo real
modelados como instâncias
da classe Brinquedo

Brinquedo
preço
idade
garantia
fabricante
nome

"Mundo do Software"

Noções de classe e objeto



Noções de classe e objeto

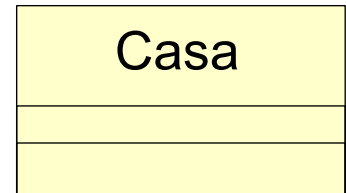


Objetos

Classe



Classificação

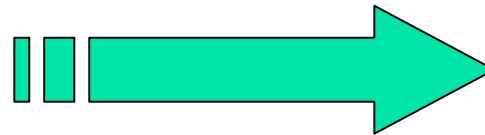


Noções de classe e objeto



Objetos

Classe



Classificação

Postura Religiosa



Noções de classe e objeto



Modela

(representa, retrata, facilita a manipulação)

Noções de classe e objeto

Palácio da Alvorada

Palácio

Classe
conceitual.

Noções de classe e objeto

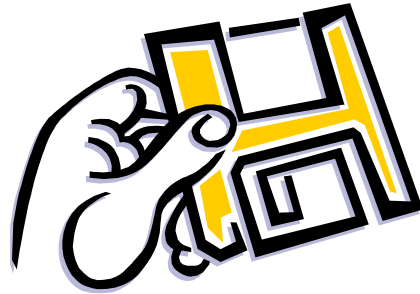
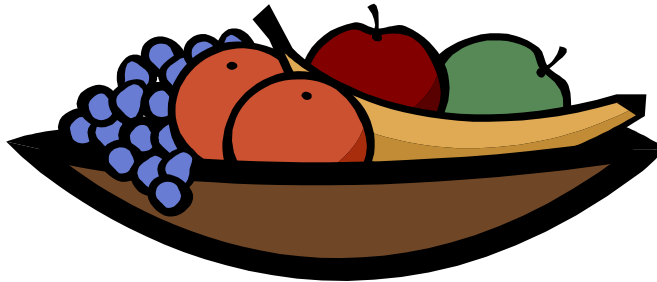
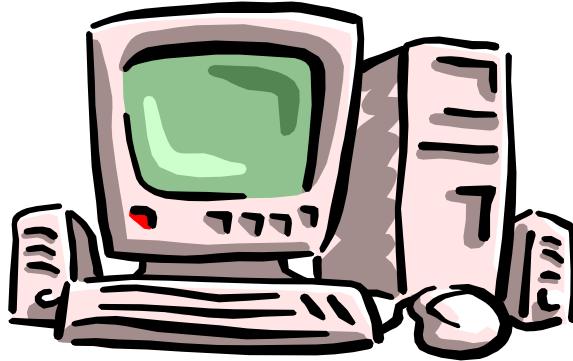
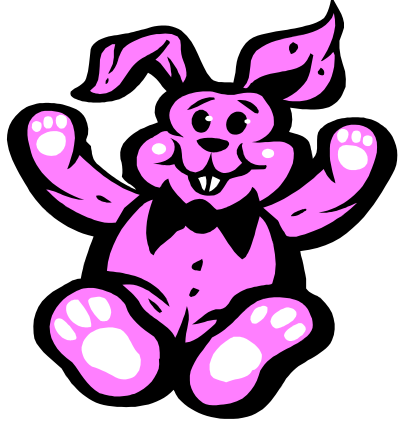
Instância *FHC* de Presidente

Presidente

Instância
Lula da
classe
Presidente



Noções de classe e objeto



Mundo real

Representação
(modelagem) da
perspectiva OO

*Todos são instâncias
da classe Objeto*

Objeto
nome

Classificação (noções de classe)



Mundo real

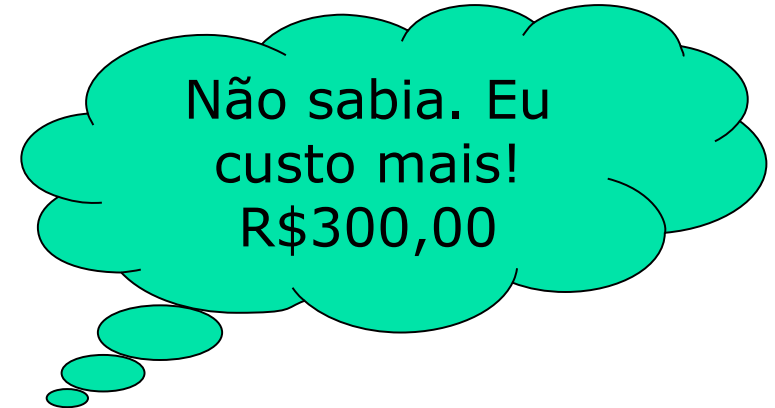
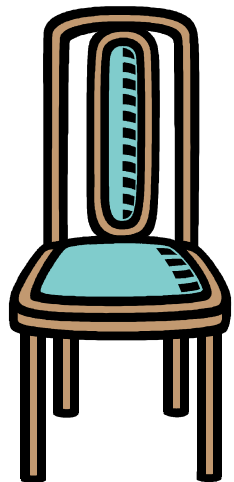
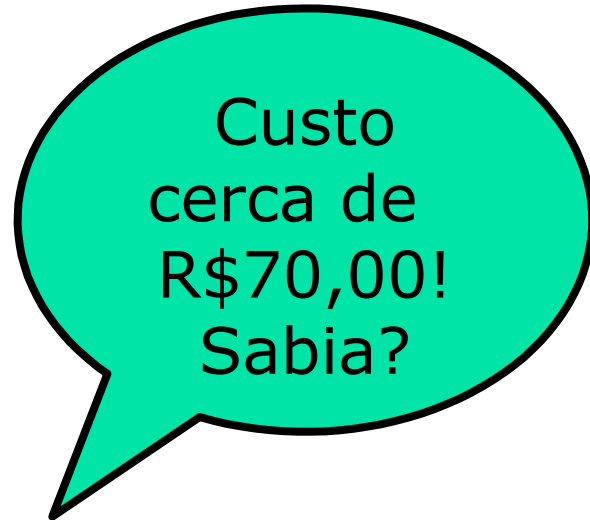


Há algo nas figuras em comum:
ambas ilustram coisas que estão se
deslocando. Talvez queiramos capturar
isso e, dessa forma, ...

Ambos se deslocam de um ponto
A até um ponto B, ambos de interesse.

Deslocamento
velocidade

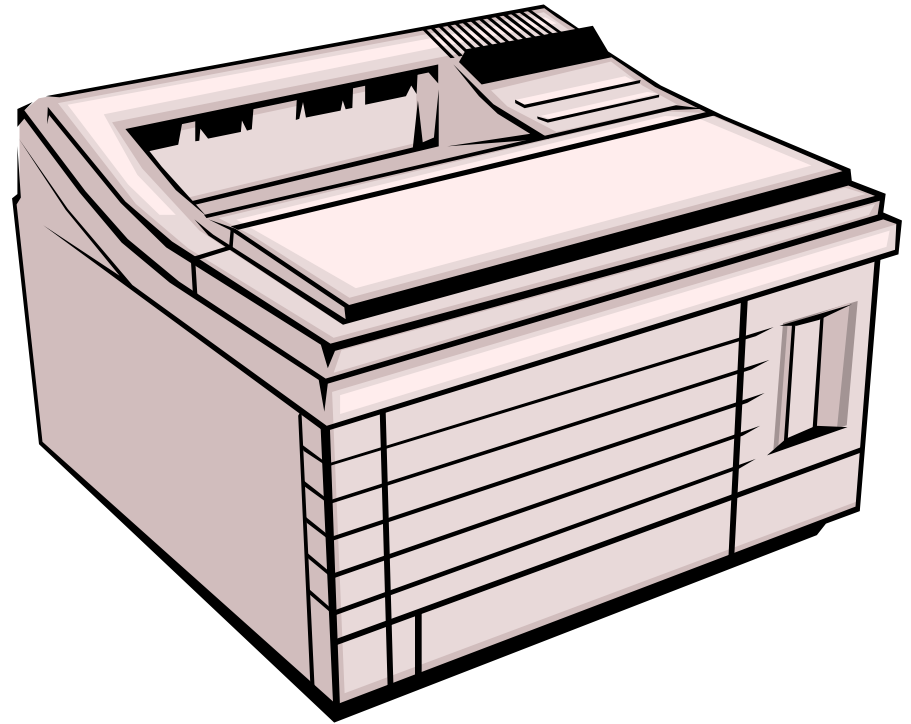
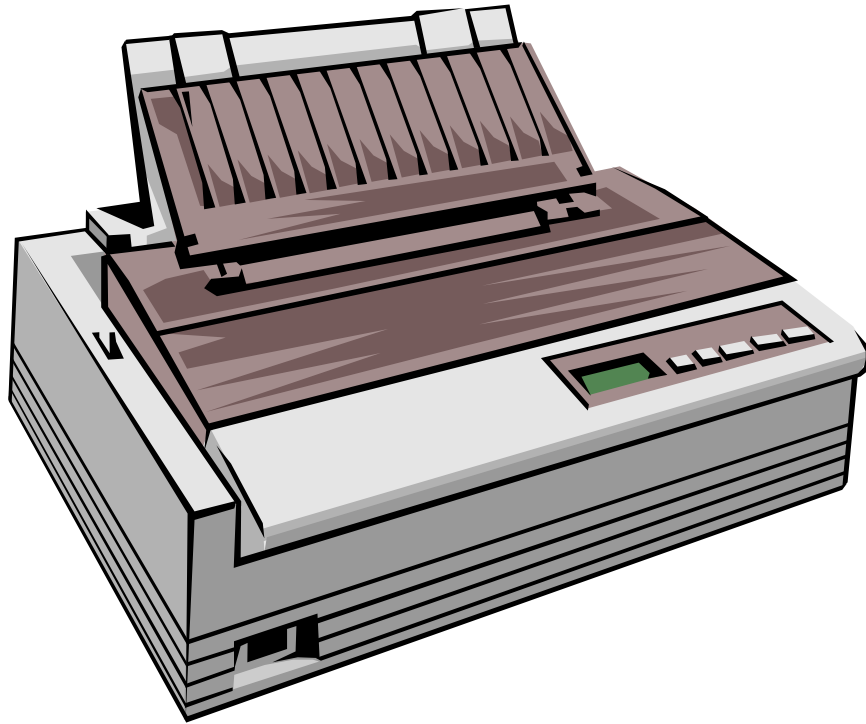
Cada objeto possui seu próprio estado



Cadeira
marca
fabricante
peso
carga
preço

Objetos compartilham comportamento

Não importa a marca, todas ...



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

Métodos (comportamento)

■ Classe **Impressora**

■ Métodos

- ligar/desligar
- testar
- imprimirArquivo
- avancarPagina
- recuarPagina
- flushBuffer
- ...

Impressora
- velocidade - resolucao - modo
+ ligar() + desligar() + testar() + imprimirArquivo() + avancarPagina() + recuarPagina() + flushBuffer()

Nome da
classe

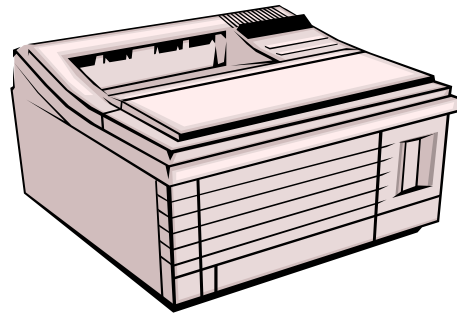
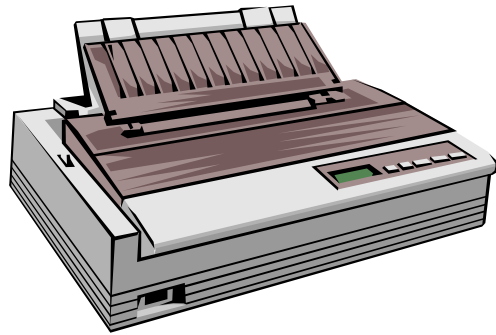
Atributos

Métodos

Representação na UML

Objetos reais e em software

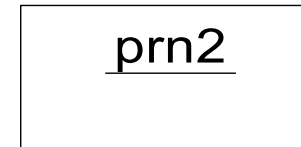
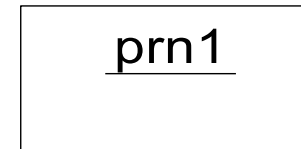
Mundo real



Instâncias de impressoras
(objetos reais)

Ligação imaginária
(entre o mundo real e a
correspondente representação
no mundo OO)

Mundo virtual



Instâncias da classe Impressora
(objetos de software)
Representação UML de objetos
identificados por prn1 e prn2

Funcionários

Não importa a atividade, todos ...



Recebem um salário, tiram férias, executam tarefas, ...

Métodos (descrevem o comportamento)

- Classe **Funcionario**

- Métodos

- getSalario
- getFerias
- getTarefas
- getLocacao

Funcionario
- codigo
+ getSalario() + getFerias() + getAtividades() + getLotacao()

Nome da
classe

Atributos

Métodos

Representação na UML

Funcionários reais e em software



Mundo real

Fronteira imaginária
(existe nas nossas cabeças)

f1

f2

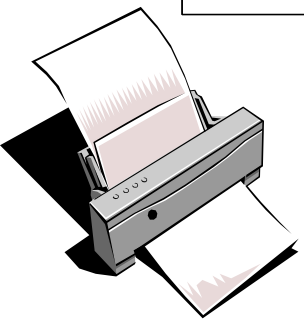
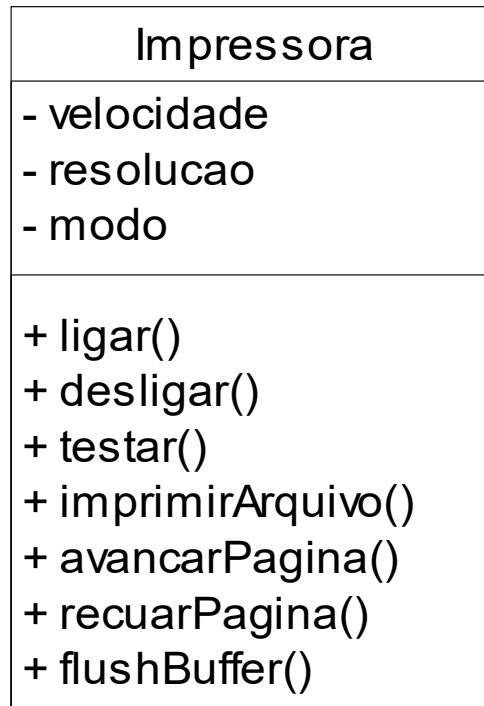


Mundo virtual

f3

Mensagem (prn instância de Impressora)

Elemento inativo
(não recebe/envia mensagens)
(classe Impressora)



prn

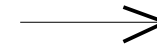
Elemento dinâmico
(recebe/envia mensagens)

prn :
Impressora

Um objeto da classe Relatorio
envia mensagem
imprimirArquivo para objeto
prn

1: ImprimirArquivo()

Balancete2003 :
Relatorio

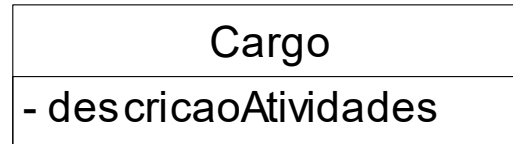


prn :
Impressora

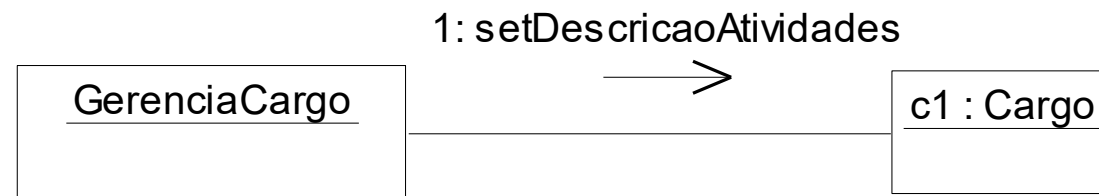
Representação na UML

Mensagem (detalhes)

Análise: 3min55s de conversa com profissional de RH (via telefone)



Projeto orientado a objetos: 2min



Significado:

1. GerenciaCargo **envia mensagem** setDescricaoAtividades **para** Cargo
2. GerenciaCargo **aguarda retorno** da mensagem.
3. GerenciaCargo **prossegue sua execução** após retorno.

Visão Orientada a Objetos

Conceitos básicos

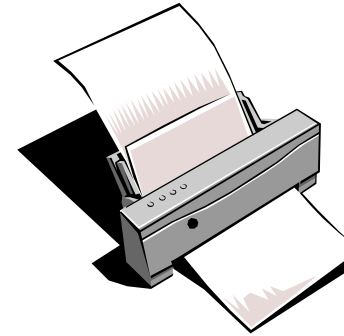
- Objetos (instâncias de classes)
- Métodos (serviços oferecidos)
- Mensagens (requisição de serviços)

Classe:
Impressora

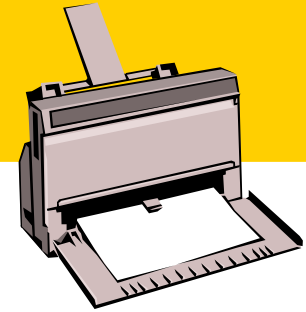
Métodos:
imprimir
teste
avancarPagina
...

Atributos:
velocidade
resolucao
modo

Mensagem:
prn.imprimirArquivo()

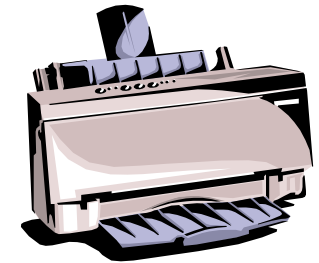


prn1



prn

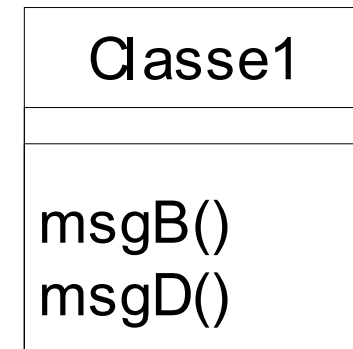
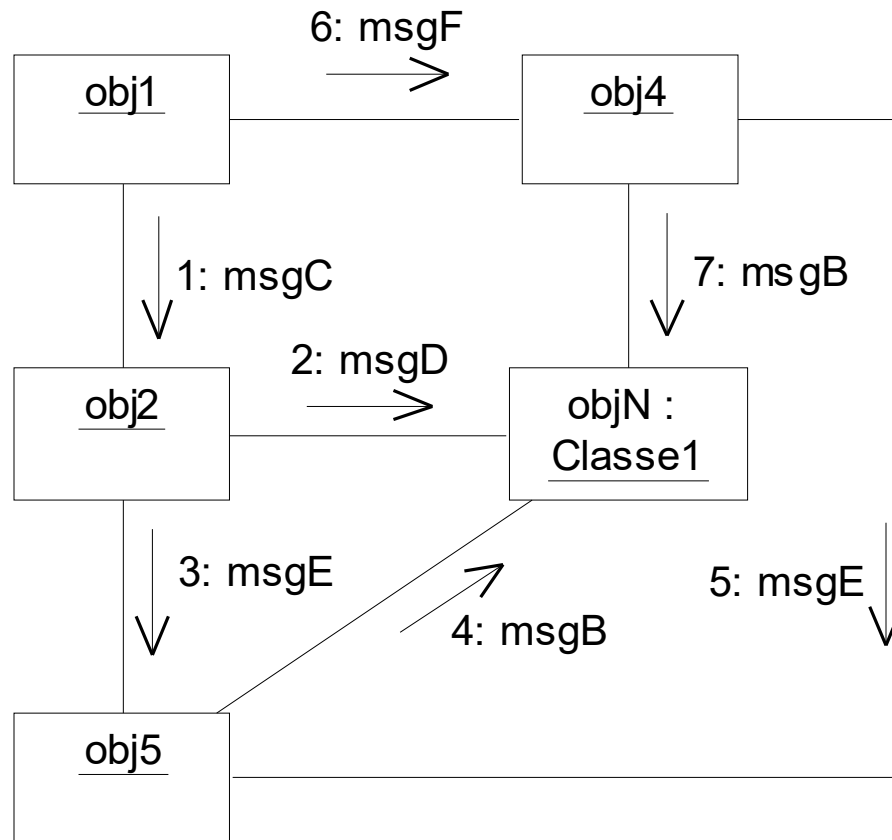
Objetos



prn2

Aplicação orientada a objetos

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



Classe do objeto **objN**

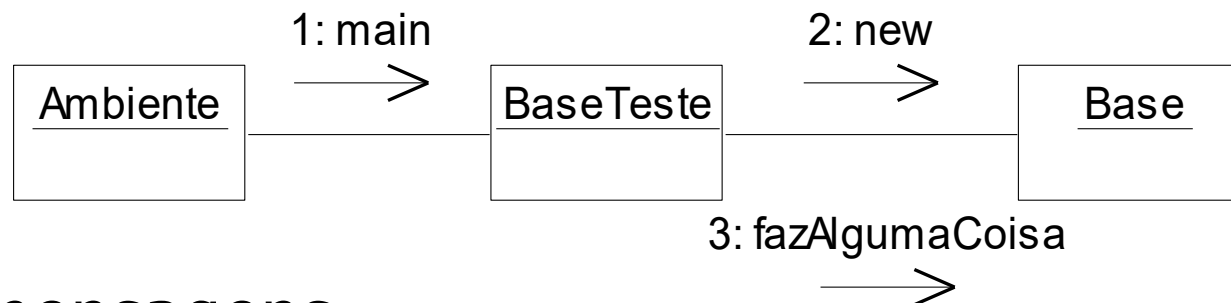
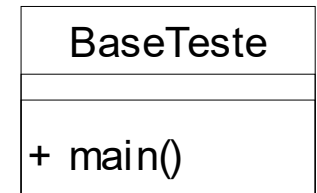
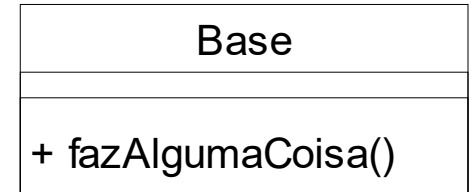
Cenário de troca de mensagens entre vários objetos
(diagrama de colaboração da UML)

Aplicação OO em Java (exemplo)

```
public class Base {  
    public void fazAlgumaCoisa() {  
        System.out.println("fiz!");  
    }  
}
```

```
public class BaseTeste {  
    public static void main(String[] args) {  
        Base b = new Base();  
        b.fazAlgumaCoisa();  
    }  
}
```

Classes



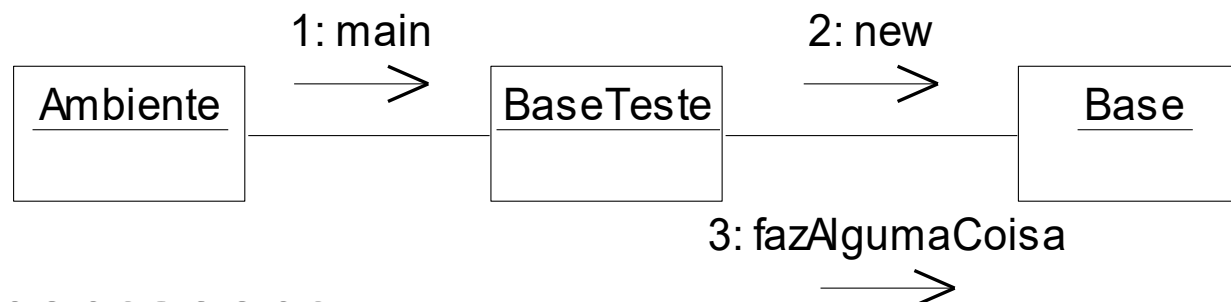
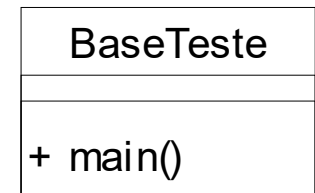
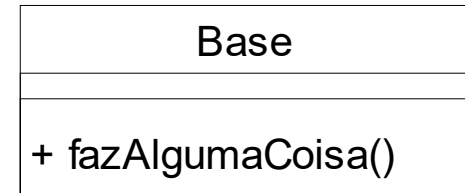
Troca de mensagens

Aplicação OO em VB.NET (exemplo)

```
Public Class Base
    Public Sub fazAlgumaCoisa()
        Console.WriteLine("fiz!")
    End Sub
End Class
```

```
Public Class BaseTeste
    Public Sub main()
        base As Base = New Base();
        base.fazAlgumaCoisa();
    }
}
```

Classes



Troca de mensagens

Resumo

- Objetos são instâncias de classes
- Objetos no mundo real são representados em software por instâncias de classes
- Classe inclui dados e comportamentos
- Objetos possuem seus próprios dados
- Objetos compartilham comportamento da classe
- Comportamento é descrito via métodos
- Chamar um método é enviar uma mensagem
- Aplicação OO é um conjunto de objetos que trocam mensagens entre eles