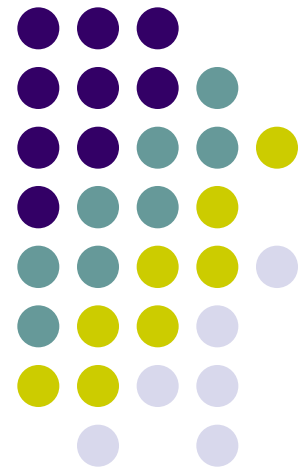


# COM220

## Aula 1: Introdução à orientação a objeto

---

Prof. Laércio Baldochi

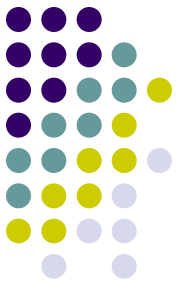




# Objetivos

- Entender os princípios básicos da orientação a objetos
- Conhecer os principais conceitos e termos da orientação à objeto
  - Entender como esses conceitos são modelados na UML
- Compreender as vantagens do uso da orientação a objeto nos projetos de software
- Entender os princípios básicos da modelagem UML

# Princípios básicos da orientação a objetos



**Orientação a Objeto**

**Abstração**

**Encapsulamento**

**Modularidade**

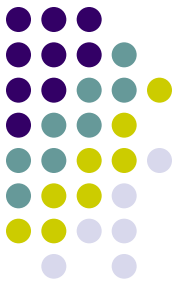
**Hierarquia**



# O que é abstração?

- Difícil conceituar...
  - É a capacidade de enxergar objetos, pessoas ou situações a partir de uma determinada perspectiva
  - Em computação
    - Concreto
      - 0 e 1
    - Tudo mais são abstrações

# O que é abstração?



Vendedor

Sem dizer  
QUAL  
vendedor –  
apenas o  
conceito de  
vendedor



Cliente

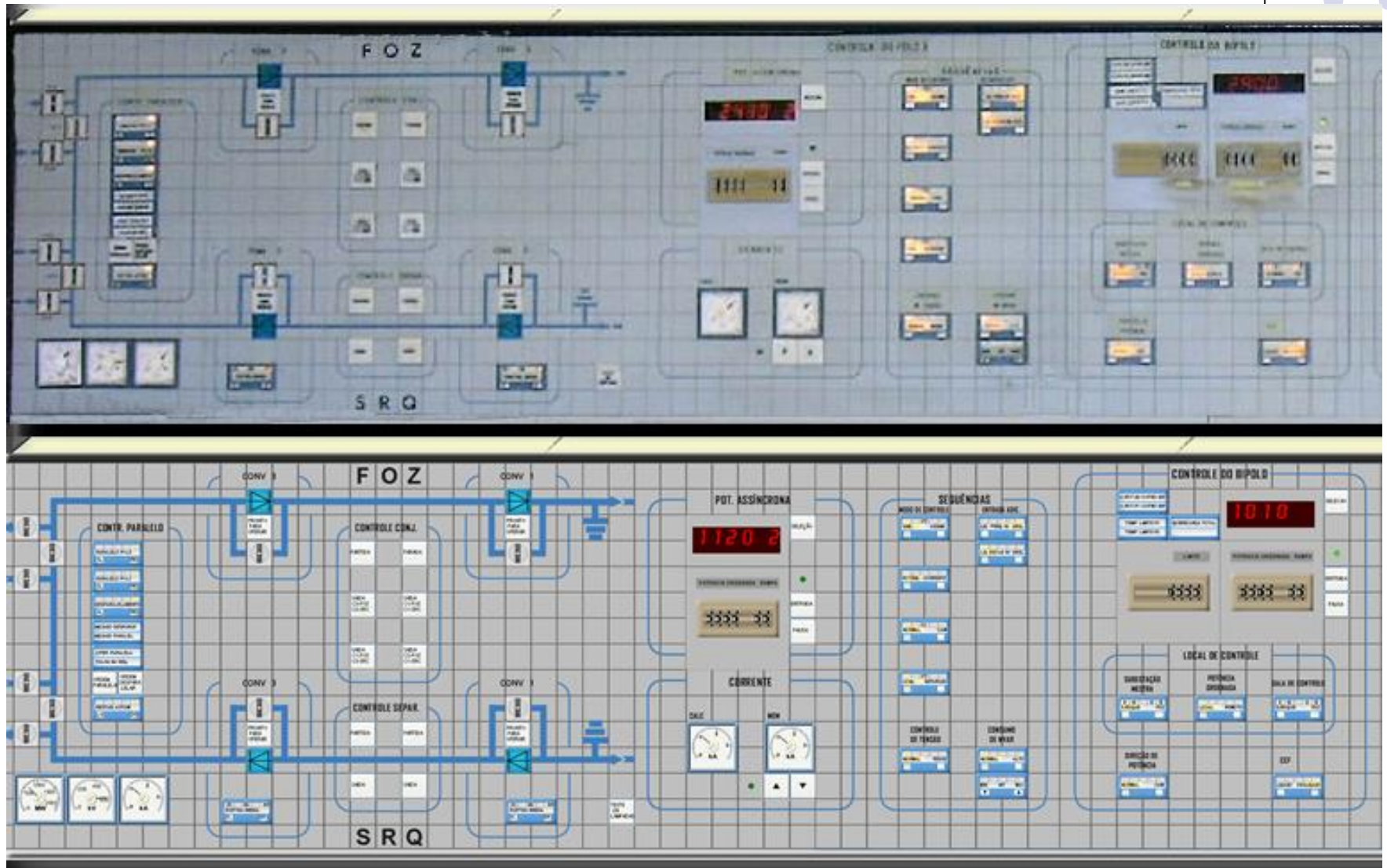
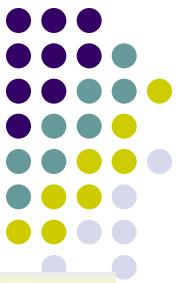


Produto

*Manipula a complexidade*

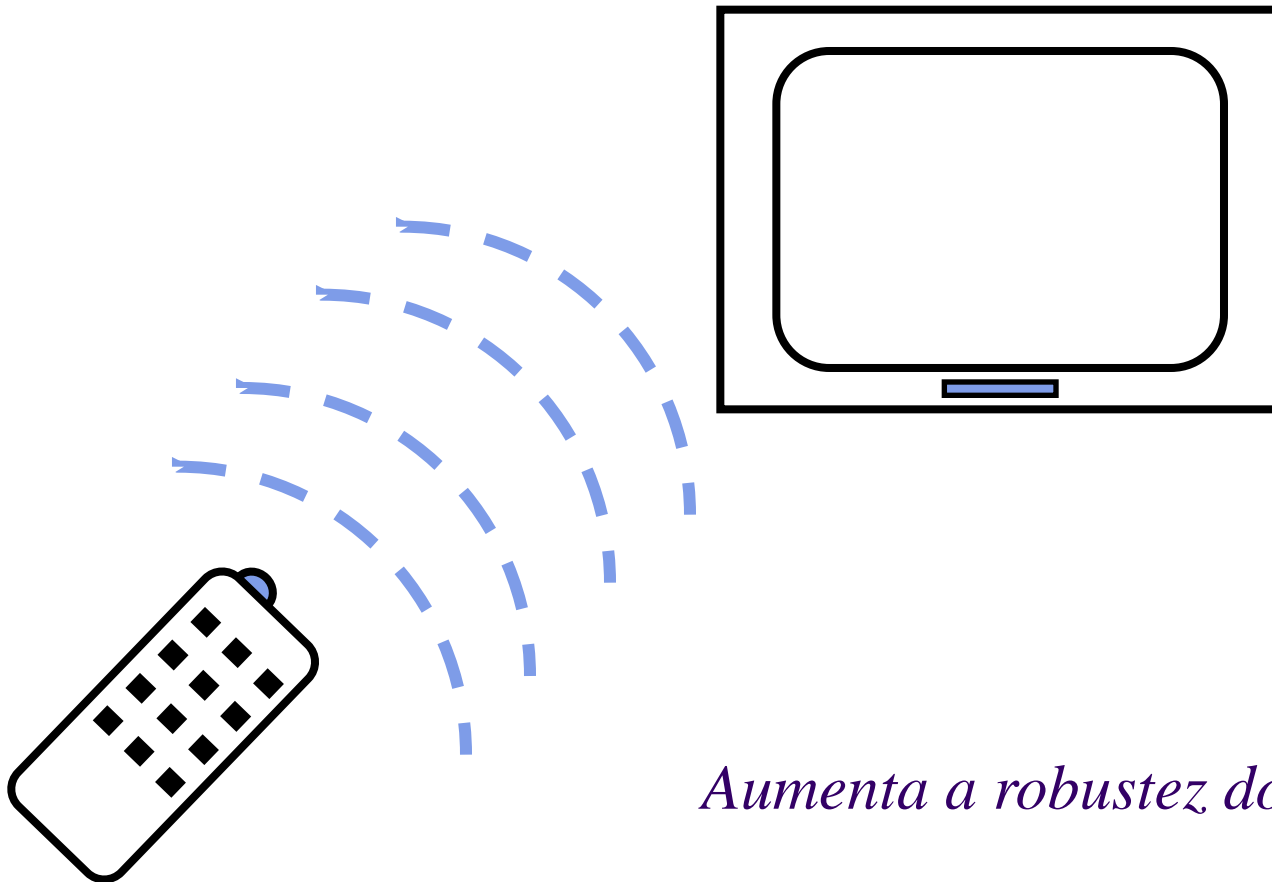
# Abstração

## Exemplo

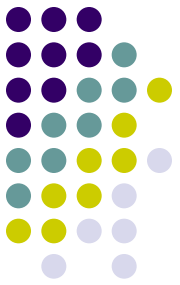


# O que é encapsulamento?

- Esconde detalhes de implementação
  - Usuário deve conhecer apenas as interfaces



*Aumenta a robustez do software*

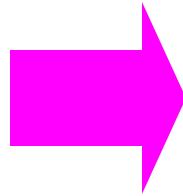




# O que é modularidade?

- Dividir algo complexo em partes mais simples e facilmente "implementáveis"

Sistema de  
processamento de  
pedidos



Preenchimento  
do pedido

Execução  
do pedido

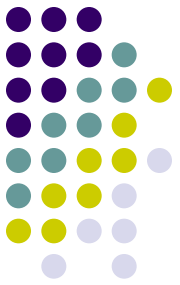
Cobrança

*Manipula a complexidade*



# O que é hierarquia?

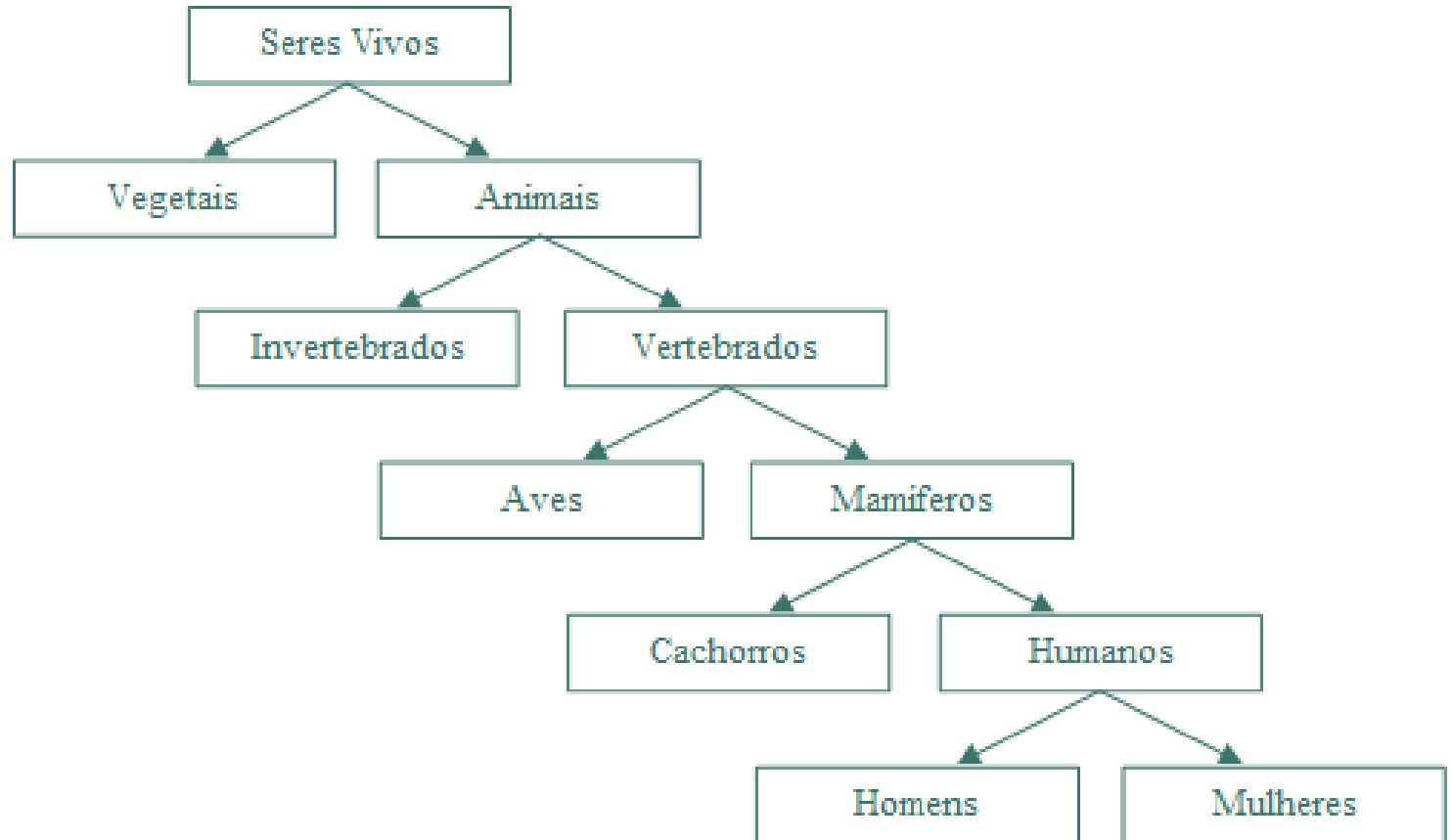
- Níveis de abstração



Maior  
abstração



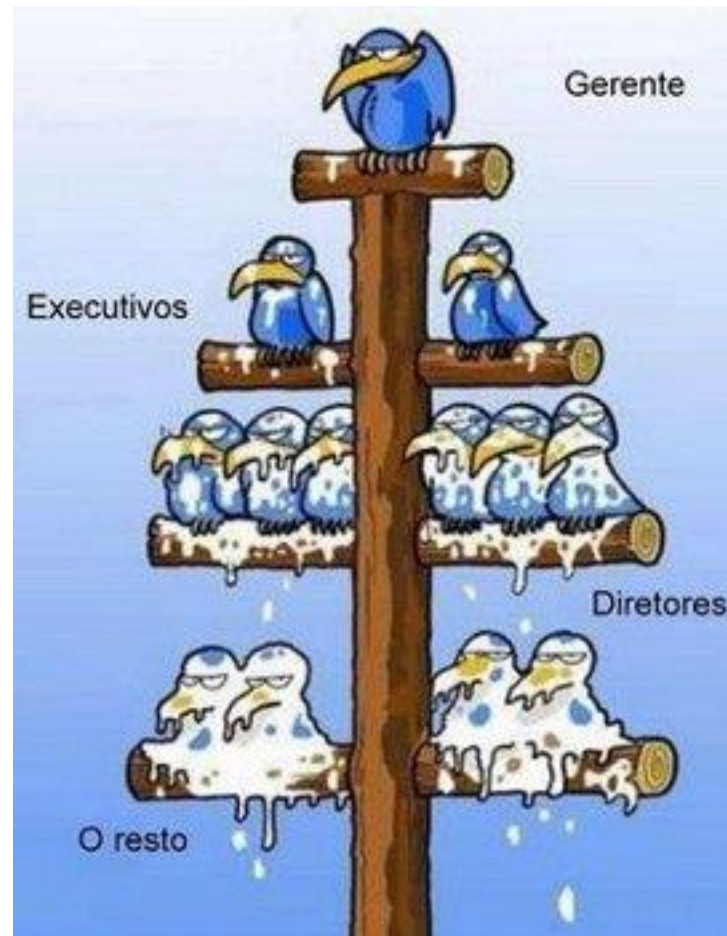
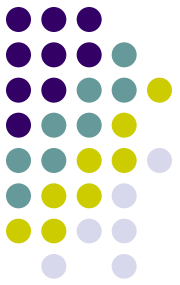
Menor  
abstração



*Elementos no mesmo nível de hierarquia  
devem estar no mesmo nível de abstração*

# Hierarquia

- Outro exemplo



# Conceitos básicos de orientação a objetos



- Objeto
- Classe
- Atributo
- Operação
- Interface (polimorfismo)
- Componente
- Pacote
- Subsistema
- Relacionamento

# Conceitos básicos de orientação a objetos

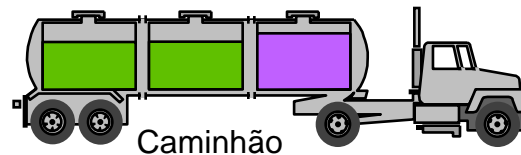


- Objeto
- Classe
- Atributo
- Operação
- Interface (polimorfismo)
- Componente
- Pacote
- Subsistema
- Relacionamento



# O que é um objeto?

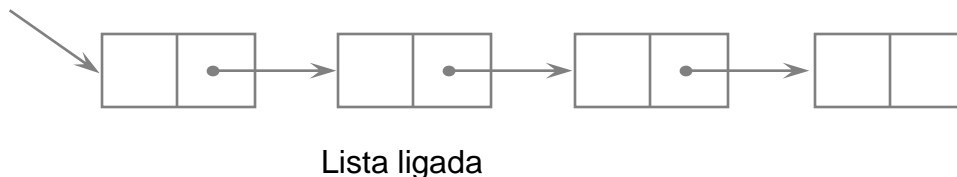
- Informalmente, um objeto representa uma entidade: física, conceitual ou de software
  - Entidade física



- Entidade conceitual



- Entidade de software



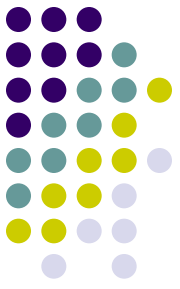


# Descrição formal

- Um objeto é um conceito, abstração ou "coisa" com características e significado no domínio de uma aplicação
- Um objeto é algo com
  - Um estado
  - Um comportamento
  - Uma identidade

# Representação de objetos

- Um objeto é representado por retângulos com nomes grifados



: Professor

Somente nome da classe

ProfessorAna

Somente nome do objeto

ProfessorAna :  
Professor

Nome da classe e do  
objeto



Professor Ana

# Conceitos básicos de orientação a objetos



- Objeto
- Classe
- Atributo
- Operação
- Interface (polimorfismo)
- Componente
- Pacote
- Subsistema
- Relacionamento





# O que é uma classe?

- É uma descrição de um grupo de objetos que possuem características comuns em termos de
  - Propriedades (atributos)
  - Comportamento (operações)
  - Relacionamentos
- Um objeto é uma instância de uma classe
- Uma classe é uma abstração, pois:
  - Destaca características principais
  - Omite características irrelevantes
- Princípio da OO: abstração

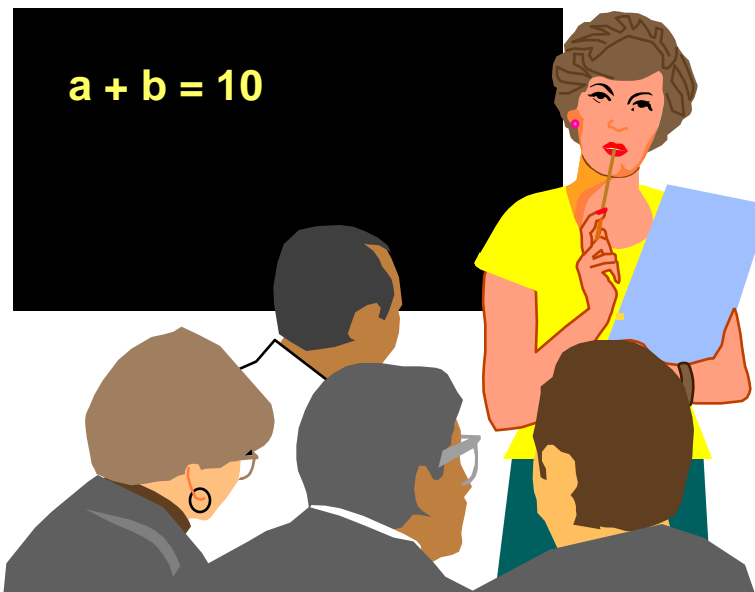
# Exemplo de classe



## Classe Disciplina

### Propriedades

Nome  
Localização  
Dias oferecim.  
Créditos  
Horário início  
Horário término



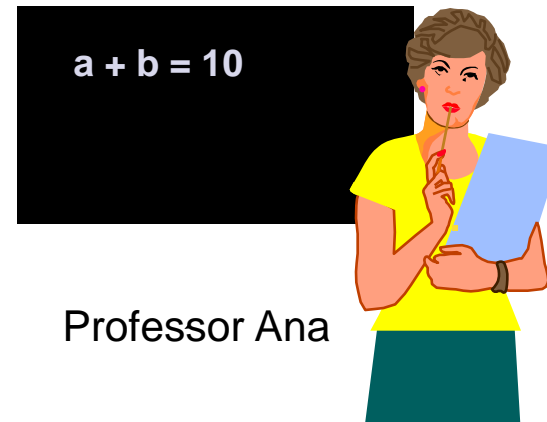
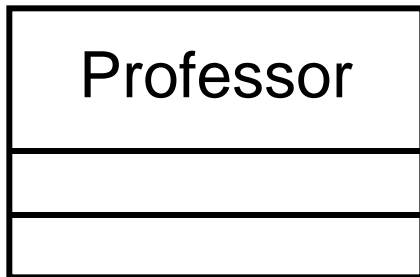
### Comportamento

Adicionar estudante  
Deletar estudante  
Gerar lista presença  
Determinar se turma  
está cheia

# Representação de classes



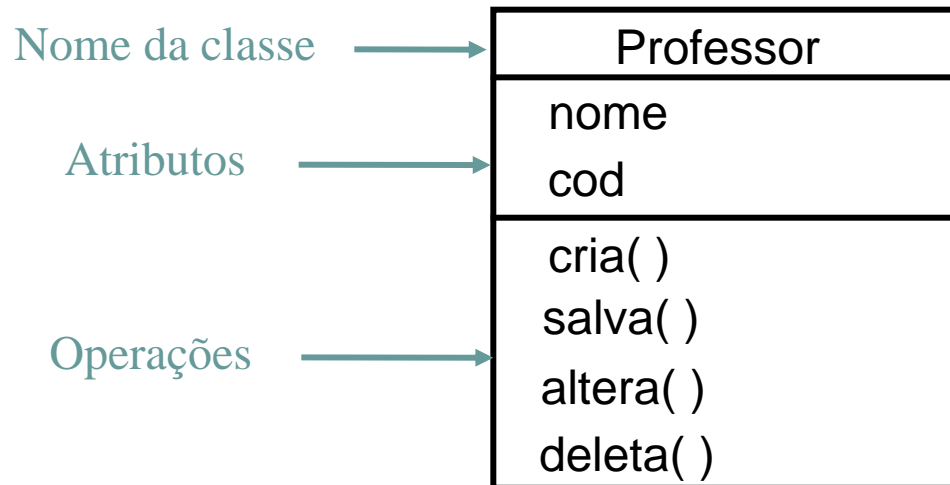
- Uma classe é representada através de um retângulo compartimentalizado





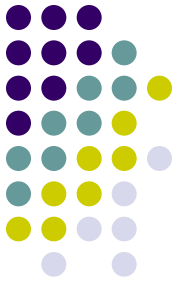
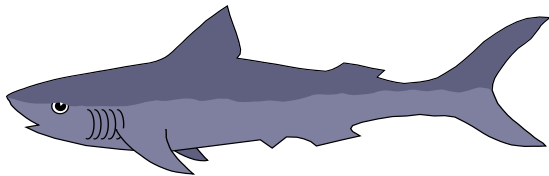
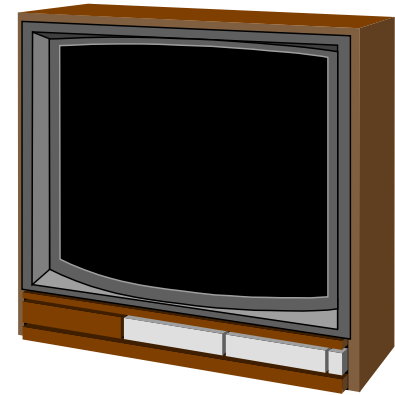
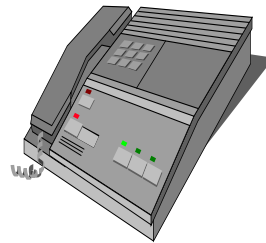
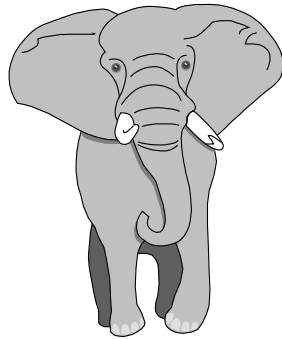
# Compartimentos de uma classe

- Uma classe é composta por três seções
  - A primeira contém o nome da classe
  - A segunda mostra sua estrutura (atributos)
  - A terceira mostra seu comportamento (operações)

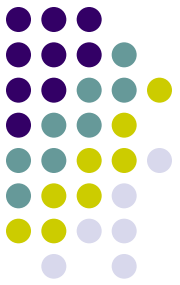


# Classes

- Quantas classes você vê aqui?



# O relacionamento entre classes e objetos



- Uma classe é uma definição abstrata de um objeto
  - Define a estrutura e o comportamento de cada objeto da classe
  - Serve como uma "forma" (template) para criar objetos
- Objetos são agrupados em classes

Objetos



Professor Ana



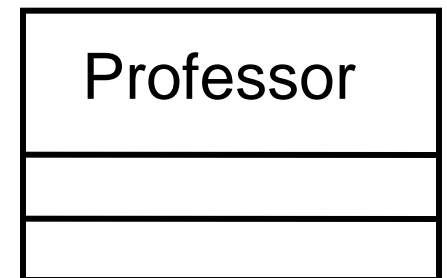
Professor Carla



Professor Maria



Classe

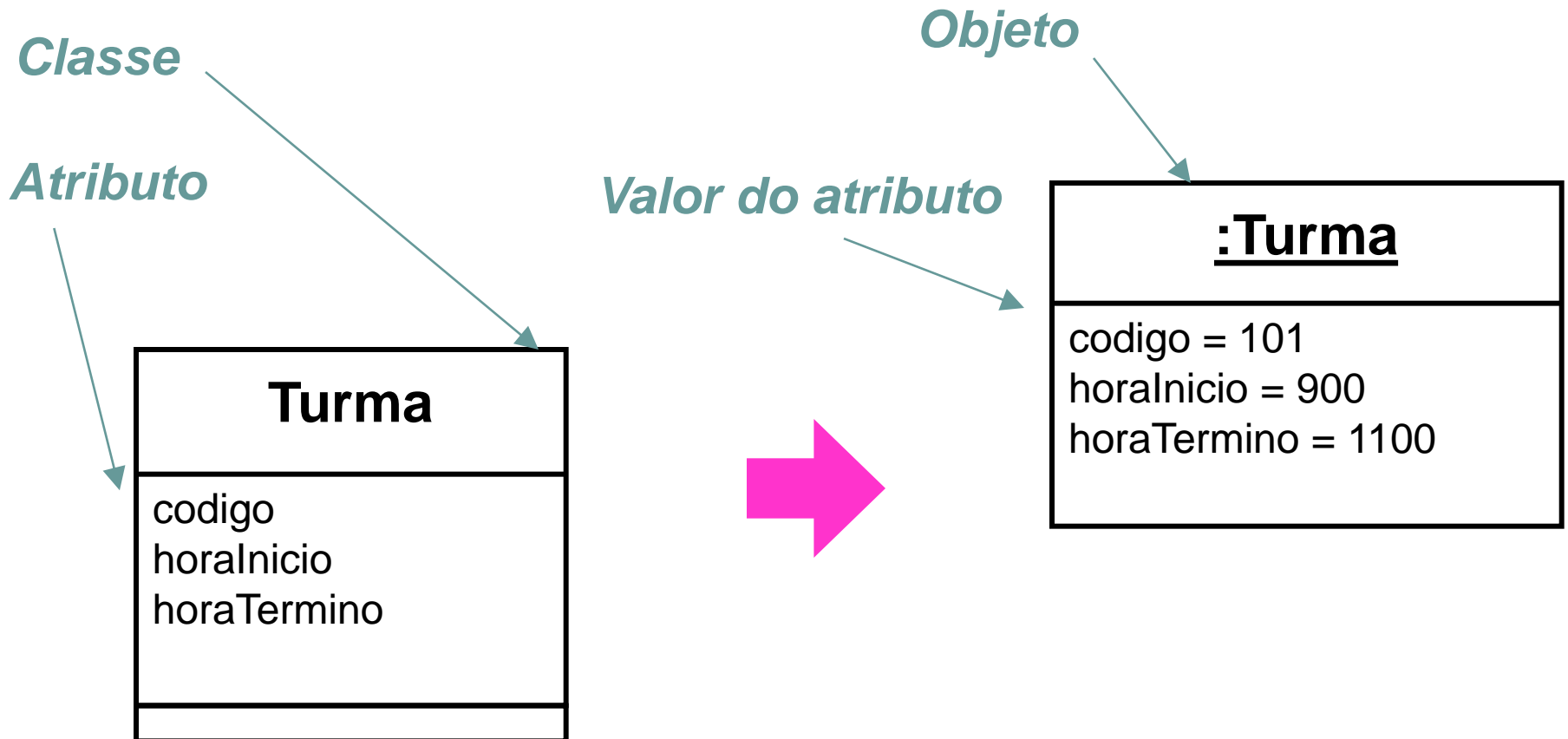


# Conceitos básicos de orientação a objetos



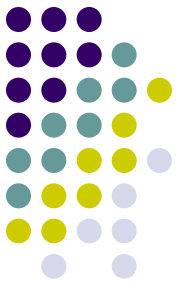
- Objeto
- Classe
- **Atributo**
- Operação
- Interface (polimorfismo)
- Componente
- Pacote
- Subsistema
- Relacionamento

# O que é um atributo?





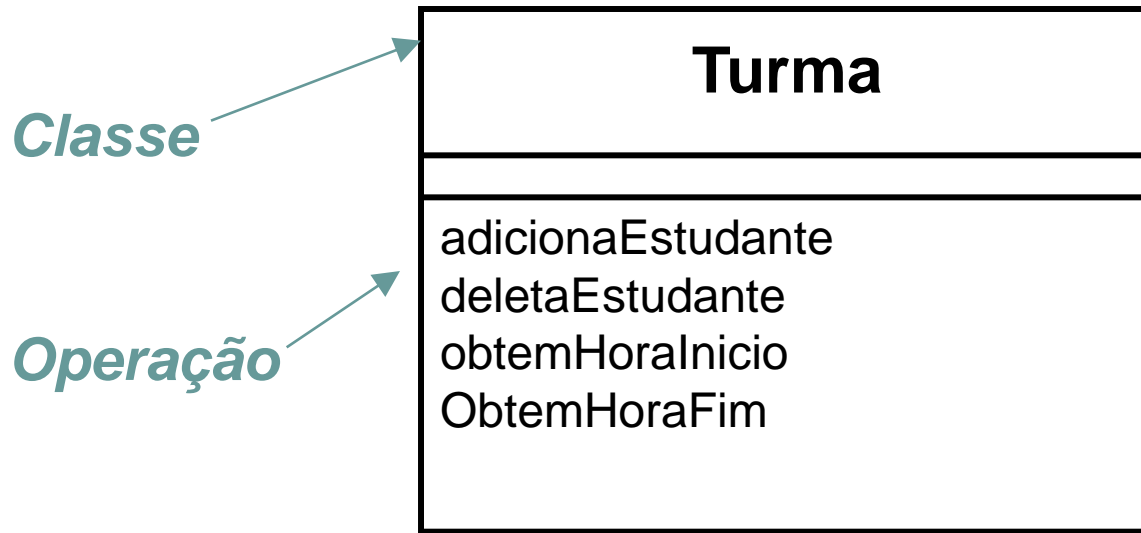
# Conceitos básicos de orientação a objetos



- Objeto
- Classe
- Atributo
- Operação
- Interface (polimorfismo)
- Componente
- Pacote
- Subsistema
- Relacionamento



# O que é uma operação?



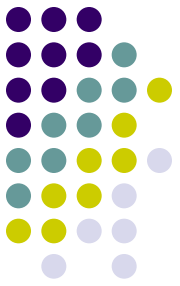
- Ações que podem ser realizadas com instâncias de uma classe

# Conceitos básicos de orientação a objetos

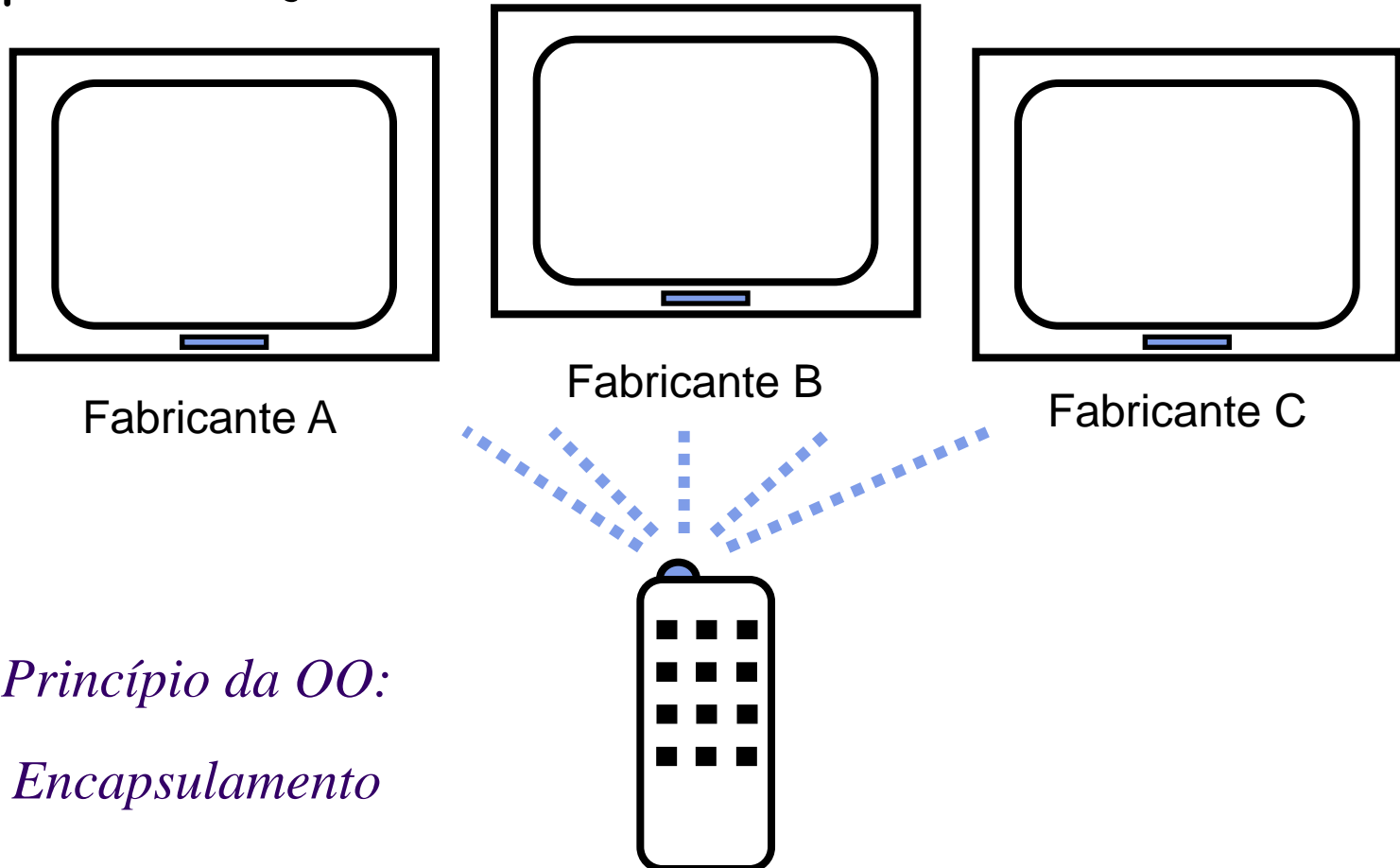


- Objeto
- Classe
- Atributo
- Operação
- Interface (polimorfismo)
- Componente
- Pacote
- Subsistema
- Relacionamento

# O que é polimorfismo?



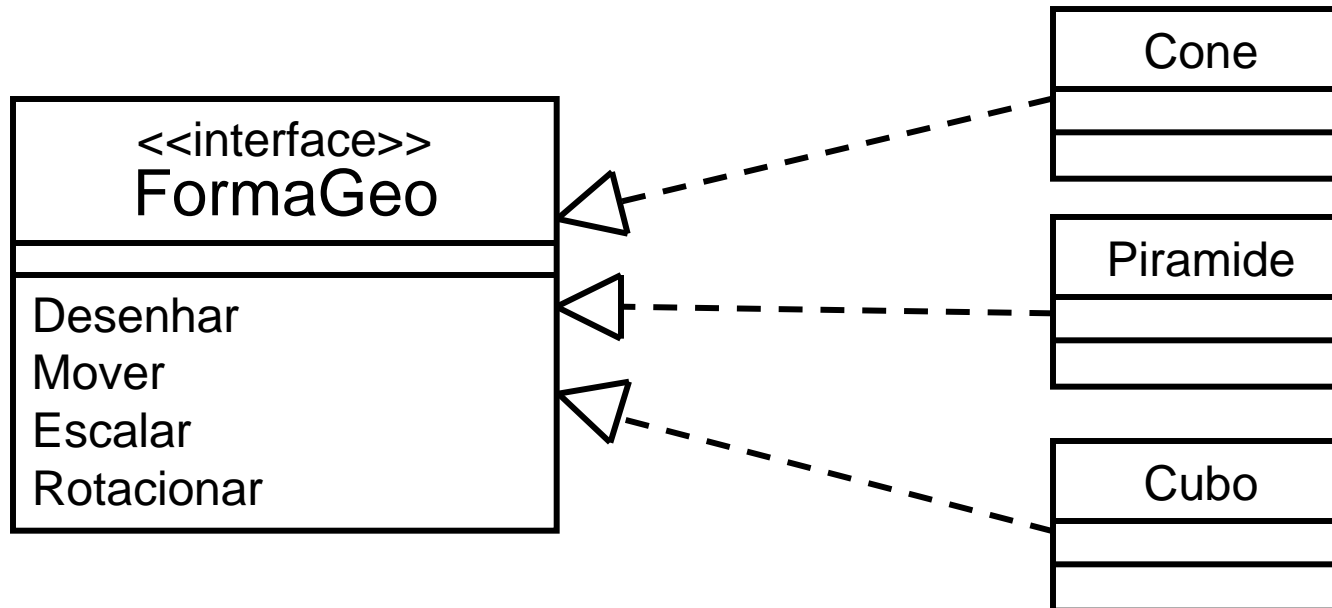
- Habilidade de "esconder" diferentes implementações utilizando uma interface única





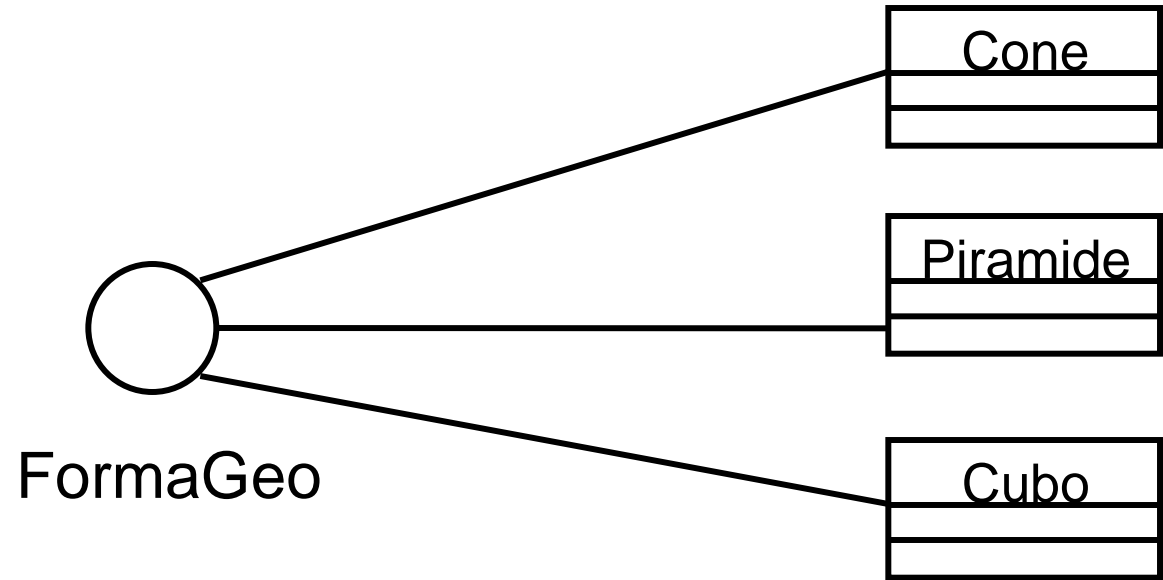
# O que é uma interface?

- Interfaces formalizam o polimorfismo
  - Suportam arquiteturas "plug-and-play"

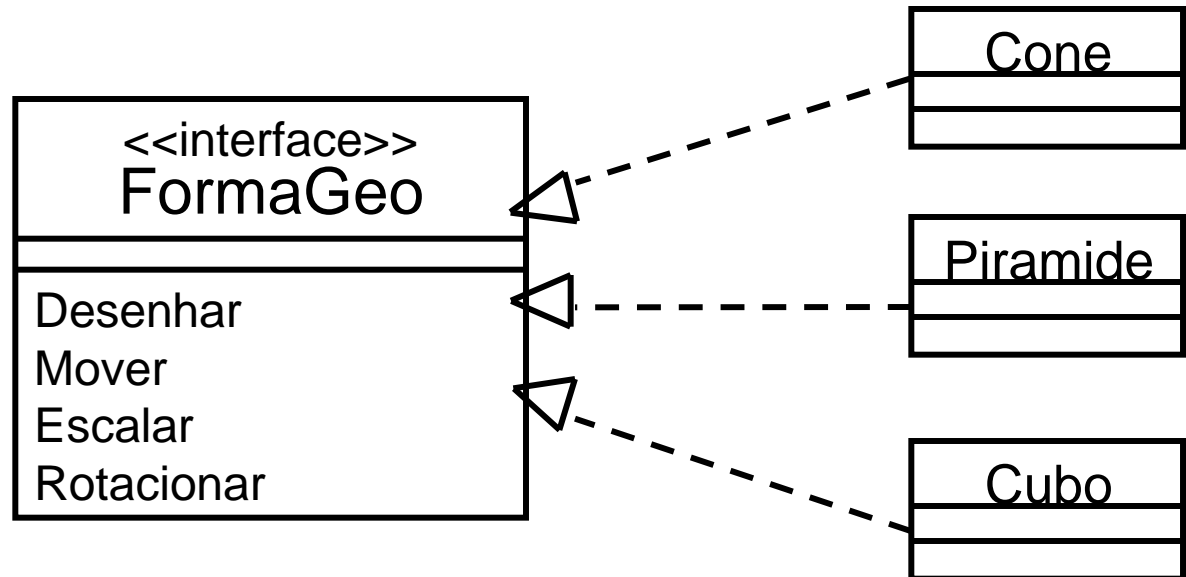


# Possíveis representações

Representação  
icônica  
("lollipop")



Representação  
canônica



# Conceitos básicos de orientação a objetos

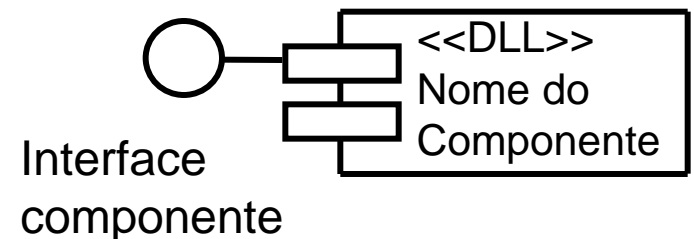
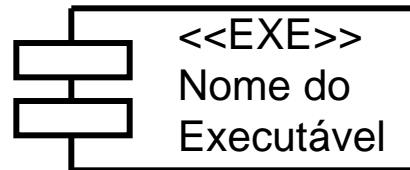
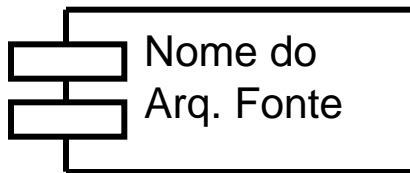


- Objeto
- Classe
- Atributo
- Operação
- Interface (polimorfismo)
- **Componente**
- Pacote
- Subsistema
- Relacionamento



# O que é um componente?

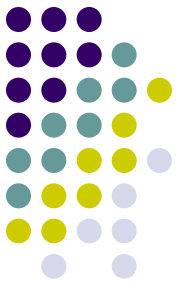
- Parte não trivial e substituível de um sistema que realiza uma função específica no contexto de uma arquitetura bem definida de software
- Um componente pode ser
  - Um trecho de código fonte componentizável
  - Um componente de run time
  - Um componente executável



Princípio da OO:  
Encapsulamento



# Conceitos básicos de orientação a objetos

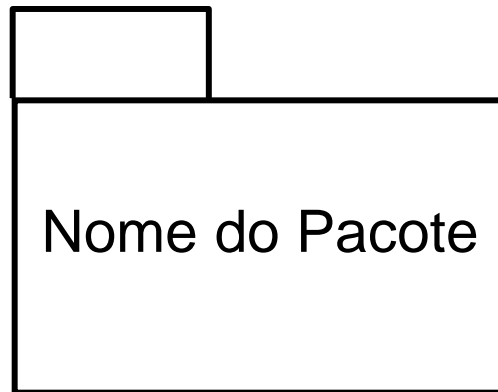


- Objeto
- Classe
- Atributo
- Operação
- Interface (polimorfismo)
- Componente
- Pacote
- Subsistema
- Relacionamento



# O que é um pacote?

- É um mecanismo que permite agrupar elementos
- É um elemento de modelagem que pode conter outros elementos de modelagem



*Princípio da OO:  
Modularidade*

- Uso principal
  - Organizar o modelo em desenvolvimento

# Conceitos básicos de orientação a objetos



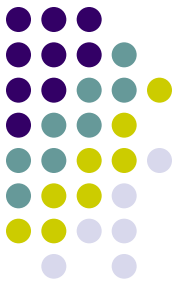
- Objeto
- Classe
- Atributo
- Operação
- Interface (polimorfismo)
- Componente
- Pacote
- Subsistema
- Relacionamento



# O que é um subsistema?

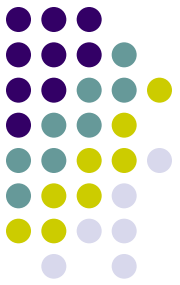
- Parte de um sistema
  - Utilizado para decompor um sistema complexo em partes quase independentes
- Permitem visualizar um sistema de um ponto de vista mais panorâmico

# Exercícios



1. Qual a diferença entre objeto e classe?
2. Discuta como os 4 princípios básicos da OO encontram-se embutidos nos conceitos básicos da OO?
3. O que é polimorfismo? Dê um exemplo de polimorfismo (diferente do que foi dado em aula, é claro!)
4. Modele a classe Aluno. Procure dar a essa classe as características (atributos) e o comportamento (operações) adequadas a um aluno universitário.
  - Use a representação compartimentalizada

# Conceitos básicos de orientação a objetos



- Objeto
- Classe
- Atributo
- Operação
- Interface (polimorfismo)
- Componente
- Pacote
- Subsistema
- Relacionamento [ver aula 2]