

• • • • • • • • • •

Construção de Aplicações Java



Fátima Pires
fatima@ccuec.unicamp.br
CCUEC/Unicamp

• • • • • • • • • •

•
•
•

Objetivo

- Introduzir os **conceitos** de Orientação a Objetos
- Familiarizar-se com a **nomenclatura e notações** da nova tecnologia
- Motivar a mudança do **paradigma** de concepção de sistemas
- Entender os conceitos aplicados em **Java**

• • • • • • • • • •

Cenário

Cenário

Vivendo uma Revolução na Indústria de Software

- ◆ Complexidade
- ◆ Confiabilidade
- ◆ Capacidade de Projeto
- ◆ Flexibilidade
- ◆ Velocidade de desenvolvimento
- ◆ Facilidade de manutenção

- # Orientação a Objetos

Alavancas

-

• • •

Roteiro

Conceitos

- ❖ **Java**
- ❖ **Objetos, atributos, métodos, mensagens, encapsulamento**
- ❖ **Classes e Relacionamentos**
- ❖ **Herança**
- ❖ **Atributos e Métodos de Classe**
- ❖ **Tipos de Herança**
- ❖ **Composição**
- ❖ **Polimorfismo**

•
•
•
•
•
•
•

•
•
•

Roteiro

Reutilização de Projeto e Software

- ❖ Padrões de Projeto (*Design Patterns*)
- ❖ *Frameworks*
- ❖ Herança e Composição
- ❖ Componentes
- ❖ RMI (*Remote Method Invocation*)

• • • • • • • • • •

● ● ●

Java e a sua história

- ✓ Java (ex-Oak)
- ✓ James Gosling – 1991 – Sun Microsystems
- ✓ Uso em equipamentos eletrônicos
- ✓ Java e a Web

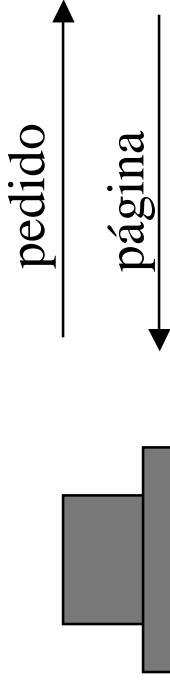
[illegible]

•
•
•

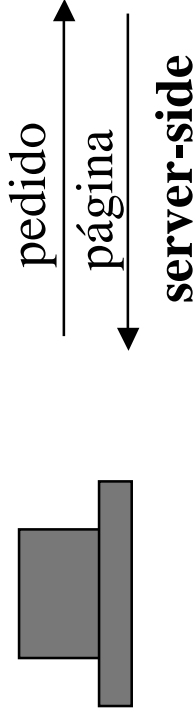
Internet - Sistema Cliente/Servidor

Web browser (cliente)

Web server

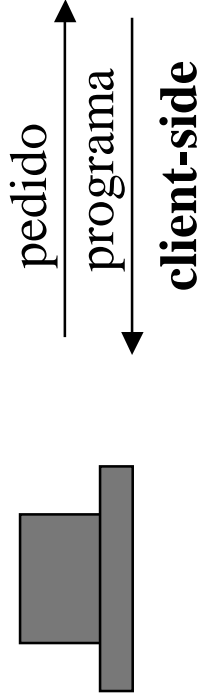


Comportamento estático



Comportamento dinâmico

- Menor desempenho
- Milhares de clientes



Comportamento dinâmico

- Problemas com segurança
- Problemas com diferentes ambientes

• • • • • • • • • • •

• • •

Java

Java é ...

- ✓ **Simples**
- ✓ **Orientada a Objetos**
- ✓ **Orientada para a Rede**
- ✓ **Interpretada**
- ✓ **Robusta**
- ✓ **Segura**
- ✓ **Independente de plataforma**
- ✓ **Portável**
- ✓ **Multithreaded**
- ✓ **Dinâmica**

[illegible]

•
•
•

Orientação a Objetos

Origens

- ❖ Linguagens de Programação - Simula, Smalltalk, Flavours, Objective C, C++,...
- ❖ Inteligência Artificial - frames
- ❖ Banco de Dados - pesquisa em modelos de dados semânticos

• • • • • • • • • •

•
•
•

Paradigma

“Paradigma é um conjunto de regras que estabelecem fronteiras e descreve como resolver os problemas dentro destas fronteiras.

Os paradigmas influenciam nossa percepção; ajudam-nos a organizar e a coordenar a maneira como olhamos para o mundo...”

Reengenharia - Reestruturando a Empresa
Daniel Morris e Joel Brandon

• • • • •

•
•
•

Orientação a Objetos

O termo orientação a objetos significa organizar o mundo real como uma coleção de **objetos**.

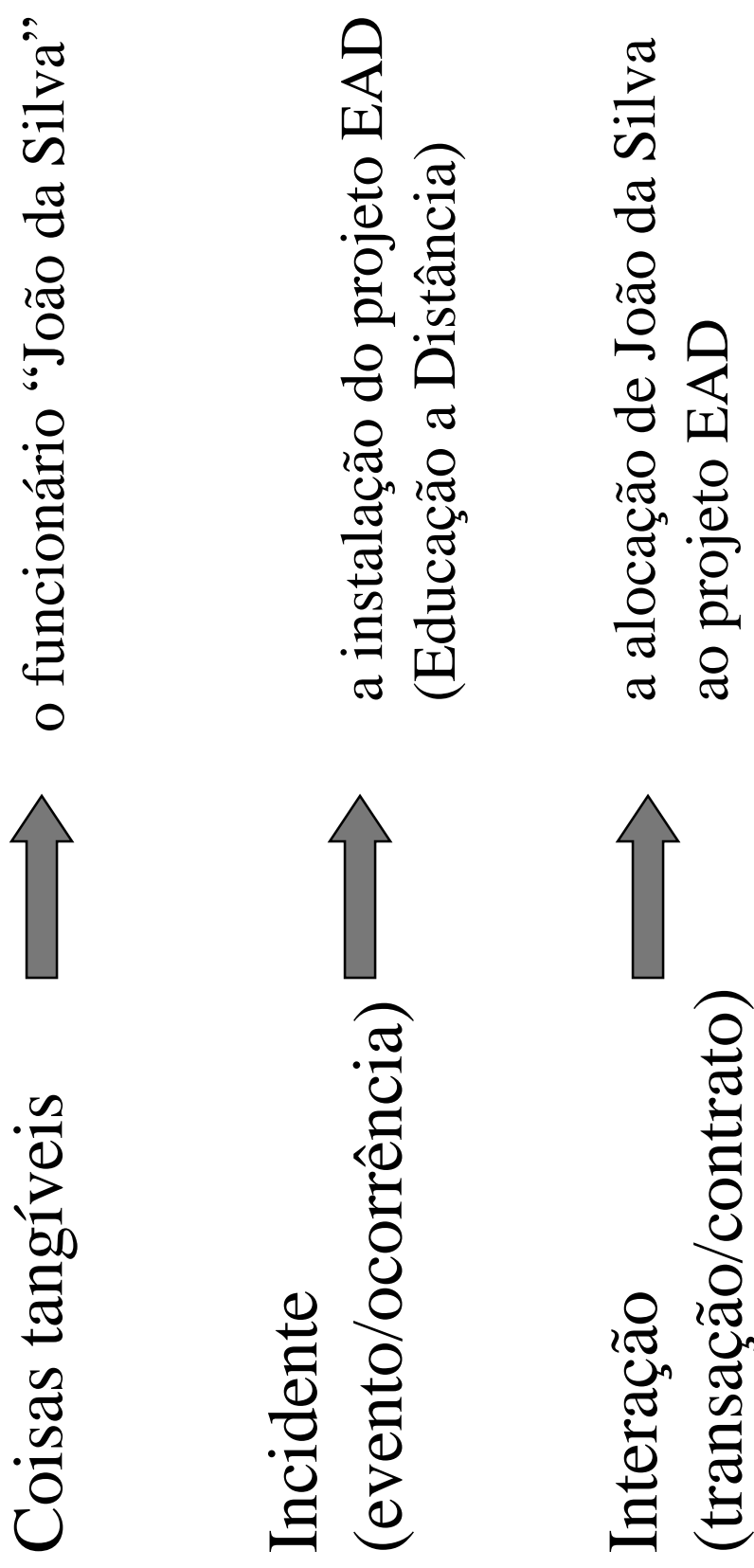
Dos objetos nos interessa a sua *estrutura de dados* e *um conjunto de operações* que manipulam estes dados.

Os objetos, a sua estrutura de dados e o conjunto de operações são selecionados de acordo com objetivos bem estabelecidos.

• • • • • • • • • •

•
•
•

Objetos - Exemplos

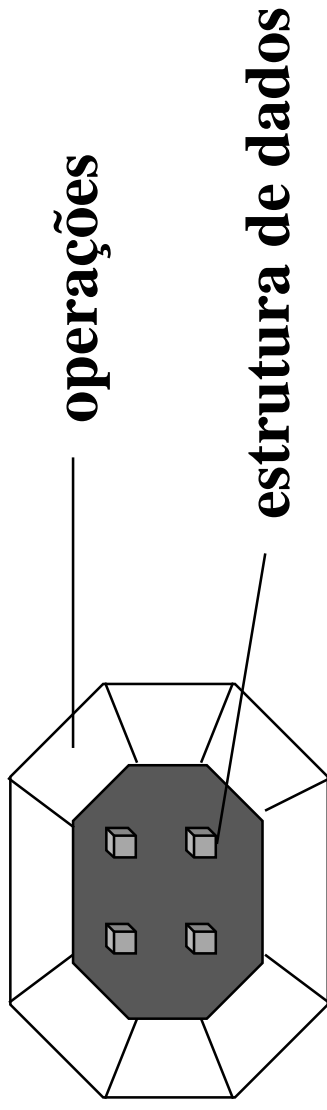


• • • • • • •

•
•
•

Objetos - Metáfora

Como visualizar objetos

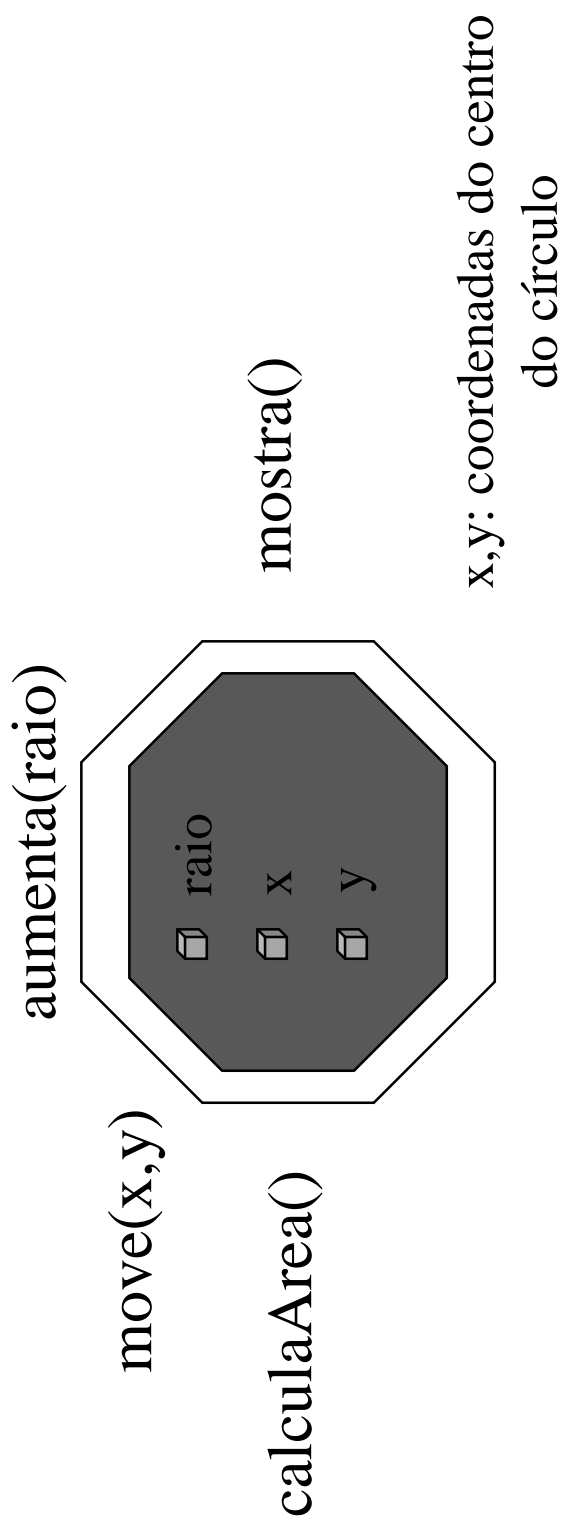


um objeto

• • • • • • • • • •

•
•
•

Objetos - Exemplo

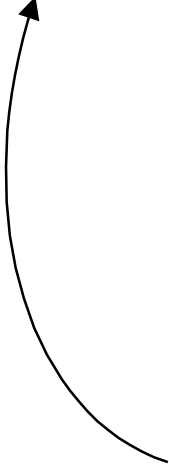
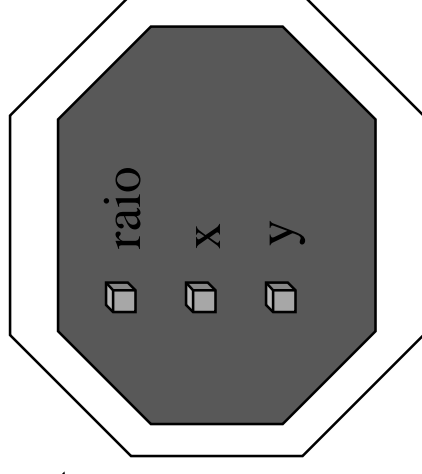


um círculo-objeto

• • • • •

•
•
•

Objetos em Java

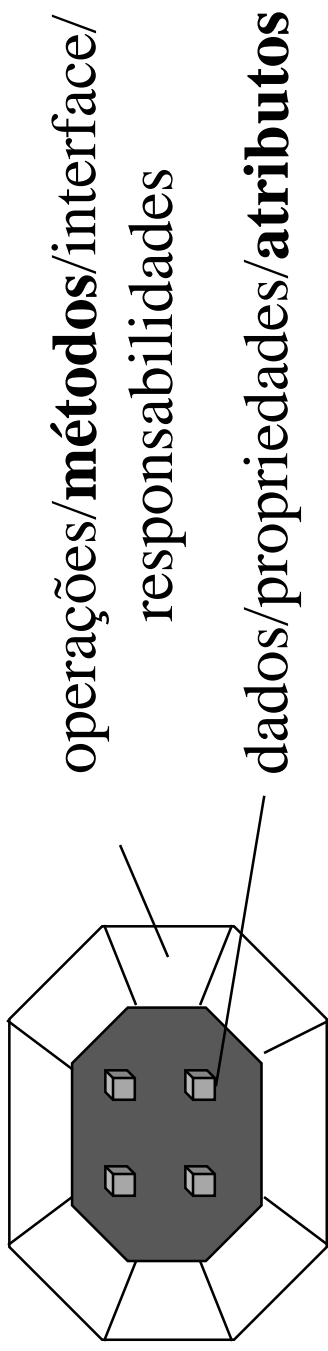


`circulo`
(variável de programa)

• • • • • • • • • •

•
•
•

Objeto - Estado e Comportamento



A estrutura de dados armazena o **estado** de um objeto (valores dos atributos)

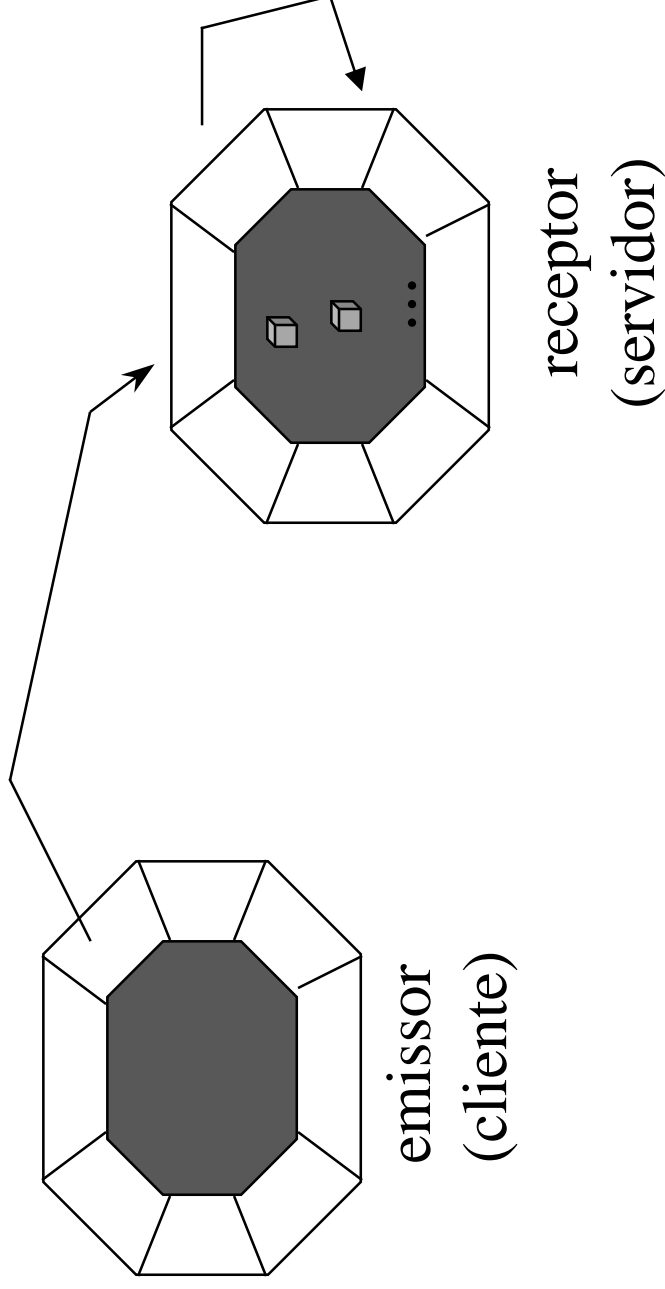
As operações definem o **comportamento** do objeto, que é a forma como um objeto age e reage em termos de mudanças de estado e respostas às operações que sabe executar.

• • • • • • • • • •

•
•
•

Mensagens

Objetos interagem e comunicam-se através de
mensagens...

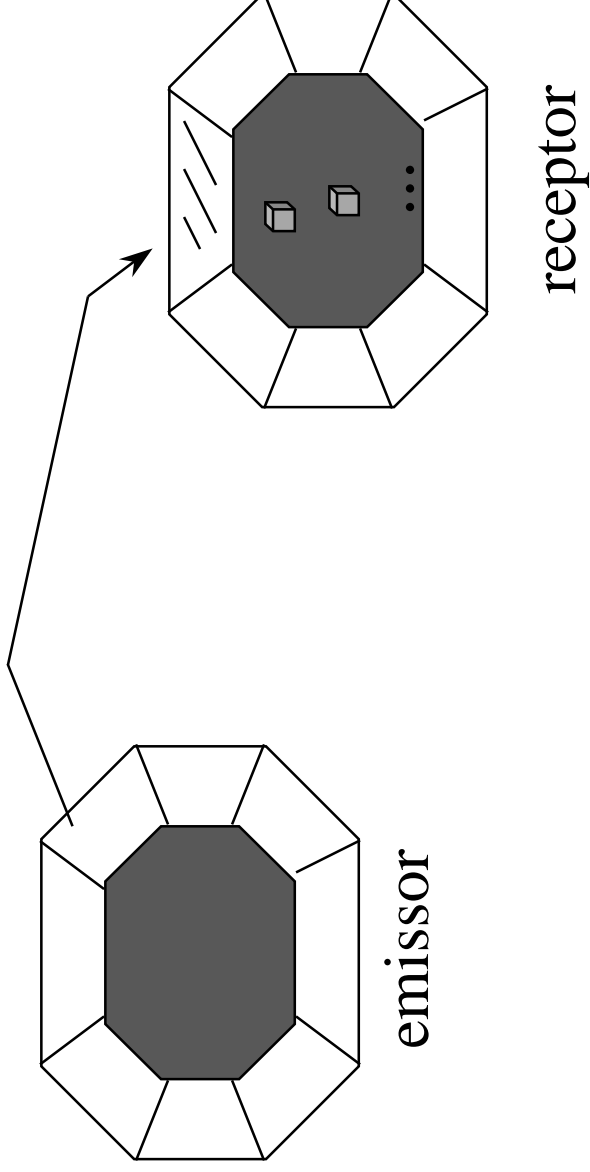


• • • • •

•
•
•

Métodos

...as mensagens identificam os *métodos* a serem executados no objeto receptor

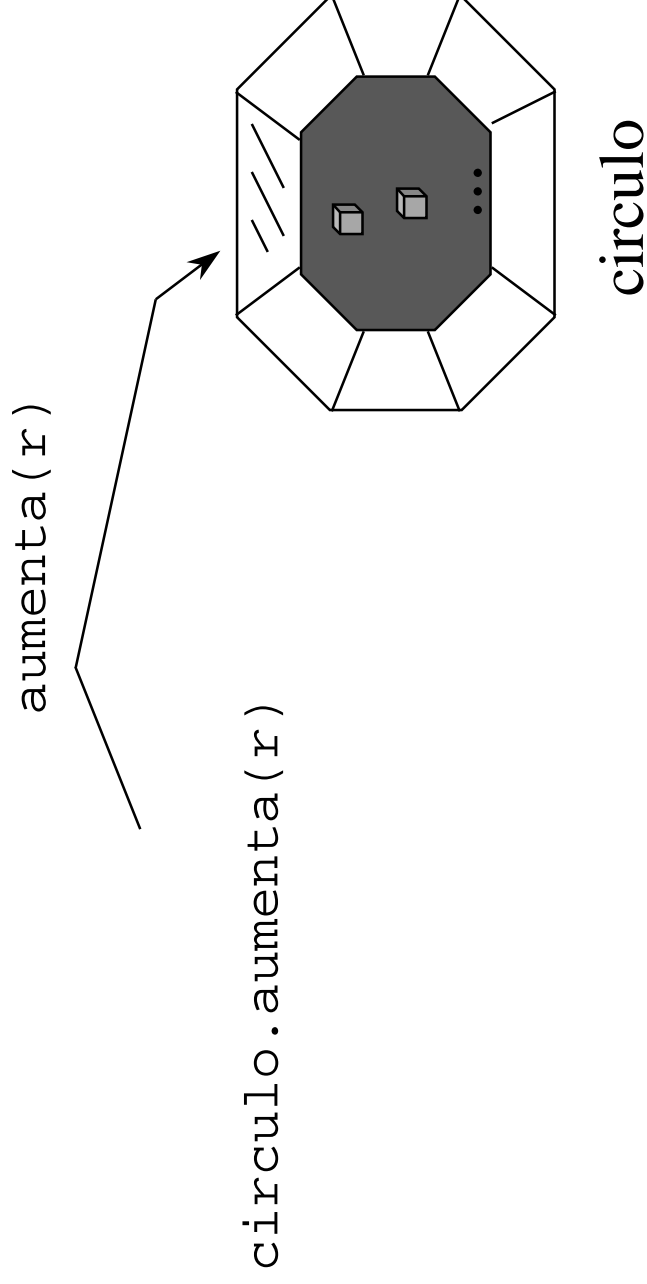


• • • • • • • • • •

•
•
•

Mensagens em Java

Enviando mensagens em Java



• • • • •

•
•
•

Mensagens e Métodos

Resumindo...

Para invocar um método de um objeto, deve-se enviar uma mensagem para este objeto.

Para enviar uma mensagem deve-se:

- identificar o *objeto* que receberá a mensagem
- identificar o *método* que o objeto deve executar
- passar os *argumentos* requeridos pelo método

• • • • • • • • • •

- ● ●

Encapsulamento

Encapsulamento é definido como uma técnica para minimizar interdependências entre “módulos” através da definição de interfaces externas.

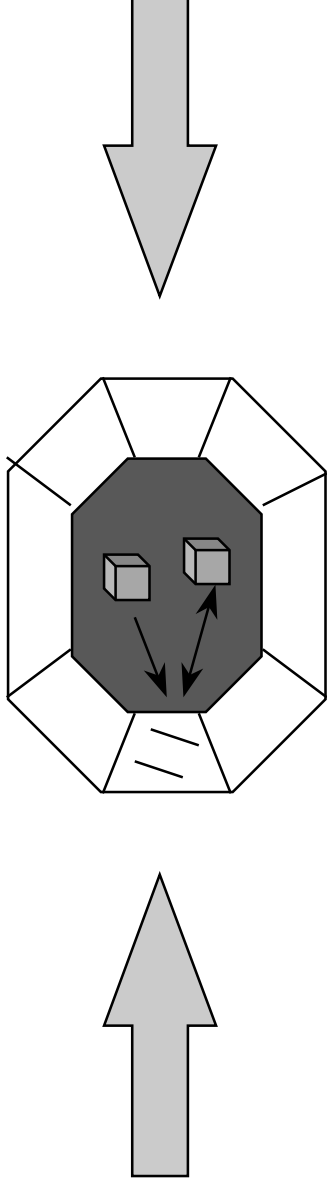
fenômeno da “caixa preta”

- [illegible]

•
•
•

Objetos - Encapsulamento

A interface de um objeto declara todas as operações permitidas



Todo o acesso aos dados do objeto é feito através da chamada a uma operação da sua interface

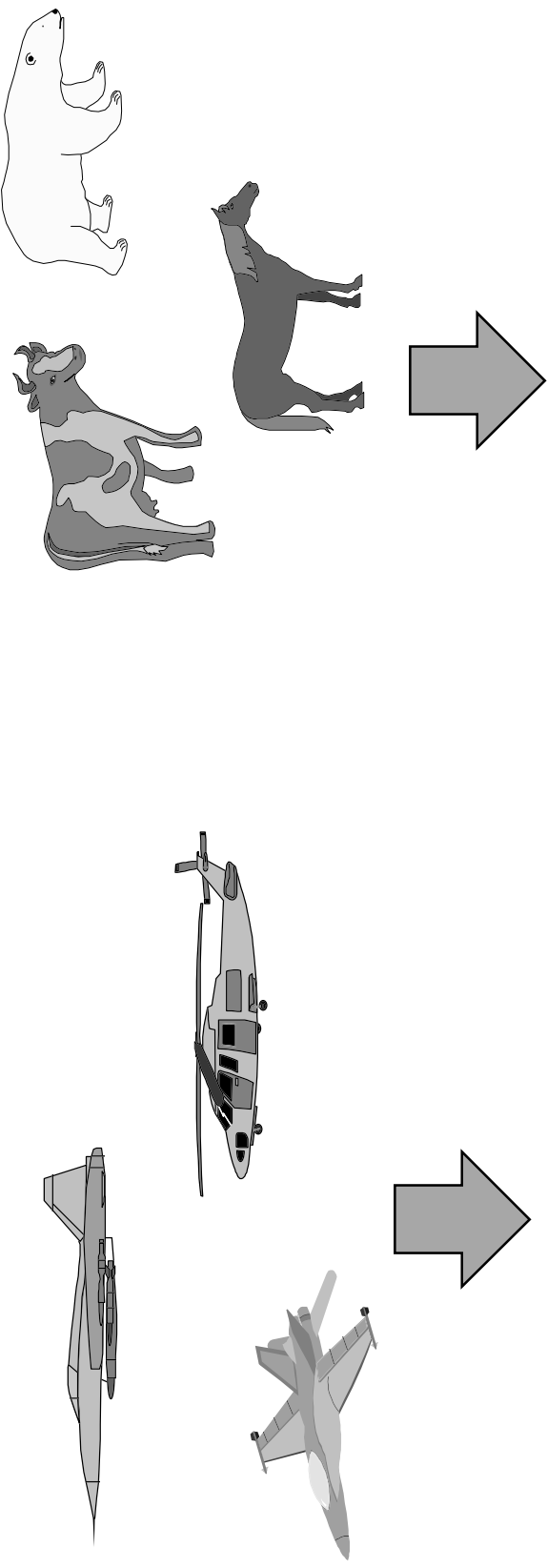
Mudanças na implementação de um objeto, que preservem a sua interface externa, não afetam o resto do sistema.

• • • • •

•
•
•

Abstração

Focalizar o essencial, ignorar propriedades acidentais



A abstração deve ser sempre feita com algum objetivo, porque este determina o que é e o que não é importante.

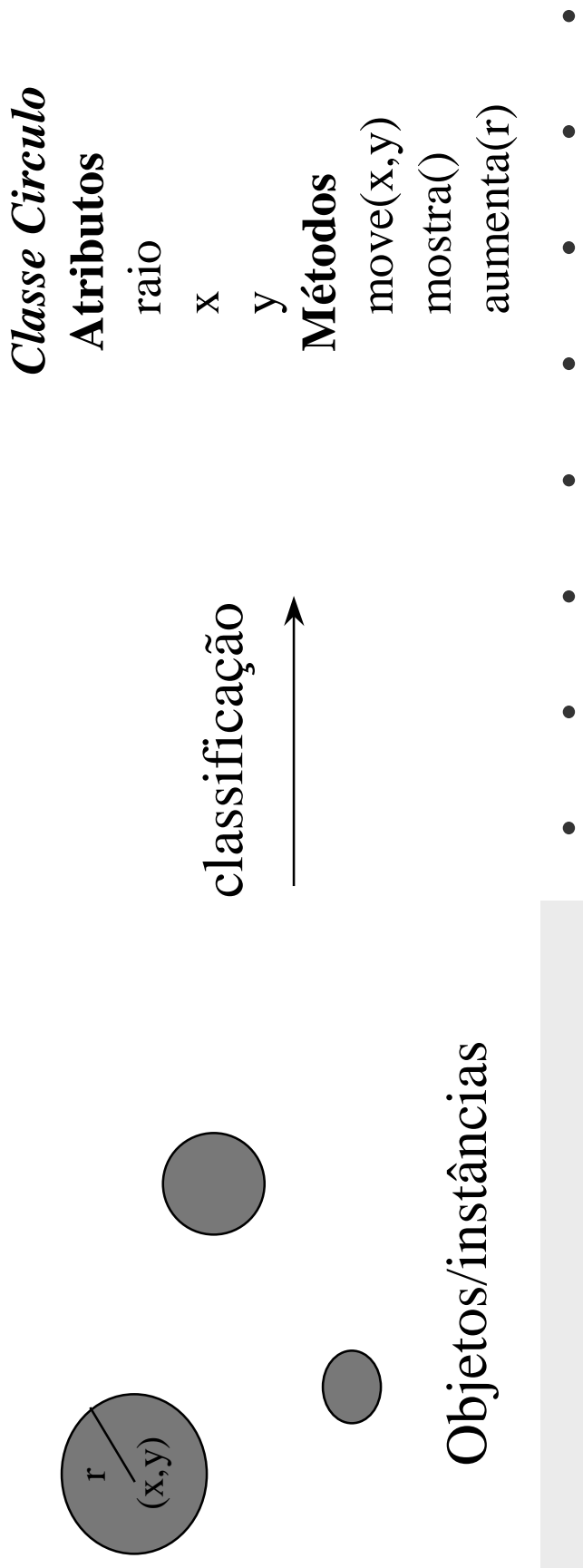
• • • • •

- # Classes

Classes

Uma classe descreve um conjunto de objetos com:

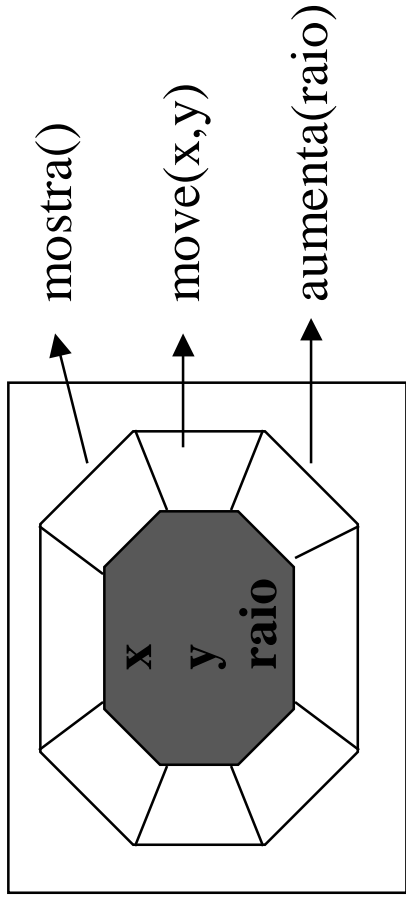
- propriedades semelhantes
- comportamentos semelhantes
- relacionamentos comuns com outros objetos



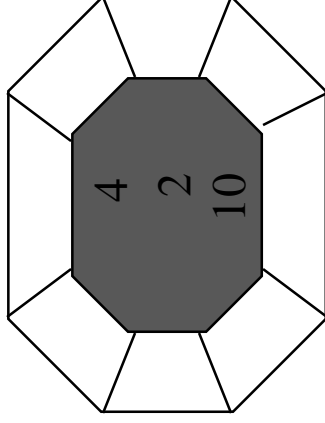
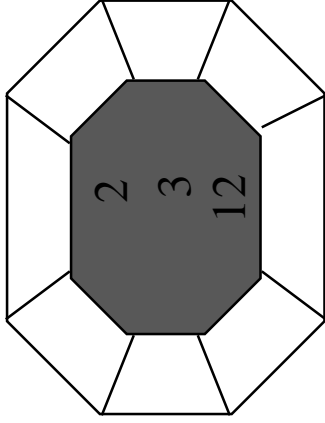
•
•
•

Classe, uma Fábrica de Objetos

Classe Circulo



instanciação de classe



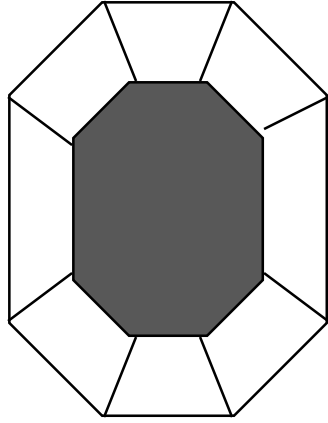
Objetos ou instâncias da Classe Circulo

• • • • •

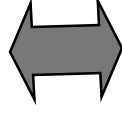
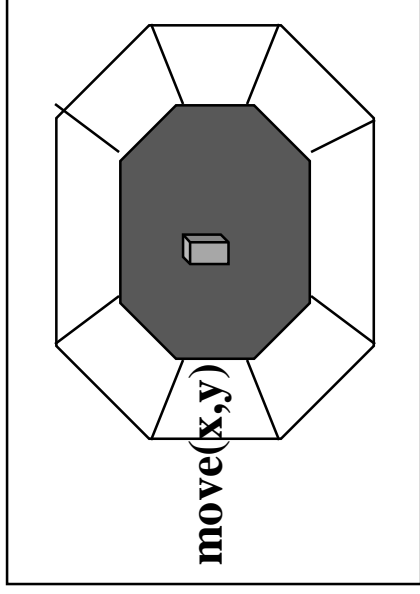
•
•
•

Comunicação entre Objetos

tela-objeto

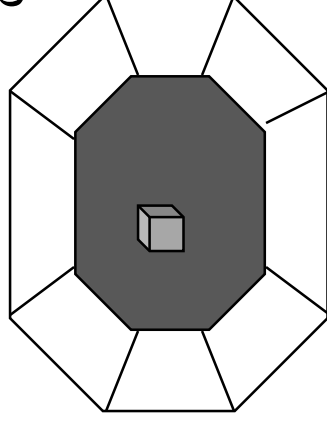


Classe Círculo



procurar o
método na definição
da classe

move (2,3)



círculo-objeto

• • • • •

•
•
•

Classes

Definição de classe em Java

```
public class Circulo {  
    public float raio, x, y;    // atributos da classe
```

```
    public Circulo (float pr, float px, float py)  
    { raio = pr;  x = px;  y = py; }
```

```
    public void move( float dx, float dy)  
    { x+= dx;  y+= dy; }
```

```
    public void mostra()  
    { System.out.println(("+x+", +y+", +raio+")); }  
}
```

• • • • •

-
-
-

Classes

Instanciando objetos em Java

```
class Principal {  
    public static void main (String args []){  
        Circulo umcirc;  
  
        umcirc = new Circulo (12,0,0);    // instânciação de Circulo  
        umcirc.mostra();  
        umcirc.move(2,3);  
        umcirc.mostra();  
    }  
}
```

-
-
-
-
-
-
-

•
•
•

Diagramas e suas notações

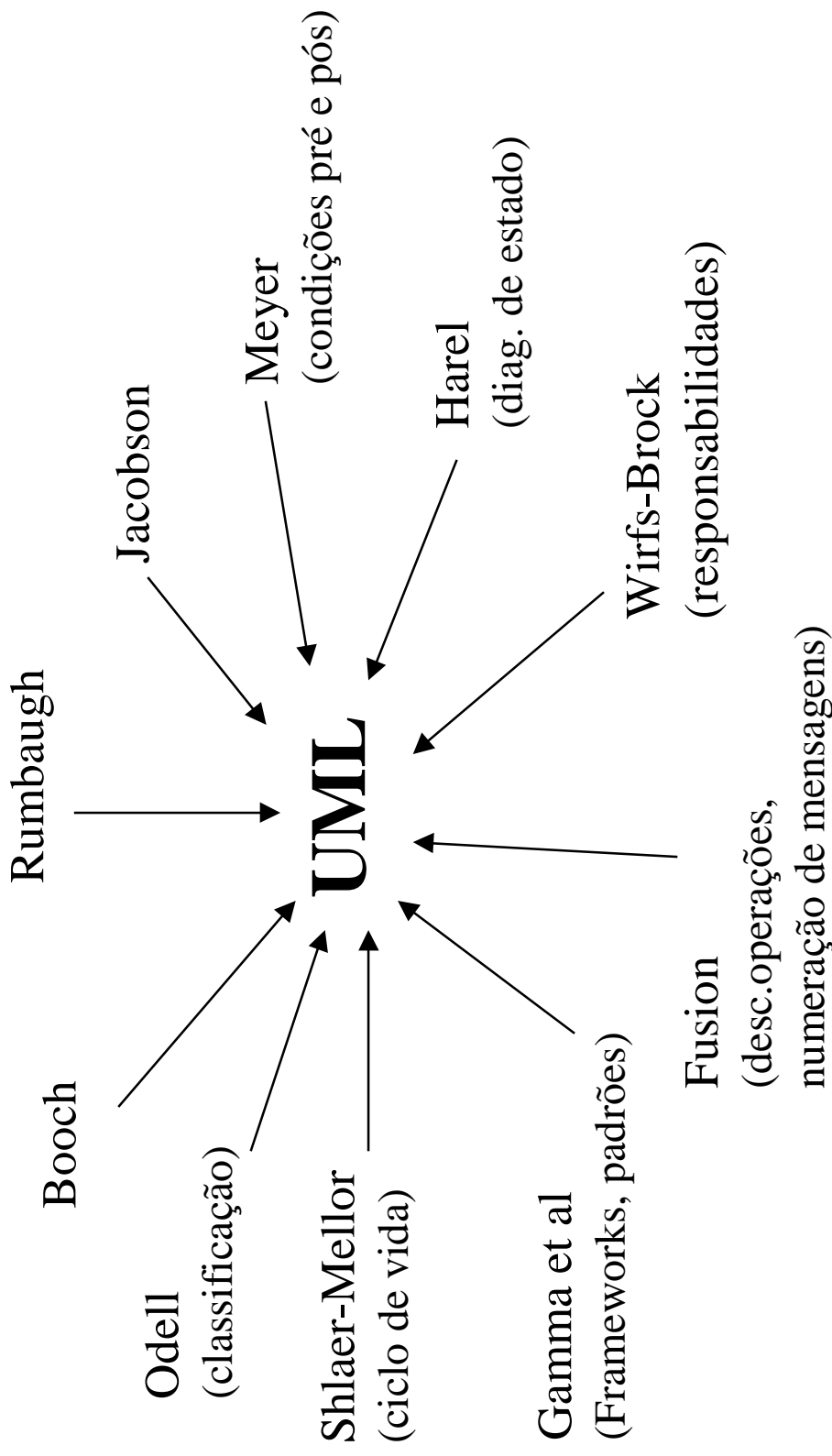
Uma pausa para falar sobre ...

UML – Unified Modeling Language

• • • • • • • • • •

•
•
•

UML - Unified Modeling Language

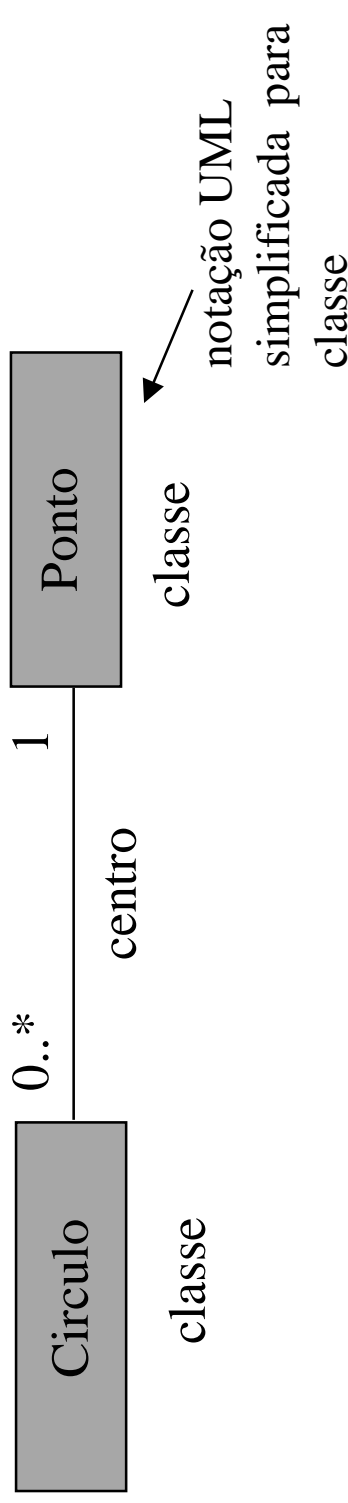


• • • • • • • • • •

-
-
-

Relacionamentos

- Um relacionamento modela uma conexão física ou conceitual entre objetos
- Relacionamentos são bidirecionais



Objetos da classe **Circulo** estão associados a
objetos da classe **Ponto**

- # Relacionamentos - Cardinalidade



Especifica quantas instâncias de uma classe podem, ao mesmo tempo, ser relacionadas com uma única instância de uma classe associada

- •
•
•
•
•
•
•

-
-
-

Relacionamentos - Exemplos



