

Programação Orientada a Objetos e a Linguagem Java

Rodrigo da Cruz Fujioka
rodrigofujioka@gmail.com



Objetivos

www.rodrigofujioka.com

- Apresentar Classes e Métodos
- Trabalhar com o Eclipse
- + Prática

Rodrigo Fujioka - Ling. de Prog 2



Classes

- Definição:
 - Local onde encontram-se definidas as propriedades (variáveis) e o comportamento (métodos) que uma categoria de objetos deve possuir



Classes

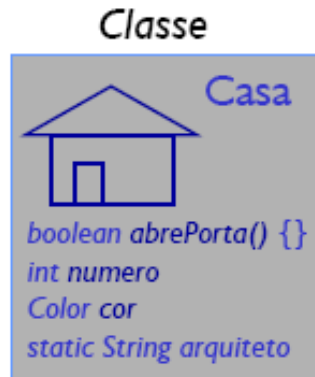
- Definição:
 - Local onde encontram-se definidas as propriedades (variáveis) e o comportamento (métodos) que uma categoria de objetos deve possuir



Classes

- Classes são uma **especificação** para objetos
- Uma classe representa um **tipo de dados** complexo
- Classes descrevem
 - Tipos dos dados que compõem o objeto (o que podem armazenar)
 - Procedimentos que o objeto pode executar (o que podem fazer)

Instâncias da classe Casa (objetos)



```
Casa c1 = new Casa();  
c1.numero = 12;  
c1.cor = Color.yellow;
```



```
Casa c2 = new Casa();  
c2.numero = 56;  
c2.cor = Color.red;
```

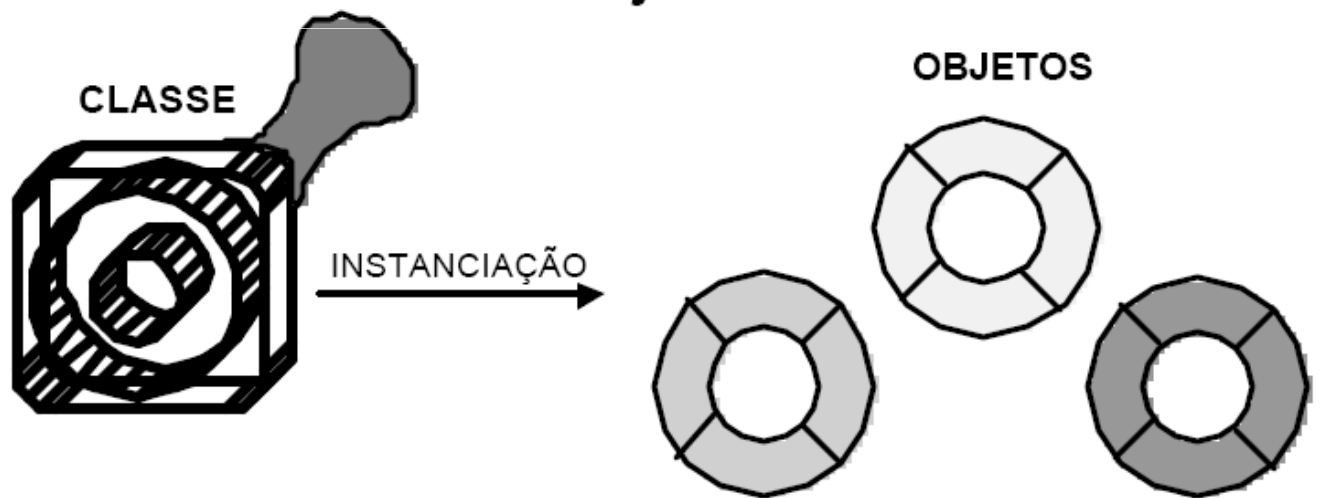


```
Casa c3 = new Casa();  
c3.numero = 72;  
c3.cor = Color.white;  
c3.abrePorta();
```



Classes

- Toda classe define **um novo tipo** de dados
- Valores de um tipo definido por uma classe recebem o nome de **objetos**



Classes

- Estrutura fundamental de programação em Java!
 - Todo e qualquer programa Java deve definir pelo menos uma classe
 - Não há como escrever código Java sem que haja a definição de classes



Classes – Exemplo

www.rodrigofujioaka.com

Nome da Classe.

```
public class Cliente {  
    private String nome;  
    private String endereco;
```

Método.

```
    public void setNome( String novoNome ) {  
        nome = novoNome;  
    }
```

```
}
```

Rodrigo Fujioaka - Ling. de Prog 2

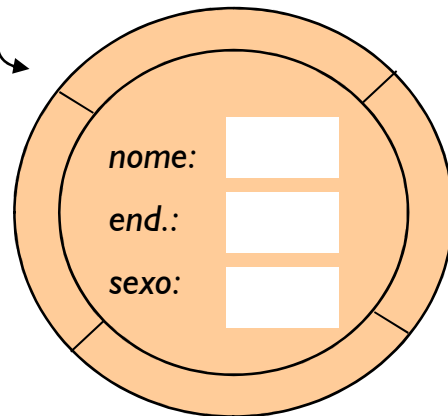


Exemplo de uma classe Java

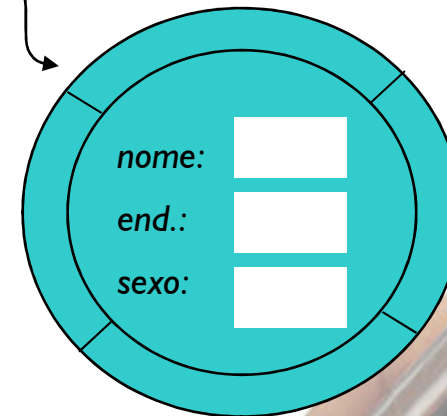
- Instanciação da classe cliente:

```
Cliente c1 = new Cliente();  
Cliente c2 = new Cliente();
```

c1



c2



Classes – Padrão Nomenclatura

www.rodrigofujioka.com

- O nome da classe deve começar com letra maiúscula.
 - Ex: **Conta**, **Cliente**, **Banco**, **Endereco**.
- Nomes compostos não são separados por _. A primeira letra da palavra seguinte é maiúscula.
 - Ex: **PessoaJuridica**, **PessoaFisica**
- Evite abreviações no nome e use nomes com alguma relação com o que a classe modela

Rodrigo Fujioka - Ling de Prog 2



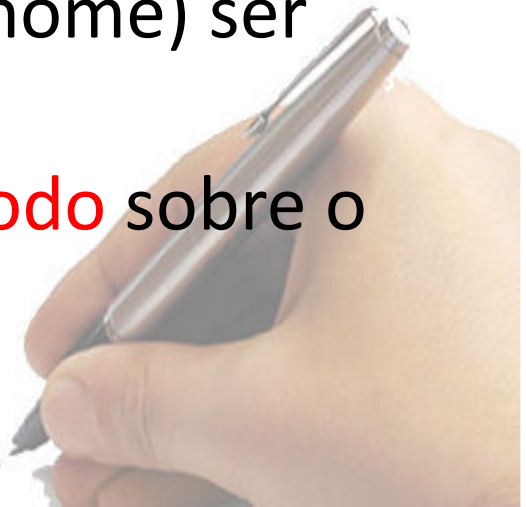
Métodos

- Realizam leitura (`getX()`) do valor dos atributos de um objeto
- Alteram o valor dos atributos (`setX()`)
- A implementação de um método pode utilizar as estruturas lógicas da programação estruturada: decisão, laços, desvio de fluxo, atribuição, etc.



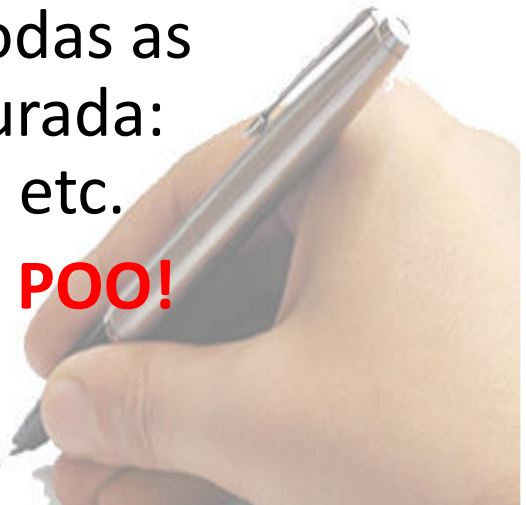
Métodos

- Um método é um **procedimento ou função** que permite aos objetos de uma classe executarem serviços
- É como o objeto implementa suas funcionalidades
- O envio de uma mensagem para um objeto faz o método correspondente (de mesmo nome) ser executado
- A mensagem é a **ativação de um método** sobre o objeto



Métodos

- Métodos podem ter parâmetros
- Métodos poder ler (**getXXX()**) ou alterar (**setXXX()**) os valores das propriedades dos objetos
 - Modificam, portanto, o estado de um objeto
 - Estes métodos, juntos, representam o comportamento de um objeto ou o protocolo de uma classe
- A implementação de um método utiliza todas as estruturas lógicas da programação estruturada: decisão, laços, desvio de fluxo, atribuição, etc.
- Métodos: **é onde as coisas acontecem na POO!**



Dúvidas ?

www.rodrihofujioka.com

? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ?

Rodrigo Fujioka - Ling de Prog 2

