

Linguagens de Programação II

Aula 04 – Orientação à Objetos com Java

Prof. Marc Antonio Vieira de Queiroz

Centro Universitário Filadélfia
Ciência da Computação/Sistemas de Informação

Orientação à Objetos com JAVA

Encapsulamento

Conteúdo

- Encapsulamento
 - Setters
 - Getters
- Exemplos
- Exercícios

Até agora

- Como usamos:
 - Acesso direto à atributos da classe
 - Todo mundo “vê” e altera o estado do objeto
- Como deveríamos usar:

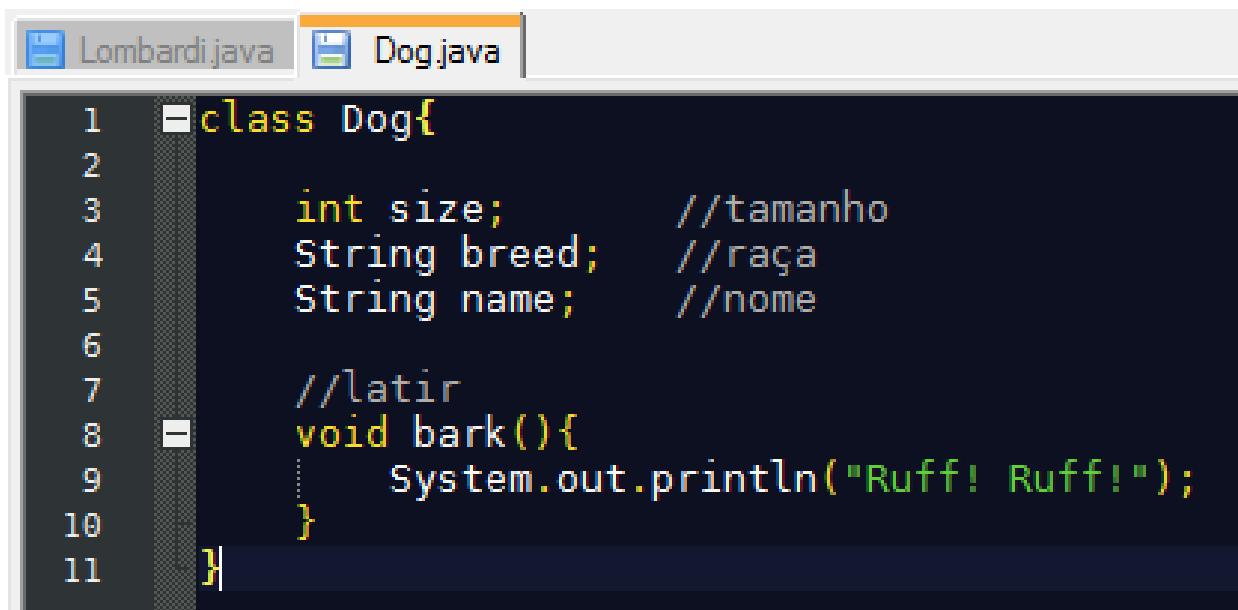
Até agora

- Como usamos:
 - Acesso direto à atributos da classe
 - Todo mundo “vê” e altera o estado do objeto
- Como deveríamos usar:
 - Mudar a visibilidade do atributo/método
 - Métodos para acessar o atributo (**getters**)
 - Métodos para alterar o atributo (**setters**)

Visibilidade

- **Public:**
 - Visível fora da classe
- **Private:**
 - Visível apenas dentro da classe
- **Protected:**
 - Visível apenas para classes filhas

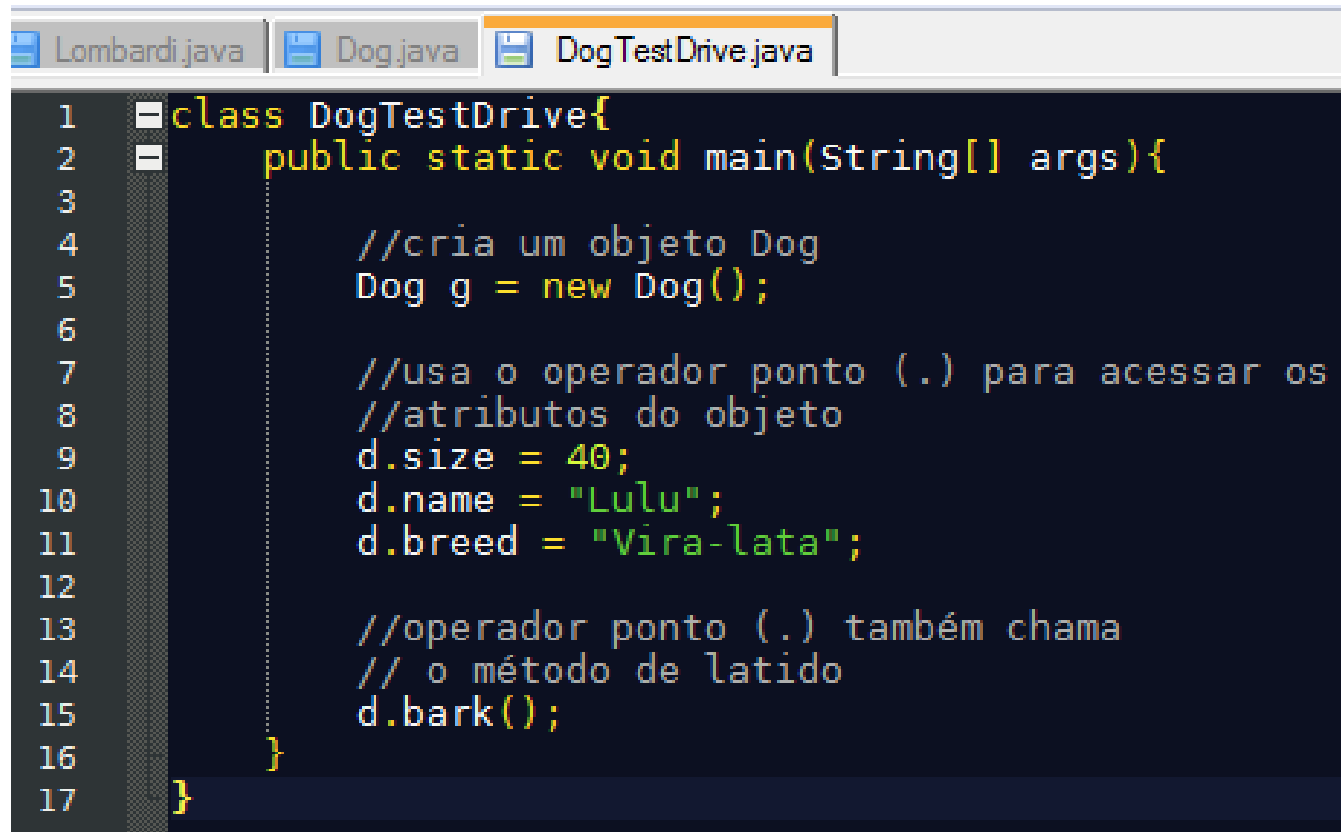
Exemplo 01 – Classe



```
Lombardi.java Dog.java
1  class Dog{
2
3      int size;        //tamanho
4      String breed;    //raça
5      String name;     //nome
6
7      //latir
8      void bark(){
9          System.out.println("Ruff! Ruff!");
10     }
11 }
```

- Quais problemas podemos identificar?

Exemplo 01 – Classe Testadora

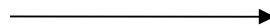


```
1  class DogTestDrive{
2      public static void main(String[] args){
3
4          //cria um objeto Dog
5          Dog g = new Dog();
6
7          //usa o operador ponto (.) para acessar os
8          //atributos do objeto
9          d.size = 40;
10         d.name = "Lulu";
11         d.breed = "Vira-lata";
12
13         //operador ponto (.) também chama
14         // o método de latido
15         d.bark();
16     }
17 }
```

- Existe **encapsulamento**?

Exercício 01

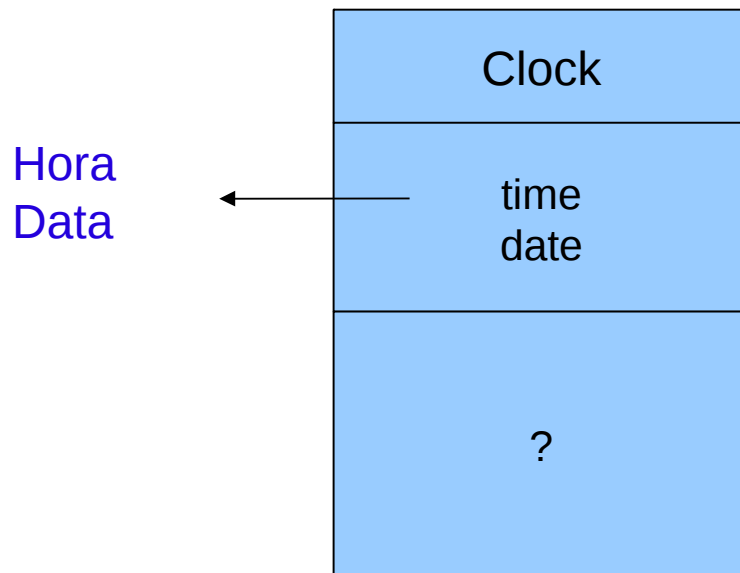
- Qual o jeito certo de encapsular a classe Dog.java ?



?

Exercício 02

- Crie uma classe Clock.java que implementa um objeto relógio usando encapsulamento. Crie também uma classe testadora (ClockTestDrive.java) para validar os métodos/atributos criados.



Exercício 03

- Crie uma classe Pilha.java para implementar uma pilha que armazena doubles em um array estático. Crie também a classe Testadora.



empilha()	→ Inserir elemento
desempilha()	→ Remover elemento
topo()	→ Ver quem está no topo
estáCheia()	→ verificar se está vazia
EstáVazia()	→ verificar se está cheia
Imprimir()	→ mostrar o conteúdo

- Usar **encapsulamento**

Entrega das Implementações

- Via moodle: <http://trab.dc.uel.br/moodle>
- Prazo de entrega: 1 semana

Continua ...

A series of horizontal lines in teal and light blue colors, spanning the width of the slide and partially overlapping the white area below.