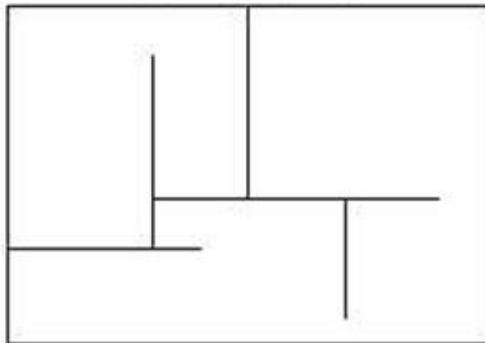


# Classes e Objetos

# Classes e Objetos

Na orientação a objetos tudo se inicia a partir da classe. A classe é uma espécie de molde, a partir do qual são criados objetos do mesmo tipo.

Planta - Classe



Casas - Objetos

# Classes

Objeto do mundo real



<http://animais.colorir.com/caes/cachorro.html>

É representado

Classe

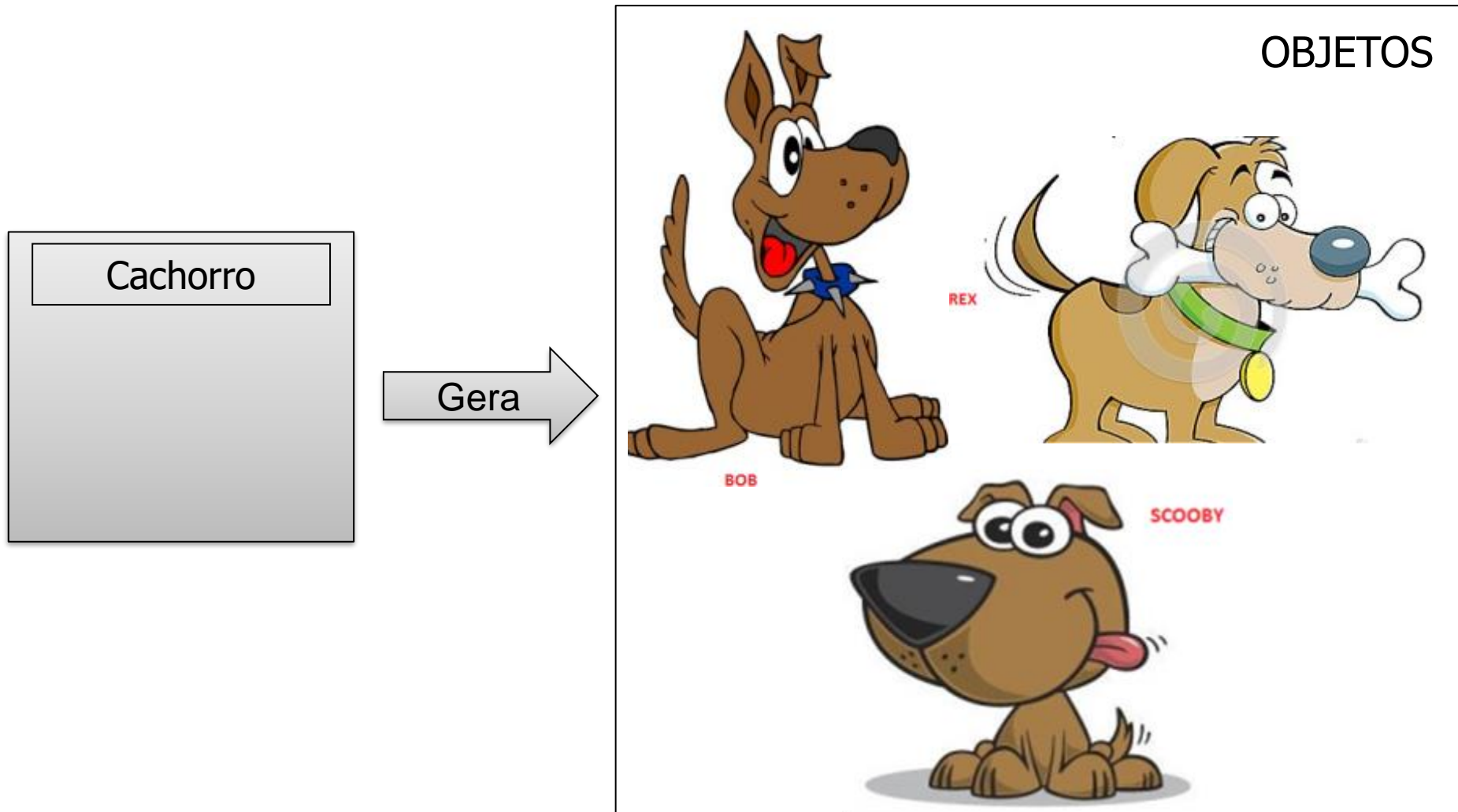
Cachorro

Contém

Atributos

Métodos

# Uma classe gera muitos objetos



## Estrutura

Uma classe é declarada contendo um qualificador (public, abstract, final), a palavra-reservada **class** seguida do nome da classe e de seu corpo entre chaves.

### Sintaxe:

```
qualificador class NomeDaClasse { ... }
```

### Regras para nomes de classes

- ✓ Não podem conter espaços
- ✓ Deve ser iniciado por uma letra (maiúscula) ou '\_' ou '\$'
- ✓ Recomenda-se não utilizar acentos
- ✓ Pode conter números
- ✓ Não pode ser uma palavra reservada da linguagem Java

# Estrutura

`/*`

\* A estrutura básica de uma classe  
\* em Java pode ser observada nesse  
\* pequeno exemplo

`*/`

**Início do bloco de comentário**

**Fim do bloco de comentário**

```
public class NomeDaClasse {
```

```
    /* Corpo da classe
```

```
    definição de atributos e métodos */
```

```
} // Fim da declaração da classe
```

**Início do comentário de linha única**

## Estrutura da classe

- ✓ Possui uma representação na UML.
- ✓ É formada por atributos e métodos.
- ✓ Uma classe define um estado e um comportamento.

## Qualificadores da classe

***public:*** indica que a classe é visível (e por isso pode ser usada) por outras classes;

***final:*** indica que a classe não pode ser herdada (ou redefinida) por outras classes;

***abstract:*** indica que a classe não admite a geração de instâncias.



# Objetos

Quando se cria um objeto, esse objeto adquire:

- ✓ Um espaço em memória para armazenar seu estado (os valores de seu conjunto de atributos, definidos pela classe)
- ✓ Um conjunto de operações que podem ser aplicadas ao objeto (o conjunto de métodos definidos pela classe).

Um objeto é considerado uma instância de classe.

- ✓ Para instanciar um objeto em Java, utiliza-se o operador **new**.

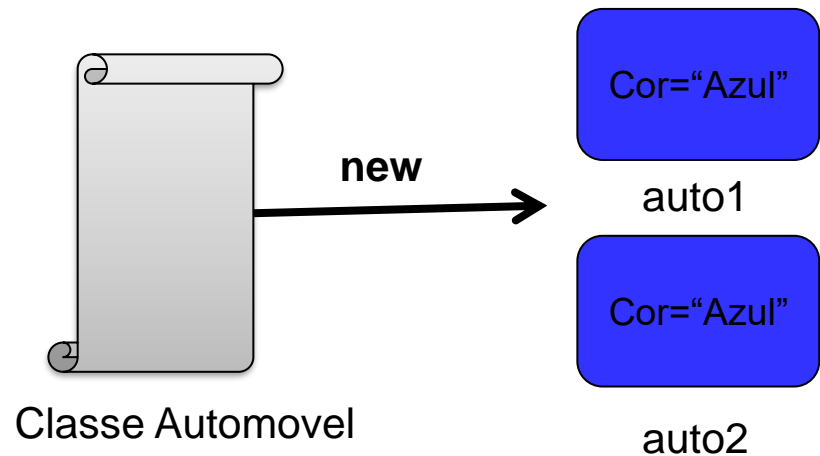
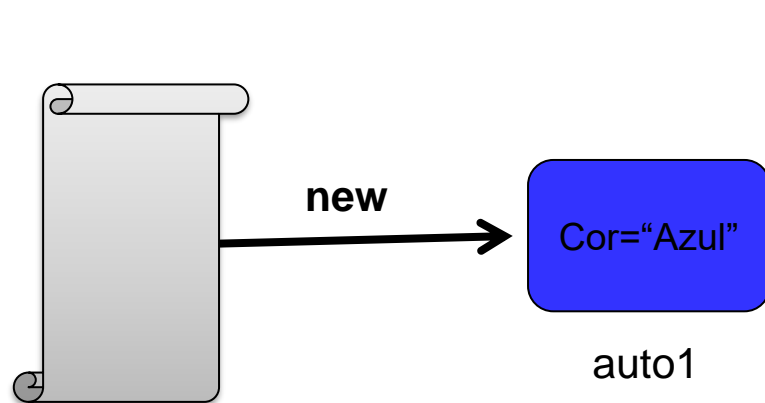
# Variáveis de instância

```
public class Automovel {  
    public String cor = "Azul";  
}
```

Todo objeto do tipo Automovel instanciado terá a sua variável cor inicializada com o valor "Azul"

```
Automovel auto1 = new Automovel();
```

```
Automovel auto2 = new Automovel();
```



Classe Automovel

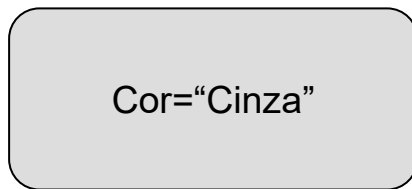
Classe Automovel

## Variáveis de instância

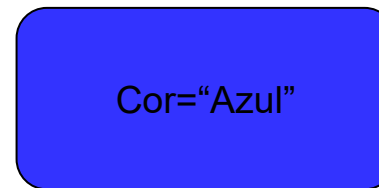
Cada um dos objetos do tipo Automovel tem sua própria variável de instância.

Uma alteração em uma variável do objeto auto1 não afeta a variável do objeto auto2.

```
auto1.cor = "Cinza";
```



auto1



auto2

# Variáveis de instância x Variáveis locais

## Variáveis de instância:

- ✓ Declaradas na classe;
- ✓ Cada objeto (instância) da classe tem uma instância separada da variável;
- ✓ Existe enquanto o objeto existir: **antes e depois** de chamadas aos métodos;

## Variáveis locais:

- ✓ Declaradas no corpo do método;
- ✓ Só podem ser utilizadas nesse método;
- ✓ Só existem **durante** a execução do método;

## Referências e criação de objetos

```
3 public class Motor {  
4     boolean status;  
5  
6     public void ligarMotor() {  
7         status = true;  
8     }  
9  
10    public void desligarMotor() {  
11        status = false;  
12    }  
13 }
```

## Referências e criação de objetos

- Agora, tanto `motor1`, quanto `motor2` estão apontando (referenciando) para um mesmo objeto, na posição 2 de memória
- O objeto na posição 2 também pode ser manipulado através da referência `motor2`

```
Motor motor1 = null;
```

```
motor1 = new Motor();
```

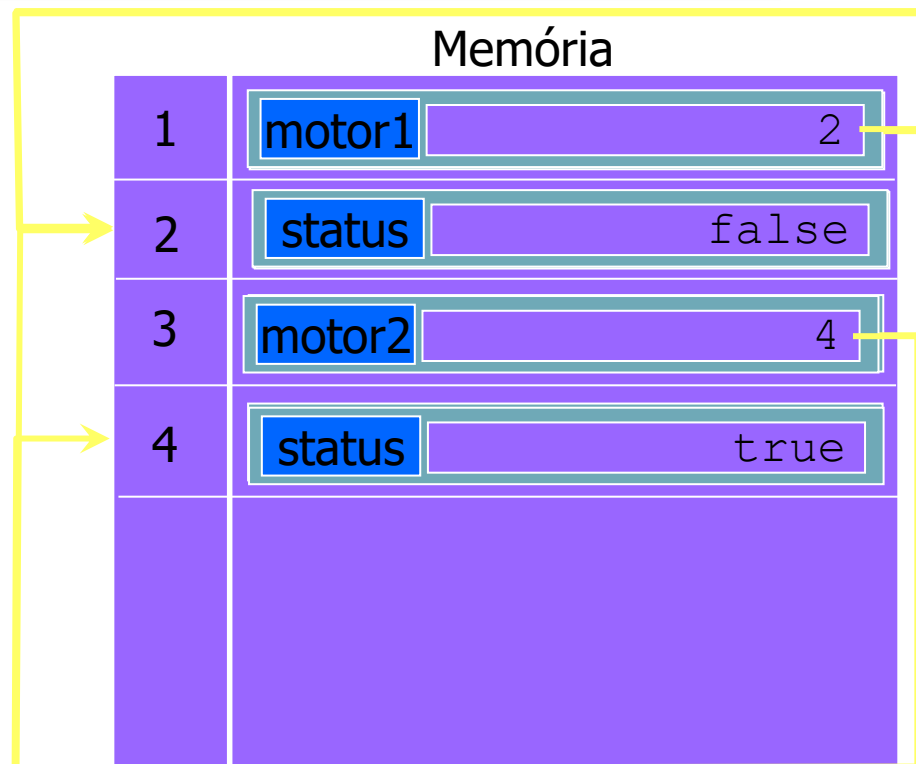
```
motor1.ligarMotor();
```

```
Motor motor2 = motor1;
```

```
motor2.desligarMotor();
```

```
motor2 = new Motor();
```

```
motor2.ligarMotor();
```



## Exercícios

1. Implemente uma classe chamada **Aluno**. Os atributos da classe são RA, nome e curso e um método chamado mostrar para apresentar seus valores em tela. Após criar a classe Aluno:
  - a) Faça uma classe chamada UsaAluno.
  - b) Crie dois objetos da classe Aluno.
  - c) Altere os valores dos atributos desses objetos e os exiba em tela.
2. Implemente uma classe chamada garrafa. A garrafa deve conter um conteúdo em ml e um status para saber se já foi aberta.
3. Faça uma aplicação identificando as classes, atributos e métodos necessários para modelar e implementar uma TV contendo um status para saber se está ligada ou desligada e valores do nível de volume e do número do canal. Métodos para mudar o canal (de um em um) e aumentar/reduzir o volume. A troca de volume e canal apenas devem funcionar com a TV ligada.