



Lendo ou escrevendo campos de objetos

Ler e escrever propriedades em objetos é uma das tarefas mais comuns em um programa java. Para acessar um atributo, usamos o operador `.` (ponto), junto a uma variável de referência para um objeto. Veja a seguinte classe:

```
class Carro {  
    String modelo;  
    int ano;  
  
    public Carro() { ano = 2014; }  
  
    public String getDadosDeImpressao() {  
        return modelo + " - " + ano;  
    }  
  
    public void setModelo(String m) {  
        this.modelo = m;  
    }  
}
```

[COPIAR CÓDIGO](#)

Vamos escrever um código para usar essa classe:

```
Carro a = new Carro();  
a.modelo = "Palio";      // acessando diretamente o atributo  
a.setModelo("Palio");    // acessando o atributo por um método
```

```
// acessando o método e passando o retorno como argumento para
// o método println
System.out.println(a.getDadosDeImpressao());

}
```

[COPIAR CÓDIGO](#)

As linhas 2 e 3 têm exatamente o mesmo efeito. Como iniciamos o valor da propriedade `ano` no construtor, ao chamar o método `imprimeDados`, o valor 2014 é exibido junto ao nome do modelo.

Quando estamos dentro da classe, não precisamos de nenhum operador para acessar os atributos de instância da classe. Opcionalmente, podemos usar a palavra-chave `this`, que serve como uma variável de referência para o próprio objeto onde o código está sendo executado:

```
class Carro{
    int ano;
    int modelo;

    public Carro(){
        modelo = "Indefinido";    // acessando variável de
                                   // instancia sem o this
        this.ano = 2014;          // acessando com o this.
    }
}
```

[COPIAR CÓDIGO](#)