



Veja a classe `Fatura`. Ela tem um método chamado `converte()`, que recebe um booleano. Veja só uma utilização desse método:

```
double valor1 = fatura.converteValor(true);  
double valor2 = fatura.converteValor(false);
```

De olhar para o código acima, você consegue saber o que esse método faz? O que significa esse parâmetro booleano? Veja a implementação:

```
public double converteValor(boolean dolar) {  
    double taxa = 1;  
    if (dolar) taxa = 2.7;  
    return valorMensal * taxa;  
}
```

Veja que o booleano indica se devemos converter o valor para dólar ou não. Se o desenvolvedor não olhar o código-fonte, ele não entenderá. Parâmetros booleanos são uma péssima maneira para você mudar o comportamento do seu código.

Vamos melhorar isso. Uma solução é quebrar isso em dois métodos, específicos e bem semânticos, por exemplo `emDolar()` e `emReal`:

```
public double emDolar() {  
    double taxa = 2.7;  
    return valorMensal * taxa;  
}  
  
public double emReal() {  
    return valorMensal;  
}
```

Evite ao máximo parâmetros booleanos. Favoreça o uso de métodos separados, sobrecarga de métodos ou até mesmo o uso de Factories (do padrão de projeto).

