

## Modificando a property internamente

Vimos que quando acessamos as properties, na verdade acessamos ou o seu getter durante a leitura ou o seu setter durante uma atribuição.

Entretanto, quando modificamos o setter adicionando um corpo vazio, vimos que a property perdeu a habilidade de ser modificada. Para que a modificação do valor da property seja possível, precisamos implementar essa ação dentro do setter.

### Modificando o valor interno da property

Como vimos o setter de uma property recebe o parâmetro `value` que indica o valor recebido no momento da atribuição, ou seja, basta apenas a gente atribuir esse parâmetro para property. Entretanto, quando tentamos fazer uma atribuição da seguinte maneira:

```
var valor: BigDecimal = valor
set(value) {
    valor = value
}
```

Entramos em um caso no qual é realizada uma chamada recursiva sem fim, isto é, se executamos o nosso código, recebemos uma [StackOverflowError](https://docs.oracle.com/javase/7/docs/api/java/lang/StackOverflowError.html) (<https://docs.oracle.com/javase/7/docs/api/java/lang/StackOverflowError.html>) justamente por fazer uma chamada sem fim. Da mesma maneira, se tentarmos modificar o getter com o seguinte código:

```
var valor: BigDecimal = valor
set(value) {
    valor = value
}
get() {
    return valor
}
```

Ao fazer a leitura da property receberemos o mesmo erro por causa da recursão sem fim.

### Utilizando os backing fields para modificar o valor da property

Para que seja possível modifica o valor interno de uma property, o Kotlin nos oferece um recurso conhecido como backing fields. Em outras palavras, utilize a *keyword* `field` para modificar o valor da property.

Aproveitando, modifique também o getter com a chamada do getter `get() {}` e retorne o backing field. Execute o mesmo código e veja o que acontece.