

05

A sintaxe get

Transcrição

Vamos ver como podemos "enxugar" a nossa classe `Negociacao`. Nós criamos métodos de leitura para poder acessar às variáveis que convencionamos que são privadas, ou seja, só a própria classe poderá acessá-las. Mas existe um atalho que poderá ser usado quando queremos acessar uma propriedade: usaremos o `get`.

```
get volume() {  
  
    return this._quantidade * this._valor;  
  
}  
  
get data() {  
    return this._data;  
}  
  
get quantidade() {  
    return this._quantidade;  
}  
  
get valor() {  
    return this._valor;  
}
```

Observe que fizemos pequenos ajustes no código. O `get` foi seguido de um método, que tem o nome das propriedades antes de adicionarmos o `_`. Essa alteração irá trazer mudanças também no arquivo `index.html`. Agora, poderemos substituir o `n1.getQuantidade` por `n1.quantidade`.

```
<script>  
  
var n1 = new Negociacao(new Date(), 5, 700);  
  
console.log(n1.quantidade);  
console.log(n1.data);  
console.log(n1.valor);  
console.log(n1.volume);  
</script>
```

Estamos criando uma propriedade getter de acesso à leitura. E mesmo sendo um método, poderemos acessá-lo como uma propriedade. Mas, debaixo dos panos, ele continuará sendo executado como um método.

The screenshot shows a browser developer tools console window. At the top, there's a URL bar with 'file:///Users/flavio/Desktop/aluraframe/client/index.html'. Below it, the title 'Negociações' is displayed. The main area contains form fields for 'Data' (dd/mm/aaaa), 'Quantidade' (1), and 'valor' (0,0). The 'Console' tab is selected, showing the following log output:

```

5
Tue May 10 2016 13:58:10 GMT-0300 (BRT)
700
3500

```

A red box highlights the first four lines of the log output. On the right side of the console, there are file paths for each log entry: index.html:65, index.html:66, index.html:67, and index.html:68.

Veja que os valores foram acessados corretamente, inclusive o valor de `volume`. Então, por mais que os valores se apresentem como uma propriedade, por debaixo dos panos, está sendo chamado um método. Agora, mesmo que definíssemos o valor `1000` para `quantidade`, ele não seria aplicado.

```

<script>

var n1 = new Negociacao(new Date(), 5, 700);
n1.quantidade = 1000;
console.log(n1.quantidade);
console.log(n1.data);
console.log(n1.valor);
console.log(n1.volume);

</script>

```

O valor `1000` não será repassado para o `_quantidade`. O valor continuará sendo `5`:

The screenshot shows a browser developer tools console window. The log output is identical to the previous one:

```

5
Tue May 10 2016 13:59:07 GMT-0300 (BRT)
700
3500

```

A red arrow points to the first '5' in the log output. On the right side of the console, there are file paths for each log entry: index.html:66, index.html:67, index.html:68, and index.html:69.

Isto acontece quando a propriedade é um getter (ou seja, é de leitura), não podemos atribuir para a mesma um valor. Mas ainda podemos acessá-la usando o `n1._quantidade`.

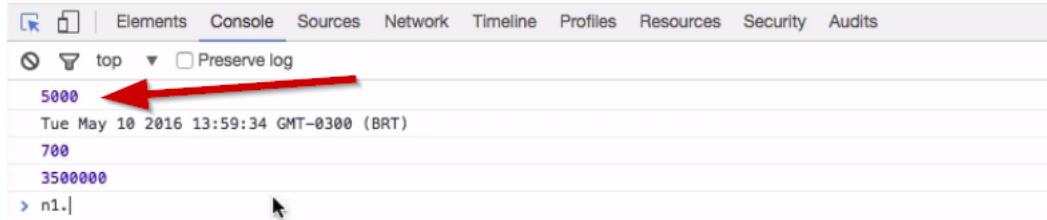
```

<script>

var n1 = new Negociacao(new Date(), 5, 700);
n1.quantidade = 1000;
n1._quantidade = 5000;
console.log(n1.quantidade);
console.log(n1.data);
console.log(n1.valor);
console.log(n1.volume);

</script>

```



```
5000
Tue May 10 2016 13:59:34 GMT-0300 (BRT)
700
3500000
> n1.|
```

No console, vemos que o valor da `quantidade` foi alterado para `5000`. Mas a propriedade `quantidade` é somente de leitura, não aparece o `_`. Assim, se o programador tentar atribuir um valor, ele não será setado. Será que existe uma forma de garantirmos que uma negociação, depois de criada, não será alterada?