

## Para saber mais: Trabalhando com Calendar

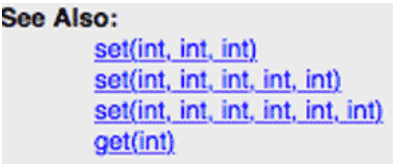
Antes da nova API de Data & Time no Java 8 a forma mais correta era usar a classe `java.util.Calendar`. Diferentemente da classe `java.util.Date` a classe `Calendar` não possui bugs e só se preocupe em trabalhar com Data (sem parsing e formatação), mas a classe possui um design nada intuitiva.

Para criar uma instância do `Calendar` devemos utilizar o método do fábrica `getInstance()`:

```
Calendar data = Calendar.getInstance();
```

Aqui já podemos ver o primeiro problema: Um `Calendar` representa data E hora, não existia nada no Java padrão para distinguir os dois (antes do Java 8, claro).

Para setar uma data específica devemos usar o método `set` que também não é nada expressivo. Veja a imagem abaixo que mostra uma parte da documentação do método `set`:



**See Also:**  
[set\(int, int, int\)](#)  
[set\(int, int, int, int, int\)](#)  
[set\(int, int, int, int, int, int\)](#)  
[get\(int\)](#)

Qual é o significado de cada `int`? Realmente, ficou muito confuso e só a documentação aberta pode nos salvar:

[https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html#set\(int,%20int\)](https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html#set(int,%20int))

(<https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html#set%28int,%20int%29>)

O primeiro método `set` define o ano, mês e dia por exemplo:

```
data.set(2016, 11, 24);
```

Você poderia pensar que colocamos a data 24/11/2016, mas não é verdade. O mês começa com ZERO!!! Zero é janeiro! A classe `Calendar` está cheia de supressas e sempre foi alvo de críticas da comunidade.

Por exemplo, existe mais uma "versão" do método `set` para mudar, por exemplo, o mês:

```
data.set(Calendar.MONTH, 10); //agora é o mês novembro :)
```

Novamente, nada intuitivo e expressivo, pois poderia ter um método como `setMonth`.

As coisas não param por aí, imagina que gostaria de voltar ou subtrair dois meses:

```
data.add(Calendar.MONTH, -2);
```

É isso mesmo, não há um método `minus`, apenas o `add` que recebe uma valor negativo.

