

DateHelper é um Factory?

Hora da reflexão! Não é necessário responder este exercício, apenas "meditar" sobre o que é exposto.

O que você acha de promovermos nosso `DateHelper` para `Factory` ? Pode ser que isso tenha passado em sua cabeça, mas na verdade nosso `DateHelper` não é um `Factory` .

O padrão de projeto `Factory` ocorre quando temos uma classe que nos ajuda a criar um objeto complexo, ou seja, ela esconde de nós os detalhes de criação desse objeto. É por isso que uma classe `Factory` possui apenas um método. Faz sentido, porque se tivéssemos que chamar mais de um para criar um objeto a responsabilidade de sua criação cairia em nosso colo.

Já nosso `DateHelper` , que está mais para um `DateConverter` (por que não pensei nesse nome antes?!), tem como responsabilidade converter datas. Ele possui dois métodos que focam a criação de texto para data e data para texto. Apesar da classe isolar a complexidade de construção de uma data, ela atua mais como um conversor do que uma `Factory` .