

Builder vs Recursos de linguagem

Vejam os dois exemplos de construção de classe: O primeiro utiliza recursos da linguagem Python, o segundo aplica o padrão de projeto Builder:

a) Apenas recurso da linguagem Python

```
nota_fiscal = Nota_fiscal(  
    cnpj='012345678901234',  
    razao_social='FHSA Limitada',  
    itens=itens  
)
```

b) Utilizando o padrão de projeto Builder. Utiliza a classe `Criador_de_nota_fiscal` para criar uma nota:

```
nota_fiscal = (Criador_de_nota_fiscal()  
    .com_razao_social('FHSA Limitada')  
    .com_cnpj('012345678901234')  
    .com_itens(itens)  
    .constroi())
```

E claro, temos o código do Builder:

```
# -*- coding: UTF-8 -*-  
# criador_de_nota_fiscal.py  
from nota_fiscal import Nota_fiscal, Item  
  
class Criador_de_nota_fiscal(object):  
  
    def __init__(self):  
  
        self.__razao_social = None  
        self.__cnpj = None  
        self.__data_de_emissao = None  
        self.__detalhes = ''  
        self.__itens = None  
  
    def com_razao_social(self, razao_social):  
        self.__razao_social = razao_social  
        return self  
  
    def com_cnpj(self, cnpj):  
        self.__cnpj = cnpj  
        return self  
  
    def com_data_de_emissao(self, data_de_emissao):  
        self.__data_de_emissao = data_de_emissao  
        return self  
  
    def com_itens(self, itens):
```

```
self.__itens = itens
return self

def constroi(self):
    if self.__razao_social is None:
        raise Exception('Razão social deve ser preenchida')
    if self.__cnpj is None:
        raise Exception('CNPJ deve ser preenchido')
    if self.__itens is None:
        raise Exception('Itens deve ser preenchido')

    return Nota_fiscal(
        razao_social=self.__razao_social,
        cnpj=self.__cnpj,
        itens=self.__itens,
        data_de_emissao=self.__data_de_emissao,
        detalhes = self.__detalhes
    )
```

Dentro do contexto de criação de uma nota fiscal, qual das duas abordagens você prefere? Por quê?

Responda

INSERIR CÓDIGO	FORMATAÇÃO