

Colocando em prática

Chegou a hora de praticar! Preparado? É importante que você tenha entendido primeiro o conceito do design pattern Strategy, só assim conseguirá implementá-lo. É claro que você ainda pode esbarrar em alguma dúvida no que diz respeito à algum recurso da linguagem Python envolvido no pattern em questão. Se isso acontecer, não se preocupe, você pode recorrer à explicação textual ou ao vídeo do capítulo para ajudá-lo.

Temos aquele cenário que já vimos no capítulo: nosso `calculador_de_impostos` precisa calcular um imposto para dado orçamento. Temos a classe `Orçamento`:

```
# -*- coding: UTF-8 -*-
# orcamento.py
class Orcamento(object):
    def __init__(self, valor):
        self._valor = valor

    @property
    def valor(self):
        return self._valor
```

E também funções de cálculo de impostos:

```
# -*- coding: UTF-8 -*-
# impostos.py
def calcula_ICMS(orcamento):
    return orcamento.valor * 0.1

def calcula_ISS(orcamento):
    return orcamento.valor * 0.06
```

Por fim, temos nossa classe `Calculador_de_impostos` que realizará o cálculo de determinado imposto. Veja que passamos uma string com o nome do imposto e realizamos uma série de 'if's dentro do método `realiza_calculo` para saber qual imposto executar:

```
# -*- coding: UTF-8 -*-
# calculador_de_impostos.py

from impostos import calcula_ISS, calcula_ICMS

class Calculador_de_impostos(object):

    def realiza_calculo(self, orcamento, imposto):

        if 'ICMS' == imposto:
            icms_calculado = calcula_ICMS(orcamento)
            print icms_calculado
        elif 'ISS' == imposto:
            iss_calculado = calcula_ISS(orcamento)
            print iss_calculado
```

```
if __name__ == '__main__':  
  
    from orcamento import Orcamento  
  
    orcamento = Orcamento(500.0)  
    calculador_de_impostos = Calculador_de_impostos()  
    calculador_de_impostos.realiza_calculo(orcamento, 'ICMS') # imprime 50.0  
    calculador_de_impostos.realiza_calculo(orcamento, 'ISS') # imprime 30.0
```

Se executamos nosso arquivo `calculador_de_impostos.py` :

```
python calculador_de_impostos.py
```

Tudo funciona, porém o código deixa a desejar. Por quê? Imagine novos impostos? Teremos que adicionar mais condições em `realiza_calculo`, isto é, a classe `Calculador_de_impostos` está muito suscetível à mudanças!

É a partir daqui que você começará. Depois de ter criado arquivos com os códigos anteriores, faça as alterações necessárias para que não seja mais necessário vários `if's` dentro de `calculador_de_impostos`. Como? Redesenhando-o seu código para utilizar o padrão `Strategy`. No final, cole o código da sua implementação aqui. Ao terminar, o recurso deve ser parecido com o que foi mostrado ao longo do capítulo.

Bom estudo!

Responda

INSERIR CÓDIGO	FORMATAÇÃO

