

Refletindo sobre o código

Nossos orçamentos podem ter diferentes estados durante o seu ciclo de vida. Um orçamento nasce "Em aprovação", e pode virar "Aprovado" ou "Reprovado". Ao final de todo o processo, deverá ser "Finalizado".

Dependendo do estado que o orçamento se encontra, algumas ações podem ser diferentes. Podemos resolver isso facilmente com a seguinte solução procedural.

```
# -*- coding: UTF-8 -*-
# Orcamento.py
class Orcamento(object):

    EM_APROVACAO = 1
    APROVADO = 2
    REPROVADO = 3
    FINALIZADO = 4

    def __init__(self):
        self.__itens = []
        self.estado_atual = 1
        self.__desconto_extra = 0.0

    def aplica_desconto_extra(self):

        if self.estado_atual == Orcamento.EM_APROVACAO:
            self.__desconto_extra+= self.valor * 0.05
        elif self.estado_atual == Orcamento.APROVADO:
            self.__desconto_extra+= self.valor * 0.02
        elif self.estado_atual == Orcamento.REPROVADO:
            raise Exception('Orçamentos reprovados não recebem desconto extra')
        elif self.estado_atual == Orcamento.FINALIZADO:
            raise Exception('Orçamentos finalizados não recebem desconto extra')

# quando a propriedade for acessada, ela soma cada item retornando o valor do orçamento
@property
def valor(self):
    total = 0.0
    for item in self.__itens:
        total+= item.valor
    return total - self.__desconto_extra # valor agora leva em consideração o desconto aplicado

if __name__ == '__main__':

    orcamento = Orcamento()
    orcamento.adiciona_item(Item('Item A', 100.0))
    orcamento.adiciona_item(Item('Item B', 50.0))
    orcamento.adiciona_item(Item('Item C', 400.0))

    print 'Valor sem desconto extra %s' % (orcamento.valor)
    orcamento.aplica_desconto_extra()
    print 'Valor com desconto extra (em aprovação) %s' % (orcamento.valor)
```

```
orcamento.estado_atual = Orcamento.APROVADO
orcamento.aplica_desconto_extra()
print 'Valor com desconto extra (aprovado) %s' % (orcamento.valor)
```

A solução funciona perfeitamente (o resultado é o esperado), ainda assim, qual é o problema dessa abordagem procedural?

Responda

INSERIR CÓDIGO	FORMATAÇÃO