



Pattern Composite per il Preventivatore



Descrizione della Soluzione

Per questo problema, il pattern Composite è la scelta ideale perché:

- Abbiamo una struttura ad albero con sezioni, sottosezioni ed elementi base
- Vogliamo trattare sia gli elementi singoli che i gruppi di elementi in modo uniforme
- Il calcolo dei prezzi segue una logica ricorsiva dove il prezzo di una sezione dipende dai suoi componenti



Vantaggi

- Permette di trattare oggetti singoli e composizioni di oggetti in modo uniforme
- Facilita l'aggiunta di nuovi tipi di componenti
- Semplifica la struttura del client
- Rende più facile aggiungere nuove funzionalità all'intera struttura



Svantaggi

- Può rendere il design troppo generico
- Può essere difficile limitare i tipi di componenti che possono essere aggiunti
- Richiede una buona comprensione della ricorsione per l'implementazione



Struttura del Pattern

- `ComponentePreventivo`: Interfaccia base per tutti gli elementi
- `ElementoBase`: Rappresenta gli elementi foglia (prodotti, servizi)
- `Sezione`: Rappresenta i nodi composti (sezioni e sottosezioni)



Implementazione Semplificata

```
class ComponentePreventivo(ABC):
    def __init__(self, nome: str, sconto: float = 0):
        self.nome = nome
        self.sconto = sconto

class ElementoBase(ComponentePreventivo):
    def __init__(self, nome: str, prezzo: float, sconto: float = 0):
```

```
        super().__init__(nome, sconto)
        self.prezzo = prezzo

class Sezione(ComponentePreventivo):
    def __init__(self, nome: str, sconto: float = 0):
        super().__init__(nome, sconto)
        self.componenti = []
```



Esempio di Utilizzo

```
preventivo = Sezione("Preventivo Completo")
sez_macchina = Sezione("Macchina base", 0.05)
sez_macchina.aggiungi_componente(ElementoBase("Macchina", 20000))
```



Output di Esempio

```
Prezzo base totale: €20000.00
Prezzo scontato totale: €19000.00
```



Note Implementative

- La struttura permette di aggiungere facilmente nuove sezioni e sottosezioni
- Gli sconti vengono applicati in cascata, rispettando la gerarchia
- Il calcolo dei prezzi è completamente trasparente per il client
- La manutenibilità è elevata grazie alla separazione delle responsabilità