Кратко обяснение:

Composite Pattern е шаблон за структуриране и неговата идея е индивидуални обекти или група от такива да изглеждат и да се използват по един и същи начин. Във всяка група обекти може да има нова група обекти или самостоятелен обект, като така се създава дървовидна структура.

Приложимост:

* Когато трябва да се направи дървовидна структура от данни.
* Когато клиентите могат да бъдат в състояние да пренебрегнат разликата между композиции от обекти и отделни обекти. Клиентите ще третират всички обекти в композитната структура равномерно.

Структура:



Последствия:

* Composite Pattern дефинира клас йерархии състоящи се от примитивни обекти и композитни обекти. Примитивните обекти може да се състоят от по-сложни обекти, които от своя страна да се също композитни и т.н. рекурсивно. Където клиентът очаква примитивен тип може да се получи ситуация, в която да взима композитен обект.
* Composite Pattern прави клиента прост и изчистен. Клиентите могат да третират композитни структури и индивидуално обекти по един и същи начин. Клиентите обикновено не знаят дали използват примитивент тип(Leaf) или композитен компонент. Това опростява клиентския код, защото се избягва писането на условни оператори и блокове за класовете които дефинират композицията.
* Composite Pattern прави по-лесно добавянето на нови видове компоненти. Новосъздадените листа(подкласове) работят автоматично със съществуващите структури и клиентския код. Също така клиентът не трябва да се променя заради новосъздадения клас.
* Composite Pattern може да направи своя дизайн прекалено общ. Недостигът на лесното добавяне на компоненти е, че така се увеличава трудността да се ограничат компонените на композицията. Има случаи, в които се нуждаем само от едни определени компоненти. С използването на Composite Pattern не може да разчитате на това. Трябва да се правят проверки по време на изпълнение на програмата.

Използван в:

Примери за използването на Composite може да се намерят в почти всички обектно-ориентирани системи.

* RTL Smalltalk compiler framework Et++
* Smalltalk

Свързани шаблони:

* Chain of Responsibility
* Decorator
* Flyweight
* Iterator
* Visitior