الگوی composite

هر جایی که بخواهیم یک ساختار درختی از اشیا ایجاد کنیم از این الگو استفاده میکنیم.

نمونه ها در برنامه نویسی

این الگو در برنامه نویسی وب فرم هم استفاده شده ، نمونه ی دیگر در منو های سایت که ساب منو دارند.

اشیا در این الگو به دو دسته تقسیم می شوند :

- اشیا گروهی

- اشیا تکی

چند نکته :

1. همه ی اشیاء از Component ارث بری کرده باشد.
2. پیاده سازی پیشفرض برای متدهای Component که Composite ها و Leaf ها فقط در صورت لزوم پیاده سازی را تغییر دهند.
3. این الگو برای ساختار درختی است و هر مولفه باید یک پدر داشته باشد.
4. اشیا فرزند برای سادگی کار میتوانند یک ارجاع به شی پدر داشته باشند.
5. در این الگو میتوانیم از الگوی Flyweight استفاده کنیم
6. Client یی که از کلاس های ما استفاده میکند نباید تفاوت Composite ها و Leaf ها برایش مهم باشد و به یک شکل باید با آن ها کار کند. به همین علت بهترین کار این است که همه عملیات مشترک را در Component پیاده سازی کنیم.
7. هرجا که از این الگو استفاده میکنیم میتوانیم از الگوی Visitor هم استفاده کنیم.
8. برای قابل شمارش بودن را به Composite اضافه کنیم میتوانیم از الگوی Iterator استفاده کنیم.

مزایا

1. ایجاد اشیا تکی یا گروهی بسیار ساده است.بدون تغییر در Client ( به راحتی میتوانیم Composite و Leaf اضافه کرد.)
2. ساده بودن الگوریتم پردازش

معایب

1. Component در اغلب موارد بسیار سنگین و شلوغ می شود.
2. تهیه روابط مشترک برای کلاس های متفاوت سخت است.
3. درخت پویا نیست ( اگر جایی ما Leaf تولید کردیم دیگه نمیتوان آن را رشد داد و برایش فرزند تولید کرد.)

موارد استفاده از این الگو

1. زمانی که ساختار درختی از اشیا داریم
2. زمانی که نیاز داریم کلاینت به یک شکل با اشیا گروهی و تکی کار کند.

الگوهای مرتبط با این الگو

1. Decorator
2. Iterator
3. Visitor
4. Flyweight