

جلسه 5:

Composite structure diagram:

از مهم ترین ویژگی های این diagram این است که میتوان object ها و ارتباط بین اون ها و کلاس های در برگیرنده اون ها رو در این روش دید.

با این diagram ساختار های تو در تو و پیچیده را میتوان نشان داد. در سیستم های جدید ، کلاس ها تعداد زیادی object تو در تو هستند (nested) به همین دلیل ، این diagram برای این کار مناسب است.

- از مهم ترین نکات این diagram ها این است که نباید object های یک کلاس را که کار های متفاوتی انجام میدهند را با هم در یک کلاس ریخت چون این دقیقاً مخالف هدف اصلی ما در استفاده از oop (object oriented programming) هست و این باعث میشود که سیستم ای که طراحی کرده ایم را قادر به maintain کردن نیستیم.

روند این اشتباهات به این شکل است:

پر کردن کلاس از object های نا مرتبت ← کاهش تغییر پذیری ← نگه داری از سیستم ها سخت تر و سیستم ها میمیرند.

از نکات ای که باعث میشود یک سیستم بمیرد:

1. داشتن god class (کلاسی که تعداد خیلی زیادی از object ها در آن هیچ ارتباطی با یکدیگر ندارند)
 2. مفهوم message chain (پیغامی که بین object های یک سیستم ارسال میشود تا نهایتاً یک عمل انجام شود) با این کار در سیستم ما dependency زیاد شده و باعث بهم ریختن روند maintaining میشود.
- تنها راه مقابله با این اتفاق ، شکستن این کلاس ها به کلاس های کوچکتر است که object های اون کلاس با هم ارتباط داشته باشند.

COMPONENT DIAGRAM:

مؤلفه های یک سیستم نرم افزاری را component میگویند.

در این diagram ما component های یک سیستم را به همراه رابطه بین آنها را تعیین میکنیم. این diagram به طور کلی مربوط به solution domain است.

ما از این diagram تنها و تنها در بخش design یا طراحی در methodology UP است.

در این diagram تنها ارتباط بین component ها interface است و خود مؤلفه ها با هم ارتباط مستقیم ندارند.

به همین دلیل داخل کد های یک component توسط دیگر component ها قابل دیدن نیست.

همه موارد بالا یکی از دلایل این موضوع است که در صورتی که اگر سیستم ها بر اساس این diagram ساخته شوند، باعث افزایش توانایی نگه داری میشود و سیستم ها در این موارد برای ایجاد یک feature جدید ، بیشترین زحمت را ایجاد میشود اما نگه داری از این سیستم بسیار بیشتر است.

در واقع این سیستم ها ویژگی های pluggability و extensivity را توسط component در خود ایجاد میکنند.

Interface in component diagram

در یک interface :

دو مؤلفه داریم :

این component مورد نیاز است required

این component که به یک component دیگر ارائه میکند provider

به این نوع از اتصالات گوی و کاسه ای میگویند.

این از ویژگی های component diagram است.

از component diagram در سیستم های استفاده میشود که نحوه توسعه آنها CBD (component based development) است.

در UP methodology کاملاً از این diagram ساقط می کنند.