



اصول و قواعد

نکاتی که باید در طراحی نمودارهای خود به آن‌ها توجه کنید:

- نام کلاس‌های شما باید معنا دار باشد
- تمام فرآیندها، مسیرها و روابط باید مشخص شود
- متودها و صفات هر کلاس باید مشخص شود

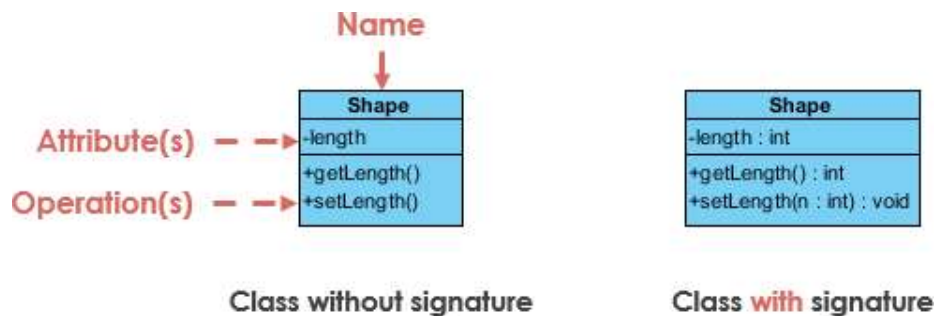
عناصر مهم در نمودار کلاس:

۱. نام کلاس

۲. صفات

۳. اعمال

۴. روابط



♦ نام کلاس:

۱. با حرف بزرگ شروع شود
۲. پررنگ نوشته شود
۳. در وسط نوشته شود (تراز در وسط)
۴. اگر کلاس **abstract** است به صورت *italic* نوشته شود

**♦ صفات:**

در این بخش باید صفاتی که این کلاس مدل می‌کند را اضافه کنید مثلاً برای یک دانشجو می‌توان مدل‌سازی زیر را قائل شد:



باید برای هر صفت دسترسی آن نیز مشخص شود که به صورت زیر است:

- Public (دسترسی در همه جا): +
- Private (دسترسی درون کلاسی): -
- Protected (دسترسی در کلاس های فرزند): #
- Package (درون پکیج): ~

نکات:

- صفات باید نام مناسب داشته باشند.
- برای صفات باید نوع داده آن‌ها را مشخص کنید؛ مثلاً: String، int و

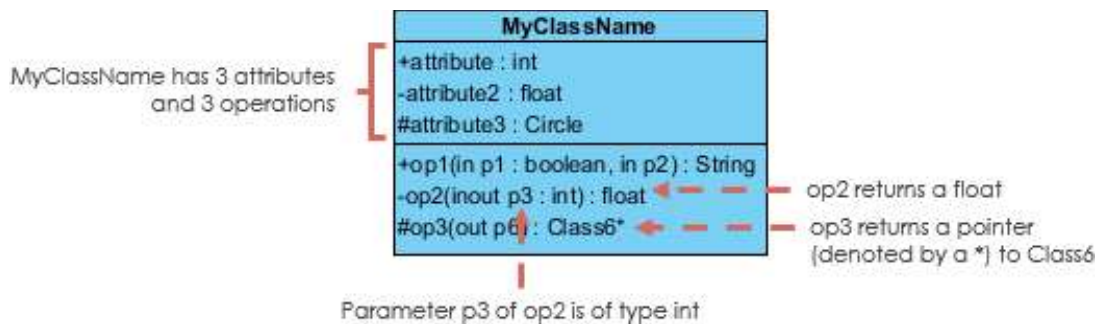


♦ اعمال :

در این قسمت باید تابع های هر کلاس را اضافه کنید و باید توجه داشته باشید که:

- ابتدا باید نوع دسترسی تابع را مشخص کنید
 - نام معناداری انتخاب کنید
 - پارامترهای ورودی را همراه نوع داده‌ی آن مشخص کنید
 - نوع داده‌ای که تابع بر می‌گرداند (return type) را در انتهای خط پس از علامت : مشخص کنید
- مثلا :

+ method (param : int) : String

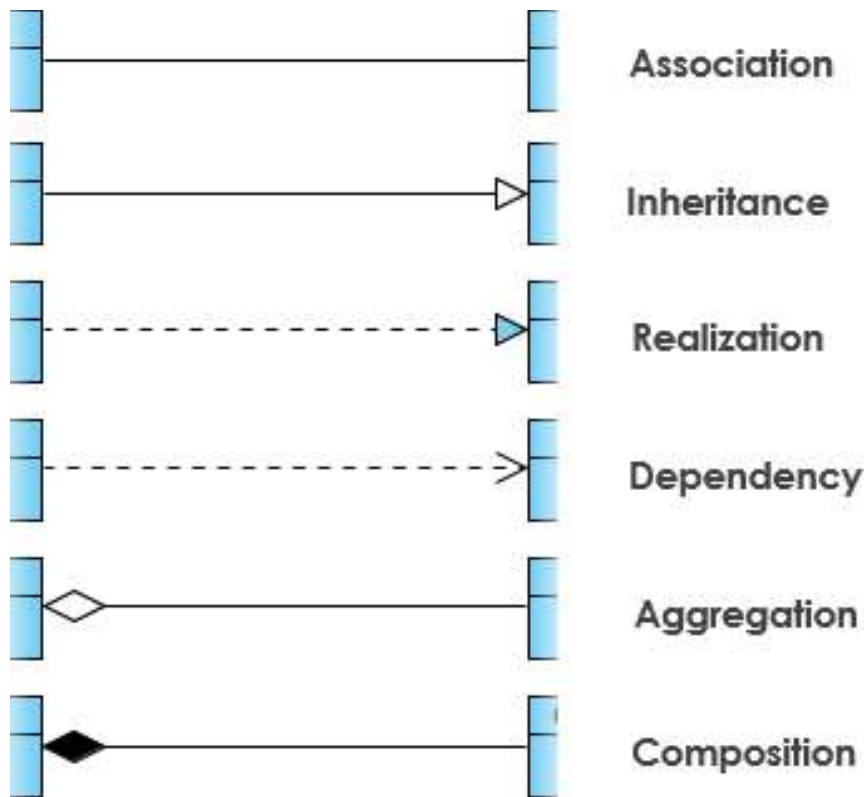


به in و inout در شکل توجه نکنید



♦ روابط:

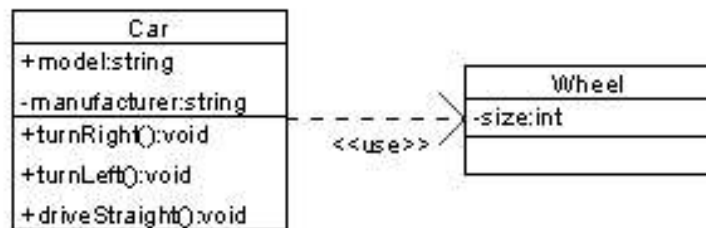
- وابستگی
- تعمیم و وراثت
- انجمنی
- تفهیم





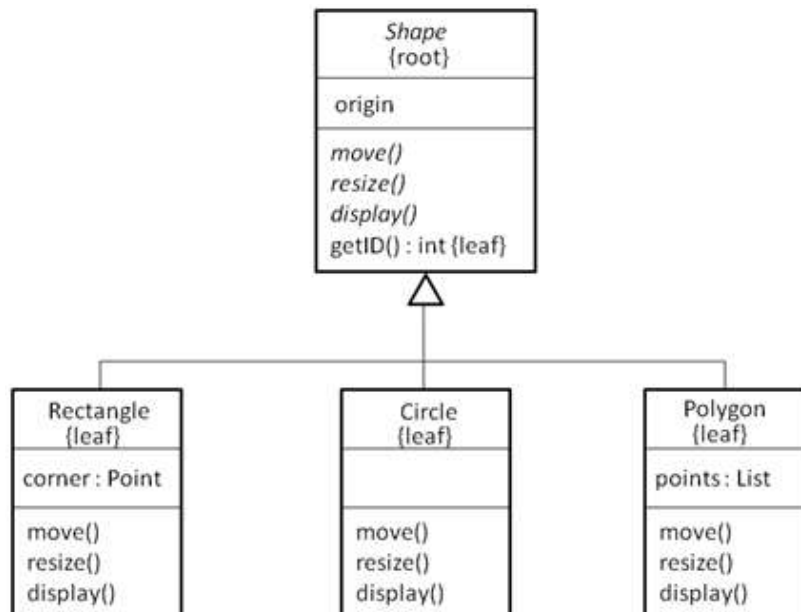
■ وابستگی (Dependency):

وابستگی به معنای رابطه بین دو یا تعداد بیشتری کلاس است که ممکن است تغییر در یک کلاس، تغییر در کلاس دیگر را ایجاد کند. همان طور که از اسم این رابطه مشخص است به این معناست که یک کلاس به دیگری وابسته است.



■ تعمیم و وراثت (Generalization and Inheritance):

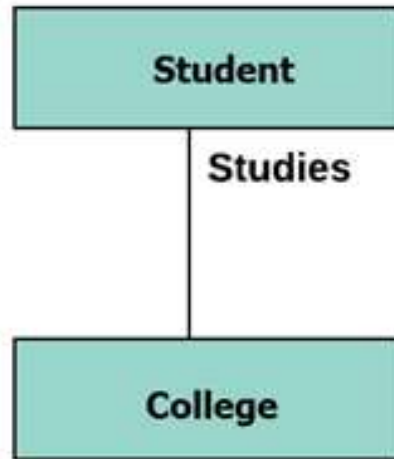
این رابطه کلاس فرزند را به پدر مربوط می‌کند؛ در واقع کلاس فرزند از پدر ارث بری می‌کند. توجه کنید ازین رابطه برای interface ها نباید استفاده کرد!!!





■ انجمنی (Association):

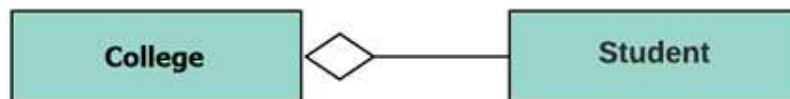
برای نمایش روابط ایستا بکار می‌رود مثلاً: کارمند کار می‌کند برای کارخانه و یا:



انواع روابط انجمنی:

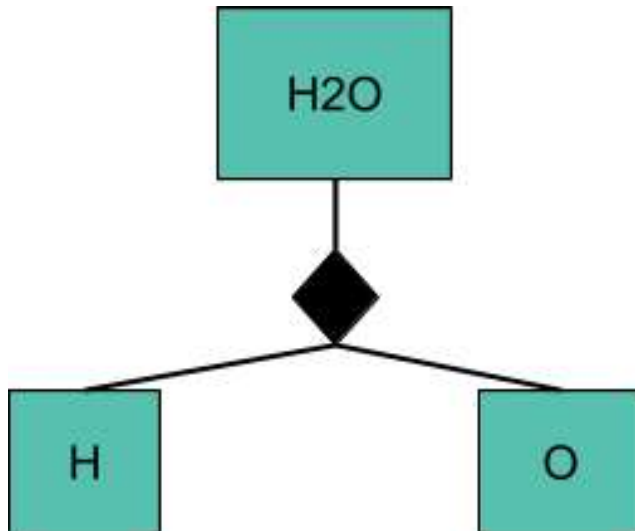
۱. تجمع (Aggregation):

این رابطه یک نوع خاص از روابط انجمنی است که رابطه بین کل و اجزای آن را مدل می‌کند. توجه کنید در این رابطه کلاس‌ها به طور کامل به هم وابسته نیستند؛ مثلاً در رابطه‌ی زیر کلاس دانشگاه حتی اگر دانشجو نیز نباشد باقی خواهد ماند.



۲. ترکیب (Composition):

این رابطه نوع قوی‌تر رابط‌هی قبلی است به این صورت که دو یا چند کلاس کاملاً به هم وابسته‌اند؛ مثلاً:



در اینجا اگر O یا H نباشد کلاس H2O نمی‌تواند باشد.

۳. نرمال (Normal):

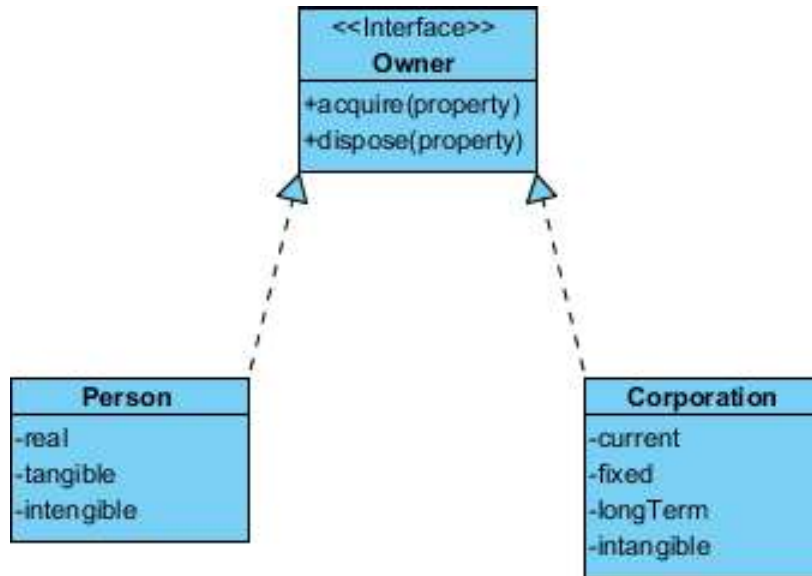
رابط‌هی عادی بین دو کلاس مثلاً:





■ تفهیم (Realization):

این رابطه به نام تفهیم یا تحقق نام دارد و کلاس‌های دیگر وظیفه‌ی تکمیل کردن دارند؛ مثلاً برای interface و enum ها بکار می‌رود.

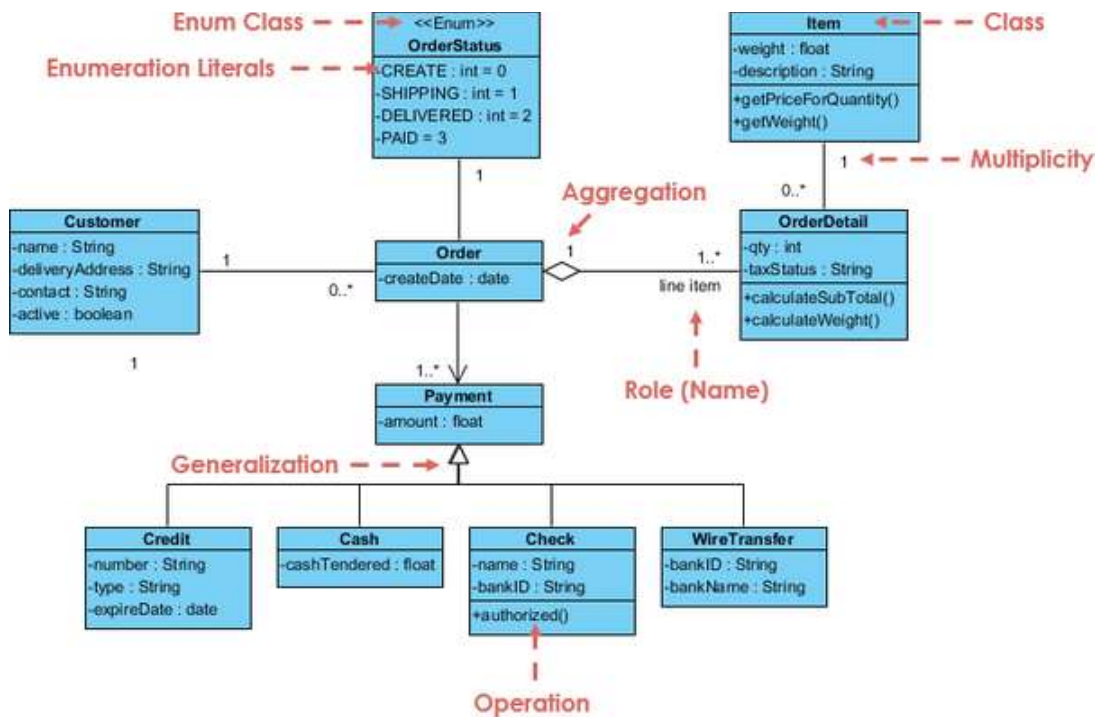


◆ نکات:

۱. برای نمایش static از زیر خط (underline) استفاده می‌کنیم.
۲. چندی (Multiplicity) را نیز می‌توانید با نوشتن عدد یک (نماینده یک نمونه) و یا * (نماینده چند نمونه) روی خط رابطه نمایش دهید. (مانند نمونه)



◆ نمونه:





نرم افزارها و سایت های رسم UML

۱. [سایت Lucidchart](#)
۲. [نرم افزار Microsoft Visio](#)
۳. [نرم افزار UMLet](#)
۴. [نرم افزار Visual Paradigm](#)
۵. [نرم افزار Modelio](#)
۶. [سایت Draw.io](#)