

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

نشان گذاری های پایه در UML

مدرس : مهندس افشین رفوآ

دوره آموزش UML

نشان گذاری های پایه در UML (Notation)

یکی از دلایل محبوبیت UML، در نشان گذاری های نموداری (diagrammatic notation) آن نهفته است. همه ی ما می دانیم که UML یک زبان مدل سازی یکپارچه برای نمایش و به تصویر کشیدن، مشخص کردن، ساختن و مستندسازی اجزا یا مولفه های سیستم های نرم افزاری و غیر نرم افزاری می باشد. گفتنی است که در اینجا آنچه از همه مهمتر است، فراگیری و درک Visualization (نمایش تصویری) می باشد.

نشان گذاری UML مهمترین عنصر در مدل سازی به حساب می آید. استفاده ی مناسب و کارآمد از نشان گذاری در ساختن یک مدل کامل و معنی دار از اهمیت خاصی برخوردار است. یک مدل، اگر در نمایش دادن و شرح مقصود نهایی خود موفق نباشد، کاملاً ناکارآمد و بیپوده خواهد بود.

از این رو، یادگیری نشان گذاری (notation) می بایست از همان ابتدای امر مورد تأکید قرار گیرد. برای اشیا (things) و رابطه ها (relationship)، طبیعتاً از نشان گذاری های متفاوت بهره گرفته می شود و نمودارهای UML نیز در نهایت از همین نشان گذاری های اشیا و رابطه ها ایجاد می شوند. توسعه پذیری یکی دیگر از ویژگی ها و قابلیت های مهم می باشد که UML را قدرتمند و انعطاف پذیر می سازد.

فصل حاضر نشان گذاری های کلی UML را به تفصیل شرح می دهد. این بخش در واقع افزونه ای برای مبحث قبلی می باشد، بنابراین برای یادگیری آن می بایست درس پیشین را فراگرفته باشید.

اشیا ساختاری (Structural things)

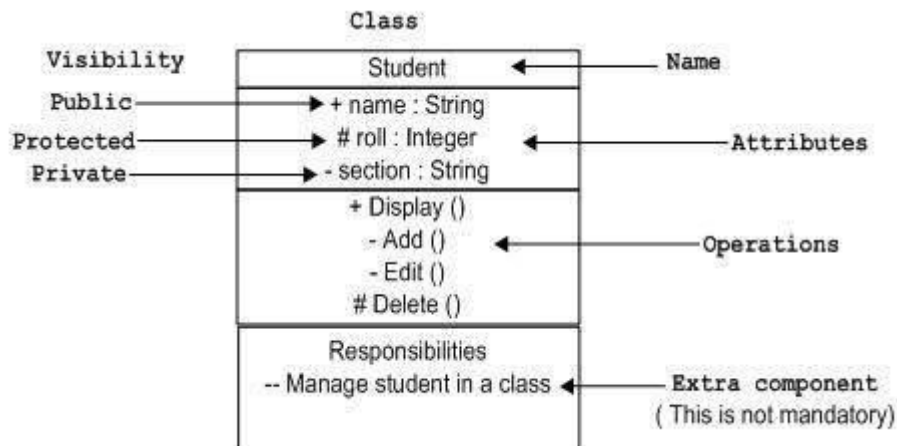
نشان گذاری های ترسیمی (**graphical notations**) پرکاربردترین در **UML** می باشد. اینها به نوعی نقش اسم را در مدل های **UML** ایفا می کنند. در زیر فهرستی از اشیا ساختاری را مشاهده می کنید:

1. کلاس ها
2. شی
3. **interface** (رابط)
4. **collaboration**
5. **Use case**
6. **active class** ها
7. **Component** ها (مولفه ها)
8. **node** ها (گره ها)

نشان گذاری کلاس

نمودار زیر، کلاس **UML** را به نمایش می گذارد. نمودار حاضر به چهار بخش تقسیم شده:

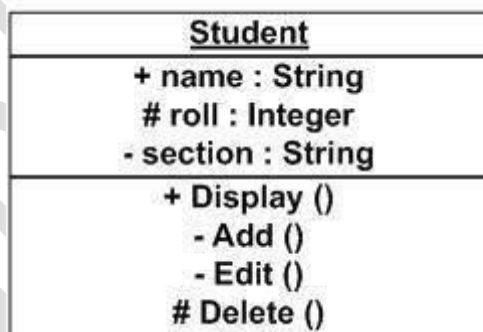
1. بالاترین بخش نام کلاس را مشخص می کند.
2. دومین بخش خصیصه ها یا **attribute** های کلاس را نمایش می دهد.
3. بخش سوم عملیاتی که کلاس مورد نظر قادر به انجام آن ها است را توصیف می کند.
4. آخرین بخش نیز اختیاری بوده و جهت نشان دادن مولفه های اضافی بکار می برد.



کلاس ها برای نمایش اشیا بکار می روند. اشیا می توانند هر چیزی که دارای رفتار و خاصیت هایی (property) (ها) هستند، باشند.

نشان گذاری شی (object notation)

شی نیز همانند کلاس نمایش داده می شود. تنها تفاوت، همان طور که در تصویر زیر می بینید، در اسم آن هاست.

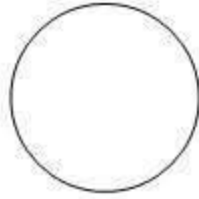


به این خاطر که شی در واقع پیاده سازی عملی یک کلاس هست، که به عنوان نمونه ای از آن کلاس شناخته می شود، مورد استفاده ی آن با کلاس یکسان می باشد.

نشان گذاری رابط (interface notation)

همان طور که در تصویر زیر مشاهده می کنید، **interface** توسط یک دایره نمایش داده می شود. این نمودار دارای یک اسم می باشد که در زیر آن درج می گردد.

Interface



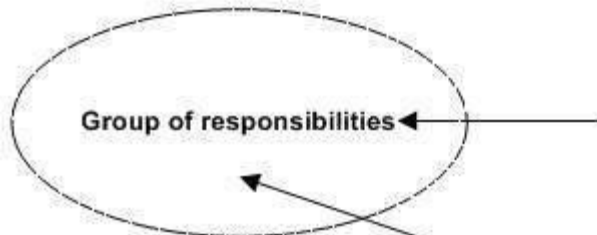
StudentApplication ← Name

مورد استفاده ی رابط (interface)، توصیف قابلیت بدون پیاده سازی می باشد. رابط دقیقا شبیه به یک قالب (template) می باشد که در آن کاربردهای مختلف تعریف می شود، اما در آن از پیاده سازی (implementation) خبری نیست. هنگامی که کلاس رابط را پیاده سازی می کند، همراه با آن و با توجه به نیاز قابلیت ها را نیز implement می کند.

Collaboration Notation (نشان گذاری همکاری)

Collaboration همان طور که در تصویر زیر نظاره گر آن هستید، توسط یک **ellipsis** نقطه چین (شکل بیضی نقطه چین) نمایش داده می شود. در داخل این شکل بیضی مجموعه مسئولیت ها درج می گردند.

Collaboration

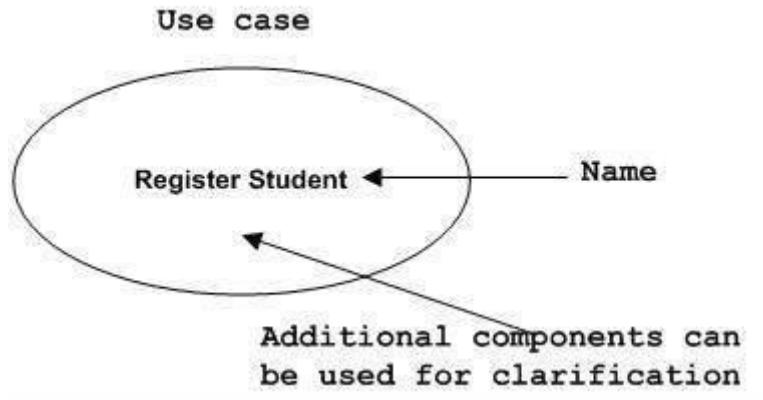


Additional components can be used for clarification

نمودار **collaboration** (همکاری) این مسئولیت ها را به نمایش می گذارد. معمولا مسئولیت ها زیر مجموعه ی یک گروه قرار می گیرند.

نشان گذاری Use case

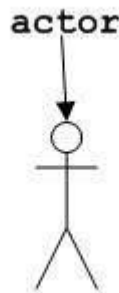
Use case به صورت یک شکل بیضی ترسیم می شود که اسمی داخل آن درج شده. در این نمودار همچنین ممکن است مسئولیت های اضافی بر سازمان جهت روشن سازی نوشته شود.



Use case در اصل به منظور نمایش دادن (و شناسایی) قابلیت ها و کاربردهای سطح بالای یک سیستم بکار می رود.

Actor Notation

می توان گفت که **actor** یک موجودیت داخلی یا خارجی است که با سیستم تعامل می کند.

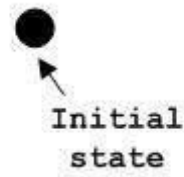


Actor در یک نمودار **Use case** جهت توصیف موجودیت های داخلی و خارجی مورد استفاده قرار می گیرد.

نشان گذاری Initial State

Initial State در واقع وضعیت اولیه یا شروع یک فرآیند را نمایش داده و تعریف می نماید. این نشان گذاری در

تقریباً تمامی نمودارها بکار می رود.



همان طور که از اسم آن پیدا است، کاربرد این نشان گذاری در مشخص کردن آغاز یک پروسه یا شروع فرایند خلاصه می شود.

نشان گذاری Final State

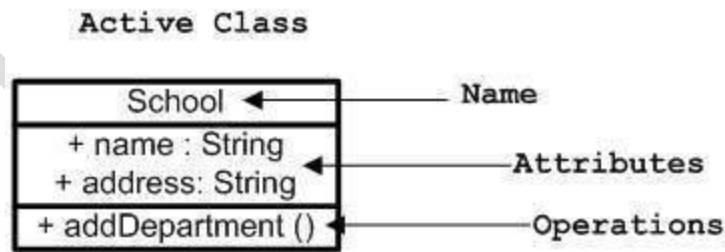
Final State همان طور که از اسم آن می تواند استنباط کرد، پایان یک پروسه یا فرایند را توصیف می کند. این نشان گذاری تقریباً در تمامی نمودارها برای مشخص کردن نقطه ی پایان پروسه بکار می رود.



بنابراین کاربرد **final state notation** در نمایش پایان فرایند خلاصه می شود.

نشان گذاری Active class

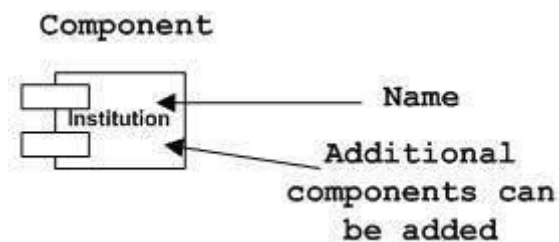
Active class بسیار شبیه به یک کلاس است که خط حاشیه ی ممتد آن را محصور می کند. این نوع نشان گذاری به طور معمول جهت به نمایش گذاشتن رفتار و عملکرد همروند (**concurrent behavior**) سیستم بکار می رود.



بنابراین مورد استفاده ی **Active class** در نمایش همروندی سیستم نهفته است.

نشان گذاری Component

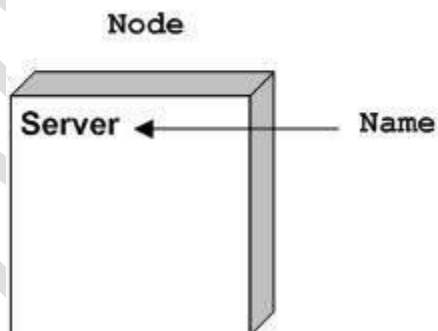
Component یا جز و یا مولفه در UML به صورت زیر و با یک اسم درج شده در داخل آن نمایش داده می شود. در صورت نیاز، می توان المان های اضافی بر سازمان را در هر جایی که لازم بود اضافه نمود.



Component به منظور نمایش دادن بخش های سیستم که نمودار های UML برای آن ترسیم و ایجاد می شود، بکار می رود.

Node notation

در UML، گره یا **Node** را به همراه اسم آن با یک جعبه ی مربع شکل نمایش می دهیم. یک node نشانگر جز فیزیکی (**physical component**) سیستم می باشد.



بنابراین با **node** می توان مولفه های فیزیکی سیستم همچون **server**، **network** و غیره را نمایش داد.

Behavioral things (بخش های پویا و دینامیک مدل های UML)

بخش های دینامیک یکی از مهمترین المان ها در UML هستند. UML با قابلیت ها و امکانات قدرتمندی که دارد، قادر است به آسانی بخش های پویای یک سیستم نرم افزاری یا غیر نرم افزاری را نمایش دهد. این قابلیت ها شامل **interaction** و **state machine** می باشد.

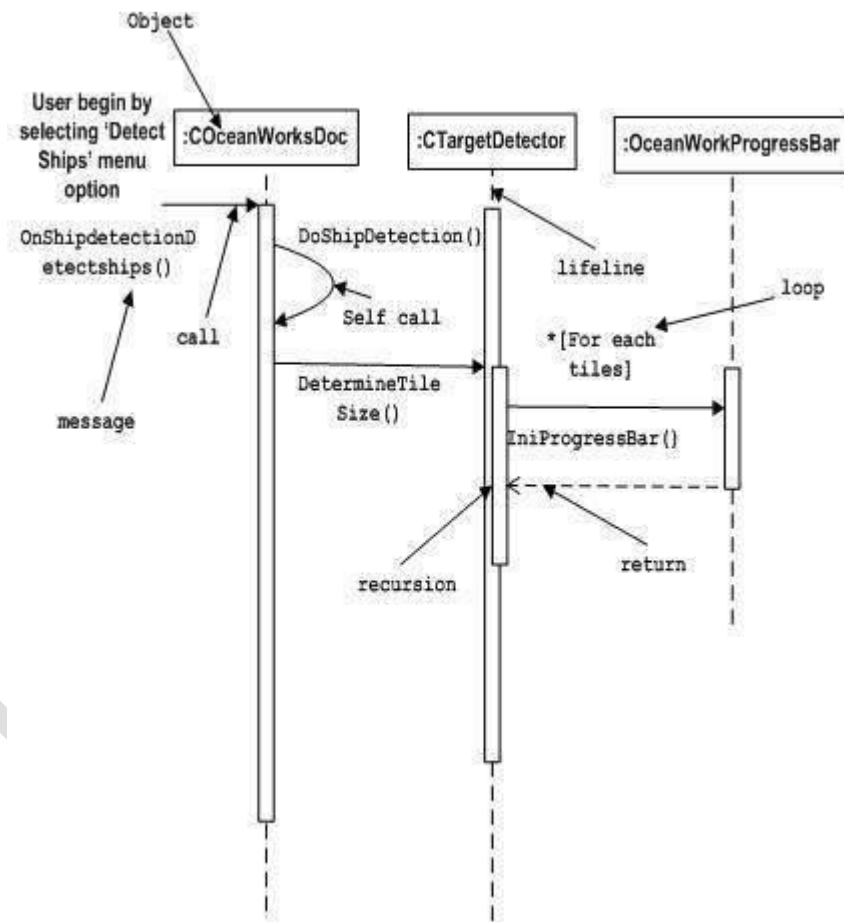
Interaction خود به دو نوع تقسیم می شود:

1. **Sequential** یا ترتیبی (توسط نمودار توالی **sequence** نمایش داده می شود)

2. **collaborative** یا مشترک (توسط نمودار همکاری **collaboration** به نمایش گذاشته می شود)

Interaction notation

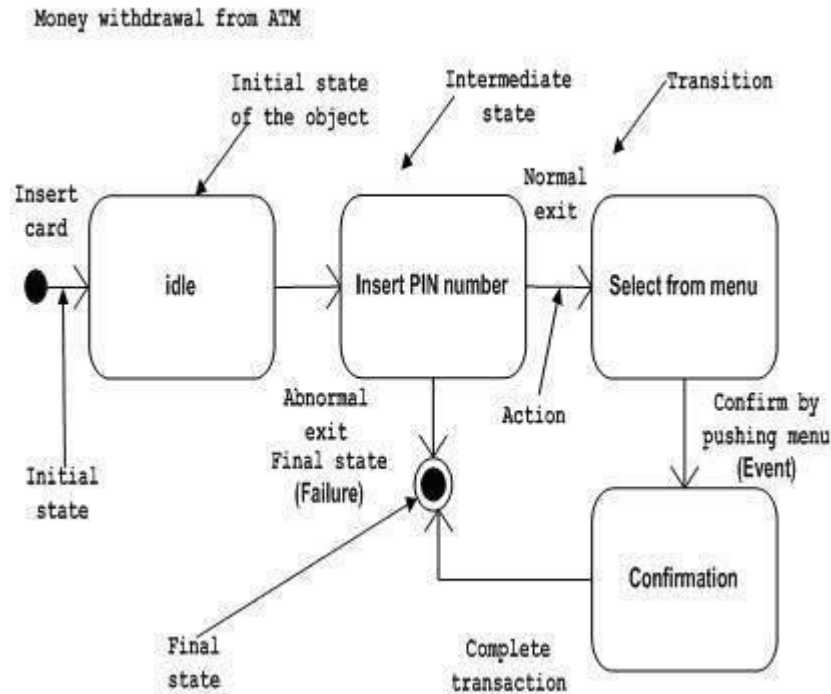
Interaction در واقع همان تبادل پیام بین دو جز یا مولفه ی **UML** می باشد. نمودار زیر نشان گذاری های مختلف که در یک **interaction** شرکت داشته و بکار می رود را نمایش می دهد.



Interaction جهت نمایش ارتباط و تعامل بین اجزا مختلف سیستم استفاده می شود.

State machine Notation

State machine به منظور شرح وضعیت های مختلف یک **component** در چرخه ی حیات (**life cycle**) آن بکار می رود. **Notation** ها در نمودار زیر نشان و شرح داده شده اند:



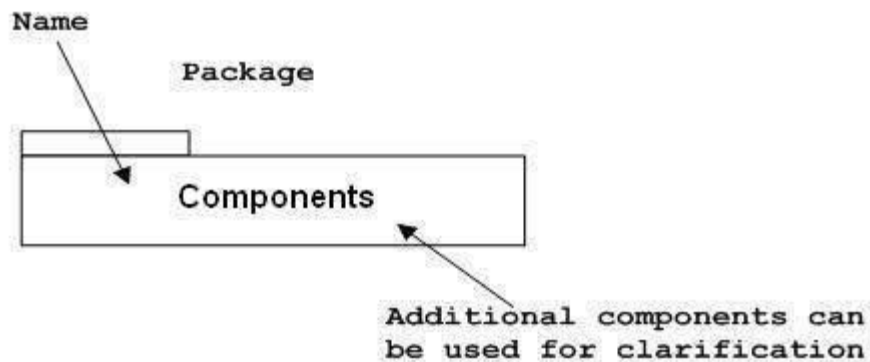
State machine وضعیت های مختلف سیستم و انتقال بین وضعیت ها را نمایش می دهد؛ به عبارتی دیگر ماشین وضعیت برای توصیف وضعیت های مختلف **component** یک سیستم بکار می رود. وضعیت یا **state** می تواند **Active**، **idle** یا بسته به شرایط هر چیز دیگری باشد و بین این ها تغییر کند.

Grouping things (المان های گروه بندی)

سازمان دهی مدل های **UML**، یکی از مهم ترین جنبه های طراحی (**design**) محسوب می شود. در **UML** تنها یک المان ویژه ی گروه بندی وجود دارد و آن هم **Package** می باشد.

package annotation

نشان گذاری **package** در زیر به تصویر کشیده شده است. این گونه نشان گذاری برای دربرگیری دیگر اجزا و **component** های سیستم کاربرد دارد.

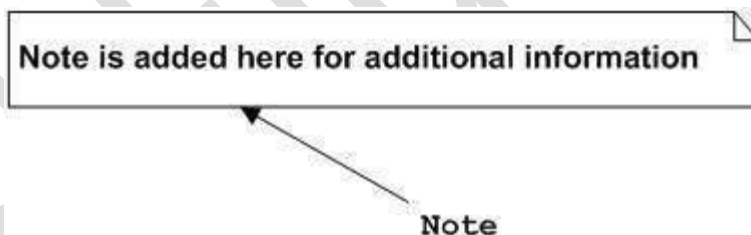


Annotational things (المان های حاشیه نویسی)

در هر نموداری، آنچه بالاترین اهمیت را به خود اختصاص می دهد، تشریح المان ها و قابلیت های آن ها می باشد. **UML** برای این منظور **notes notation** را ارائه می دهد.

Note Notation

این نشان گذاری را می توانید در تصویر زیر مشاهده نمایید. با استفاده از این نشان گذاری می توان تمامی اطلاعات مورد نیاز یک سیستم را فراهم نمود.



Relationships (رابطه ها)

یک مدل کامل قلمداد نمی شود مگر اینکه روابط بین المان های آن به درستی شرح داده شود. این **Relationship** یا رابطه است که معنی یک مدل **UML** را تکمیل می کند. در زیر انواع روابط موجود در **UML** نام برده و شرح داده شده است:

1. **Dependency** (وابستگی)

2. **Association** (انجمنی)

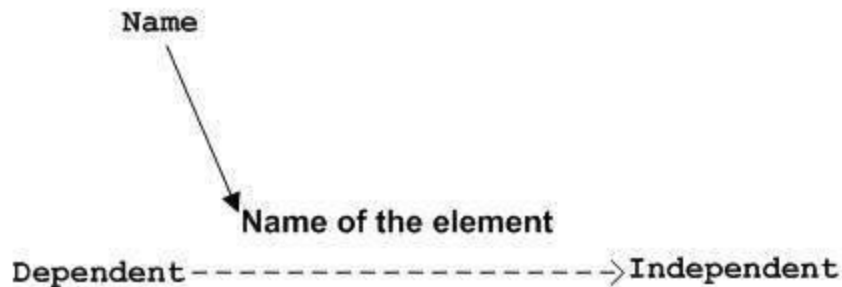
3. **Generalization** (رابطه ی وراثت)

4. **Extensibility** (توسعه پذیری)

Dependency notation

Dependency یکی از جنبه های مهم در المان های **UML** می باشد. این نشان گذاری المان های وابسته را و جهت وابستگی را مشخص می کند.

در تصویر زیر، **Dependency** به وسیله ی یک پیکان نقطه چین مانند نمایش داده شده است. نوک پیکان به المان مستقل اشاره دارد و طرف دیگر آن عنصر وابسته را مشخص می کند.



با توجه به آنچه گفته شد، رابطه ی **Dependency**، وابستگی بین دو المان سیستم را نشان می دهد.

Association notation

Association چگونگی ارتباط بین المان های یک نمودار **UML** را شرح می دهد. به عبارت ساده تر،

Association مشخص می کند چه تعداد المان در یک تعامل (**interaction**) شرکت دارند.

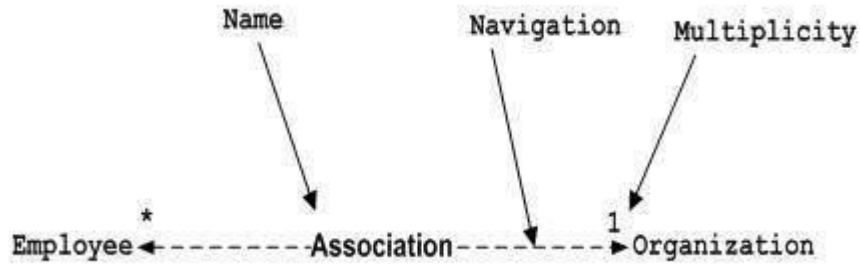
Association توسط خط نقطه چین با یا بدون پیکان در دو طرف خط نمایش داده می شود. دو طرف خط نشانگر

دو المان متصل به هم می باشد. یکی از association های مهم که در تحلیل سیستم وجود دارد تناظر

چندتایی یا **Multiplicity** می باشد : در واقع **multiplicity** تعداد اشیای مرتبط از یک کلاس را با اشیا کلاس

دیگر بیان می کند.

Multiplicity را در تصویر زیر مشاهده می کنید که نشان می دهد چه تعداد شی به هم مرتبط هستند.



با توجه به آنچه گفته شد، **Association** رابطه بین دو المان را در یک سیستم نمایش می دهد.

Generalization Notation

Generalization رابطه ی وراثت در دنیای شی گرا را تشریح می کند یا به عبارتی دیگر بیانگر رابطه ی پدر و فرزندی بین اشیا می باشد.

Generalization در نمودار به صورت یک خط که در سر آن یک پیکان تو خالی قرار می گیرد، نمایش داده می شود. در یک طرف عنصر پدر و در طرف دیگر این خط ممتد المان فرزند نمایش داده می شود.



با استناد به آنچه در بالا گفته شد، **Generalization** رابطه ی پدر-فرزندی بین دو المان در یک سیستم را توصیف می کند.

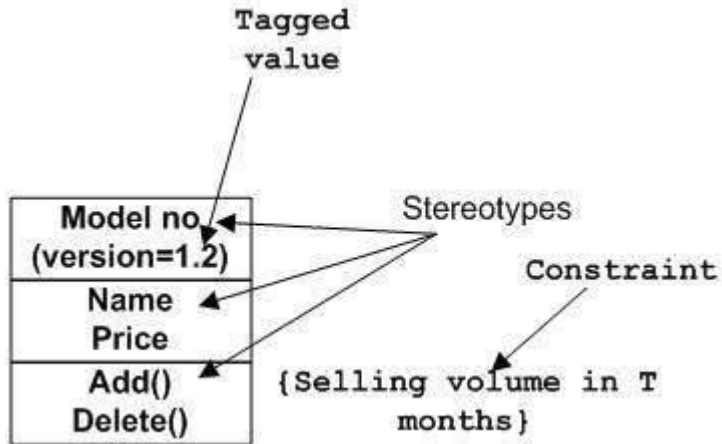
Extensibility Notation

تمامی زبان ها (برنامه نویسی یا مدل سازی) دارای مکانیزم هایی هستند که امکان بسط و توسعه ی قابلیت های آن را فراهم می آورد، از جمله می توان به **syntax**، **semantics** اشاره کرد. **UML** مکانیزم هایی دارد که قابلیت توسعه پذیری به آن می دهد. این سازوکارها عبارتند از:

1. **Stereotype** ها یا کلیشه ها (بیانگر المان های جدید هستند)

2. **tagged values** (نشانه گر خصیصه های جدید می باشد)

3. **Constraint** یا محدودیت (نشان دهنده ی قیود می باشد)



Extensibility notation زمینه ی بسط و افزایش قدرت زبان را مهیا می سازد. **Extensibility notation** ها عملاً المان های اضافی بر سازمان یا الحاقی هستند که تعدادی رفتار و عملکرد اضافی سیستم را معرفی می کند. این رفتارهای اضافه بر سازمان توسط **notation** های متعارف موجود تحت پوشش قرار نمی گیرد.