

فصل اول

زبان مدلسازی یکپارچه

هادی سلیمی

دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران

hsalimi@iust.ac.ir

مهندسی نرم افزار ۲- زبان مدلسازی یکپارچه

تاریخچه

- پیدایش زبانهای برنامه سازی شیء گرا از اواسط دهه هفتاد
- رسیدن روش های تولید شیء گرا تا سال 95 به پنجاه عدد
- استاندارد شدن زبان UML توسط کنسرسیوم OMG و شرکت Rational (امروزه با نام IBM Rational شناخته می شود).
- ترکیبی از بهترین روش های مدلسازی موجود

سه جزء اساسی زبان UML

- نمودارها (Diagrams)
- ساختاری (Structural)
- رفتاری (Behavioral)
- گروه بندی (Grouping)
- حاشیه نویسی ها (Annotationals)

نمودار کلاسی (Class Diagram)

- رایج ترین نمودار زبان UML است.
- مجموعه ای از کلاس ها، واسط ها و ... است که توسط نوع خاصی از روابط به هم متصل گشته اند.
- دیدگاهی ایستا از سیستم را نمایش می دهد.

اجزای کلاس

نام	Account	
ویژگی ها	balance: Real = 0	مقدار پیش فرض
عملگرها	<pre><<constructor>> +addAccount() <<process>> +setBalance(a : Account) +getBalance(a: Account): Amount ... <<query>> isValid(loginID : String): Boolean</pre>	شکل توابع سه نقطه برای نمایش عملگرها و یا ویژگی های دیگر کلشه

نام گذاری کلاس

نام کلاس می تواند شامل یک توالی از فضاها (namespace) نام تا رسیدن به آن کلاس هم باشد. مانند:

House::Appliance::WashingMachine

ویژگی های کلاس

- ویژگی های یک کلاس حالت (Status) اشیای از آن نوع را مشخص می کنند.

WashingMachine
brandName modelName SerialNumber capacity

شیء (Object)

- یک شیء را هم می توان به عنوان یک نمونه از یک کلاس به شکل مقابل نشان داد.

<u>myWasher: WashingMachine</u>
brandName = "Samsung" modelName = "Tina" SerialNumber = "GL5774" capacity = 16

مقادیر پیش فرض

- می توان برای برخی از ویژگی های کلاس مقدار پیش فرض تعریف کرد.

WashingMachine

brandName : String = "Samsung"

modelName : String

SerialNumber : String

capacity : Integer

نمایش کلاس

- در عمل می توان برخی از ویژگی ها و عملگرهای کلاس (و یا کل ان را) نمایش نداد.

WashingMachine

نمایش جزئی اجزای کلاس

- با علامت ... می توان مشخص کرد که کلیه ویژگی ها و یا عملگرهای کلاس نمایش داده نشده است.

WashingMachine

brandName

...

addClothes()

...

کلیشه ای کردن

- توسط کلیشه ها می توان اطلاعات بیشتری به کلاس اضافه کرد.

WashingMachine

<<id info>>

brandName

<<machine info>>

capacity

<<clothes-related>>

addClothes()

<<machine-related>>

turnOn()

محدود کننده ها (Constraints)

توسط محدود کننده ها می توان برخی از محدودیت های ویژگی ها و یا عملگرها را نمایش داد.

WashingMachine

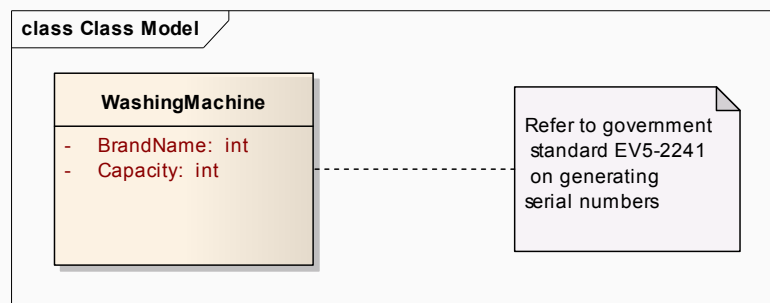
brandName
modelName
SerialNumber
capacity

addClothes()

{Capacity = 16
or 18 or 20lb }

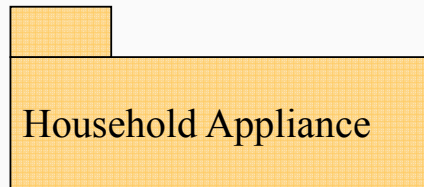
توضیحات

- به کمک نماد زیر می توان توضیحات اضافه تری به اجزای مدل اضافه کرد.



بسته ها (Package)

- از بسته ها می توان به منظور دسته بندی کلاس ها و سایر اجزای UML استفاده کرد.

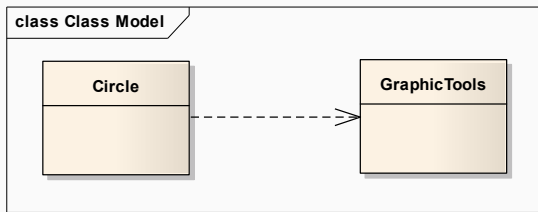


روابط

- به منظور مدلسازی ارتباط کلاس ها و سایر اجزای UML به همدیگر روابط مختلفی تعریف شده است.
- هر رابطه معنی خاص خود را داشته و در نمودارهای مختص خود می تواند کاربرد داشته باشد.

رابطه وابستگی (Dependency)

- ساده ترین شکل وابستگی بین دو کلاس را نمایش می دهد.



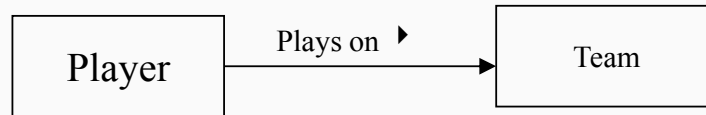
- به عنوان نمونه شکل مقابل یک وابستگی ساده بین دو کلاس را نشان می دهد.

رابطه وابستگی

- این نوع رابطه مشخص می کند که یک تغییر در کلاس اصلی (GraphicTool) ممکن است به هر شکلی تعاریف مرتبط با کلاس وابسته (Circle) را دستخوش تغییر کند.
- مثلاً در شکل قبلی ممکن است کلاس Circle دارای متدی باشد که جنس یکی از پارامترهای آن GraphicTool است.

رابطه تناظر (Association)

- این نوع رابطه مشخص می کند اشیایی از نوع یک کلاس ممکن است به اشیایی از نوع دیگر متصل باشند.



رابطه تناظر

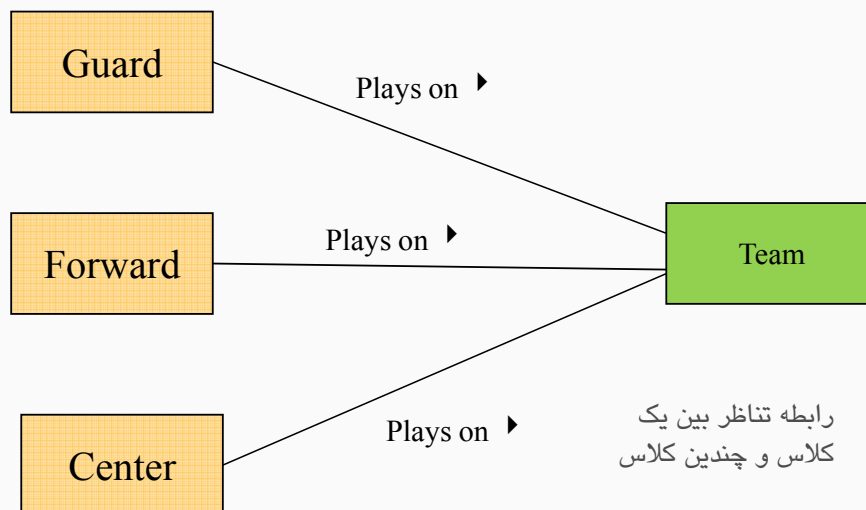
- به عنوان نمونه شکل رابطه بین تیم و بازیکن مشخص می کند که اشیای ساخته شده از جنس تیم و بازیکن به هم متصل هستند.
- می توان گفت که تناظر نوع خاصی از وابستگی است که لزوماً رابطه بین اشیاء از جنس کلاس های دوطرف وجود دارد.

رابطه تناظر

- یک رابطه تناظر می تواند دارای نام و جهت باشد.
- هر کدام از شرکت کنندگان می توانند یک نقش در رابطه داشته باشند.



رابطه تناظر



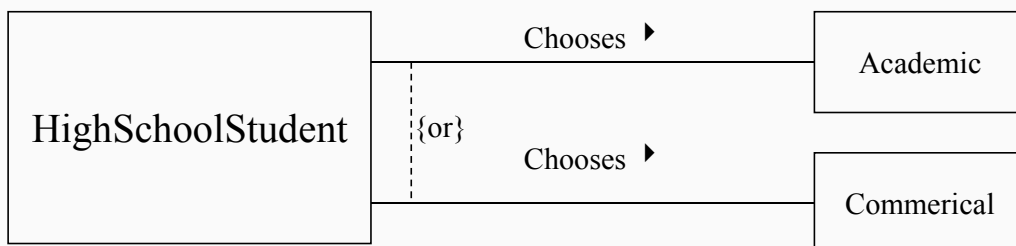
- رابطه تناظر بین یک کلاس و چندین کلاس

محدودیت در روابط



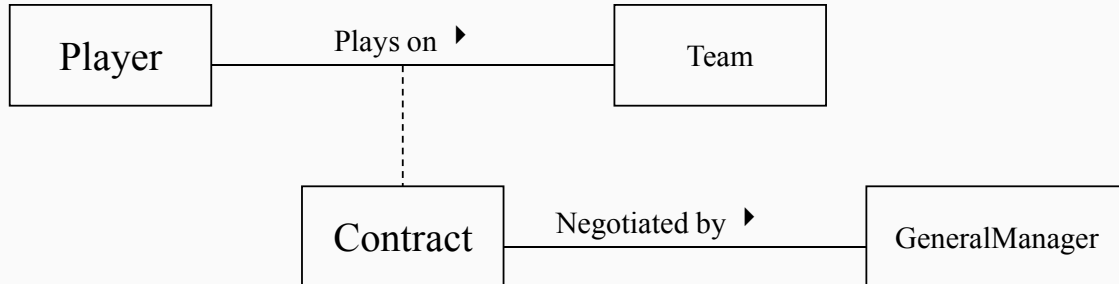
- در روابط نیز می توان محدودیت داشت. در شکل بالا محدودیت `{ordered}` مشخص کرده است که متصدی باجه برای مشتریان ترتیب قایل است.

رابطه تناظر



- گاهی از موارد می توان معین کرد که یک رابطه تناظر با یکی از دو کلاس معرفی شده برقرار باشد.

کلاس تناظر



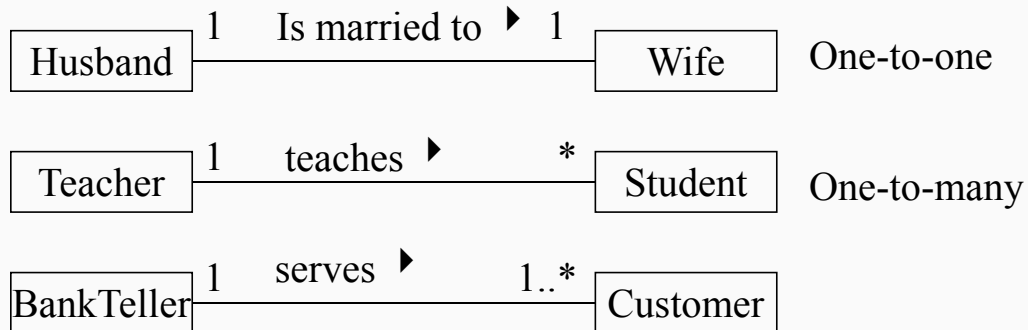
- برای نمایش ویژگی‌های یک رابطه تناظر از کلاس تناظر (Association Class) استفاده می‌کنیم.

اتصال (Link)



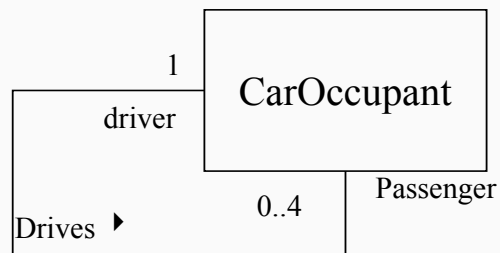
- برخلاف رابطه تناظر که ارتباط بین دو کلاس را نشان می‌دهد، یک اتصال بیانگر رابطه بین دو شیء است.

چندگانگی روابط



- روابط می توانند شامل درجه چندگانگی هم باشند.

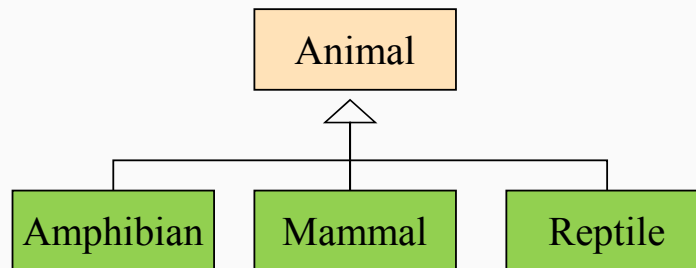
روابط بازتابی



- رابطه تناظر خاصیت بازتابی دارد، یعنی می توان بین یک کلاس و خودش رابطه تناظر برقرار کرد.

رابطه تعمیم (Generalization)

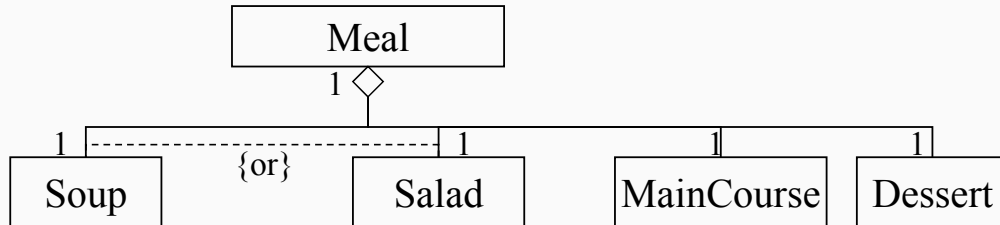
- یک رابطه از نوع تعمیم ارتباطی بین یک چیز و یک چیز خاص از همان نوع است.



رابطه تجمیع (Aggregation)

- نوع خاصی از رابطه تناظر رابطه تجمیع نام دارد.
- در این حالت یک رابطه کل به جزء (Whole/Part) بین طرفین رابطه برقرار است.
- برای نمایش آن از یک علامت الماس (diamond) در سمت کل استفاده می کنیم.

رابطه تجمیع

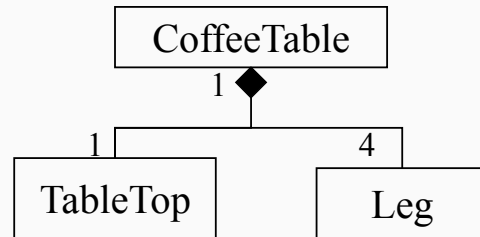


- در شکل فوق رابطه تجمیع یک وعده غذایی با اجزای آن مشخص شده است.

رابطه ترکیب (Composition)

- این رابطه نوع خاصی از رابطه تجمیع است.
- یک رابطه تجمیع یک رابطه ترکیب نیز هست اگر:
 - هر جزء فقط به یک کل تخصیص داشته باشد.
 - چرخه زندگی جزء در اختیار کل باشد.
- به عنوان نمونه رابطه اتاق و دیوار یک رابطه تجمیع است، اما رابطه ترکیب نیست، چون هر دیوار به یک اتاق خاص اختصاص ندارد.

رابطه ترکیب

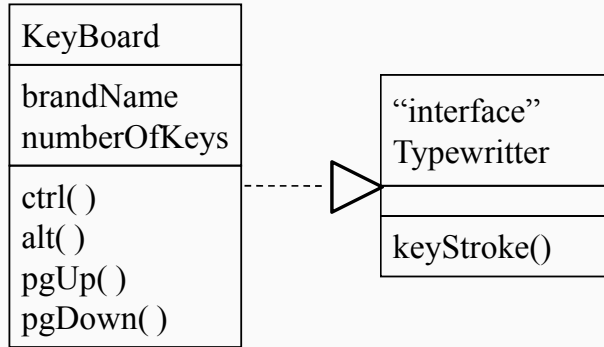


- رابطه ترکیب با یک الماس توپر نمایش داده می شود.

واسط ها (Interfaces)

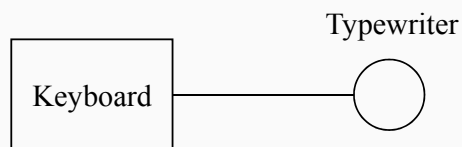
- یک واسط تعریف مجموعه سرویس هایی است که یک چیز (thing) مثلا یک کلاس باید آنها را پیاده سازی کند.
- رابطه بین یک کلاس و واسط مربوطه محقق سازی (Realization) نام دارد.
- شکل رابطه همانند تعمیم است، با این تفاوت که به جای خط ممتد از خط چین استفاده می شود.

رابطه محقق سازی



- محقق سازی یک واسط توسط یک کلاس

نمایش واسط



- واسط ها را می توان به شکل فوق نیز نمایش داد.