

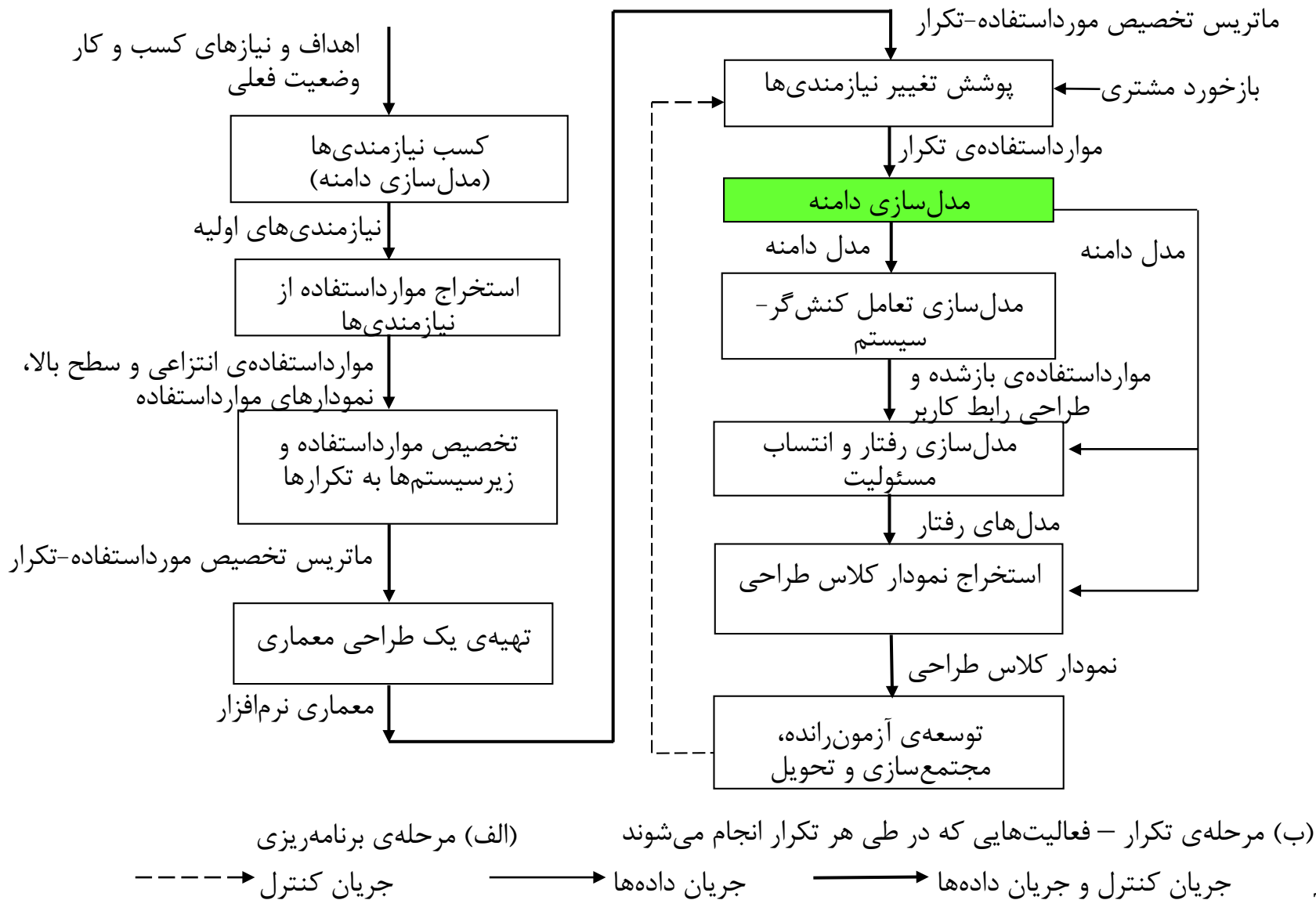
فصل ۵: مدل سازی دامنه

بخش ۱ – نمودار کلاس UML

نکات کلیدی مهم

- مدل سازی دامنه، یک فرایند مفهوم سازی برای کمک به فهم تیم توسعه از دامنه ی کاربرد است.
- پنج گام آسان: جمع آوری اطلاعاتی در رابطه با دامنه ی کاربرد؛ طوفان فکری (brainstorming)؛ رده بندی نتایج طوفان فکری؛ بصری سازی مدل دامنه با استفاده از نمودار کلاس UML؛ و انجام بازرسی و بازبینی.

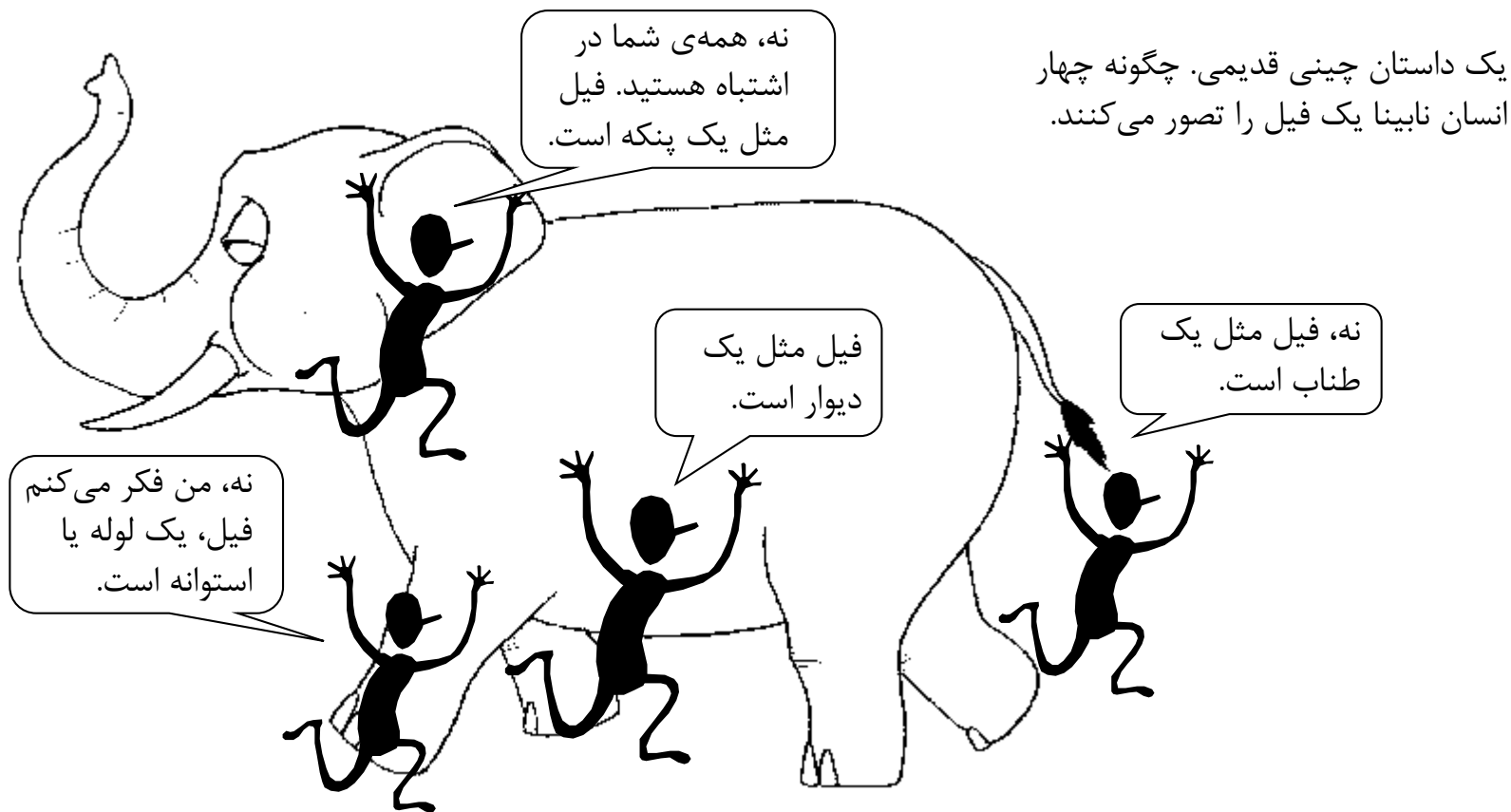
مدل سازی دامنه در متن متدولوژی



مدل چیست؟

- نمایش مفهومی یک چیزی.
- توصیف شماتیک یک سیستم، تئوری یا پدیده‌ای که به‌خاطر معروفیتش یا خواص استنباط شده‌اش شناخته شده و ممکن است برای مطالعه‌ی بیشتر مشخصه‌های آن استفاده شود.
(تعریف لغت‌نامه)

چرا به مدل سازی نیاز داریم؟



درک ما از دنیا به واسطه ی وجود تفاوت هایی در پس زمینه و نقطه ی دید است. مدل سازی، درک جمعی کاربرد را تسهیل می بخشد.

چرا به مدل سازی نیاز داریم؟



به این دلیل که، اعضای تیم و کاربران لازم است تا درک خودشان را در رابطه با یک تکه واقعیت مبادله کنند. مدل به اعضای تیم و کاربران کمک می کند تا درک خودشان را مبادله کرده و ایده ها را به آسانی طراحی نمایند.

چرا به مدل سازی نیاز داریم؟



چون در طی مرحله‌ی نگهداری برای انجام نگهداری و بهبود مراقبت به مدل‌ها نیاز داریم.

مدل سازی دامنه

- چستی:
 - فرایندی است که به تیم کمک می‌کند تا کاربرد یا دامنه‌ی کاربرد را درک کند.
 - به تیم کمک می‌کند تا به یک درک مشترک برسند.
- چرایی:
 - مهندسان نرم‌افزار نیاز دارند تا در دامنه‌های متفاوت یا پروژه‌های متفاوتی کار کنند. آن‌ها به دانش دامنه نیاز دارند تا سیستم را توسعه دهند.
 - مهندسان نرم‌افزار با پس‌زمینه‌های متفاوتی می‌آیند که روی درک آن‌ها از دامنه‌ی کاربرد تأثیر می‌گذارد.
- چگونگی:
 - جمع‌آوری اطلاعات دامنه، انجام طوفان فکری و رده‌بندی و بصری‌سازی دانش دامنه با استفاده از یک نمودار کلاس UML

مدل سازی دامنه

- مدل دامنه، مفاهیم دامنه‌ی کاربرد را با تعابیر کلاس‌ها، صفات و روابط تعریف می‌کند.
- ساخت مدل دامنه
 - به تیم توسعه یا تحلیل‌گر کمک می‌کند تا کاربرد و دامنه‌ی کاربرد را درک کند
 - به اعضای تیم اجازه می‌دهد که درک خودشان از کاربرد و دامنه‌ی کاربرد را به صورت اثربخش مبادله کنند
 - در برخی موارد، ارتباط میان تیم توسعه و مشتری/کاربر را بهبود می‌بخشد
 - پایه‌ای برای طراحی، پیاده‌سازی و نگهداری فراهم می‌کند
- مدل دامنه با نمودارهای کلاس UML (بدون نشان دادن عملیات) نمایش داده می‌شوند

مدل سازی دامنه در سبک OO

- دید سبک OO از دنیای واقعی شامل
 - اشیاء
 - که با همدیگر در ارتباطند و
 - با همدیگر تعامل دارند، می باشد
- بلاک های سازنده ی مبنایی و نقطه ی شروع، اشیاء هستند.

مفاهیم مهم شیء‌گرایی

Class ----- کلاس، یک نوع است

- تجریدی از اشیاء با خواص و رفتار مشابه
- یک تعریف ذهنی از مجموعه‌ای از اشیاء

Attribute ----- صفت، خواص کلاسی از اشیاء را تعریف می‌کند

Operation ----- عملیات، رفتارهای کلاس اشیاء را تعریف می‌کند

Object ----- شیء، نمونه‌ای از یک کلاس

Encapsulation ----- پنهان‌سازی یا کپسوله‌ساختن، تعریف/ذخیره‌ی خواص و رفتار یک کلاس/شیء با همدیگر

مفاهیم مهم شیء‌گرایی

Information hiding پنهان‌سازی اطلاعات، مخفی کردن جزئیات پیاده‌سازی
برای کاهش تأثیر تغییرات بر سایر بخش‌های یک برنامه

Polymorphism چندریختی، یک چیز می‌تواند بیش از یک شکل یا ریخت
داشته باشد

نمایش مدل دامنه با نمودار کلاس UML

- نمودار کلاس UML یک نمودار ساختاری است.
- کلاس‌ها، صفات و عملیات آن‌ها و روابط میان کلاس‌ها در این نمودار نمایش داده می‌شوند.
- مدل دامنه با یک نمودار کلاس، بدون نشان دادن عملیات، نمایش داده می‌شود.

نمودار کلاس UML: مفهوم و نمادگان

Class: a type (in OO)

نام کلاس

نام کلاس

Attributes of class

بخش صفات

Operations of class

بخش عملیات

دید فشرده

دید باز یا گسترده

Example:

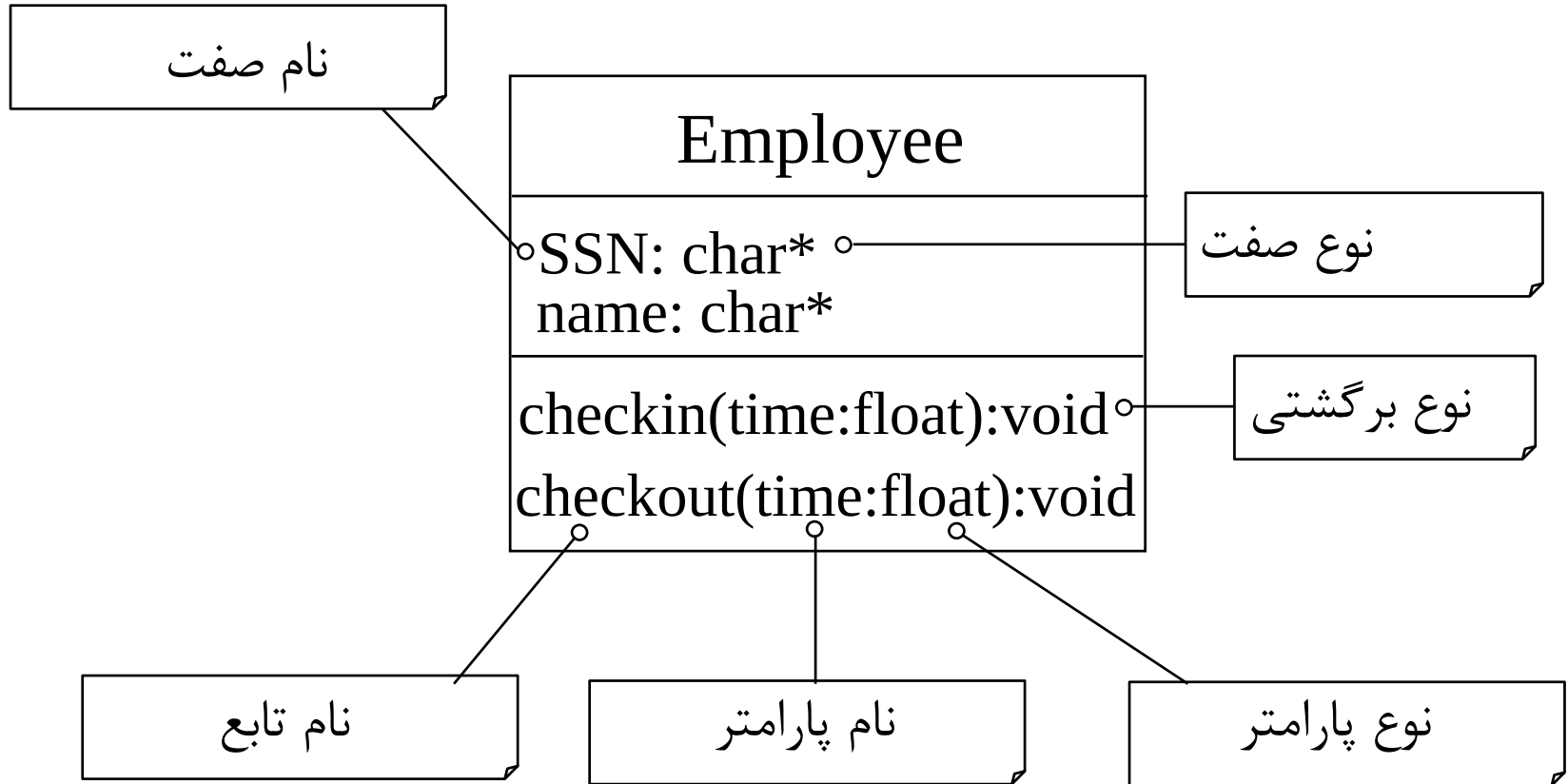
Employee

Employee

SSN
name

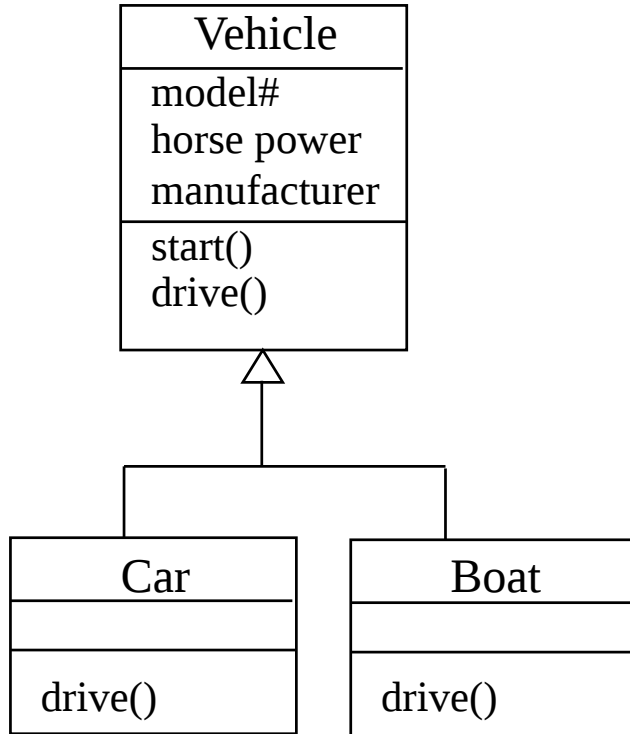
checkIn(time)
checkOut(time)

نمایش نوع در UML



نحو کلی، نوع : نام

رابطه‌ی ارث‌بری



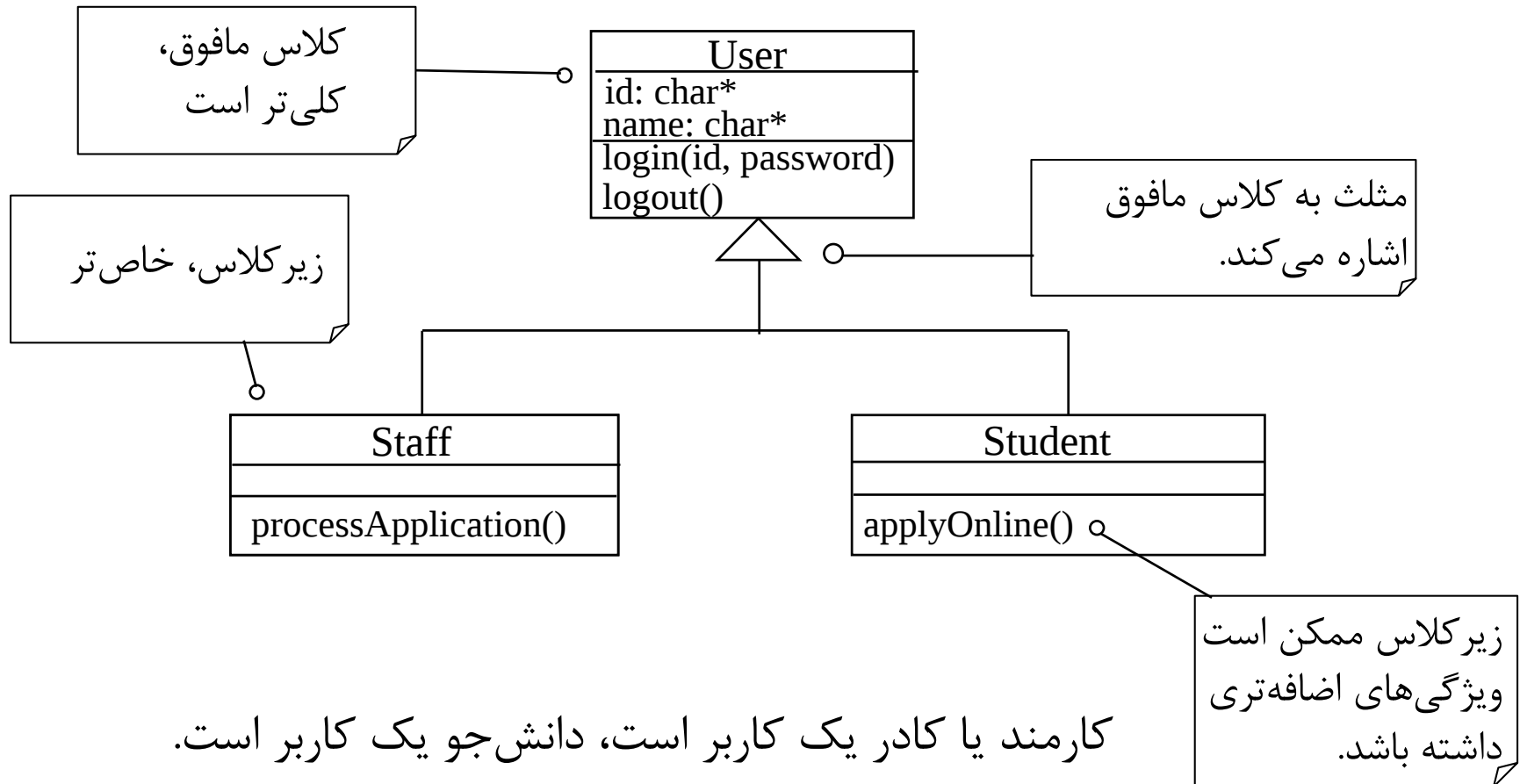
- روابط کلی/خاص را میان مفاهیم بیان می‌کند.

- یک مفهوم، کلی‌تر/خاص‌تر از دیگری است.

- مثال: وسیله‌ی نقلیه، تعمیم یک اتومبیل است؛ اتومبیل، مورد خاصی از وسیله‌ی نقلیه است.

- رابطه‌ی IS-A هم نامیده می‌شود.

مثال: ارث‌بری



کارمند یا کادر یک کاربر است، دانش‌جو یک کاربر است.
ویژگی‌های (یعنی، صفات و عملیات) تعریف شده برای کلاس
مافوق، به صورت خودکار برای زیر کلاس‌ها تعریف می‌شوند.

شیء و صفت

- یک اسم/عبارت اسمی می‌تواند یک کلاس یا یک صفت باشد، چگونه از همدیگر متمایز کنیم؟
- این موضوع غالباً چالشی برای مبتدیان است.
- قواعدی که قابل اعمال هستند عبارتند از:

شیء در کاربرد/دامنه‌ی کاربرد دارای «وجود مستقل» بوده، اما صفت نیست (نمی‌تواند داشته باشد).

مثال: «تعداد صندلی‌ها»، کلاس یا صفت است؟

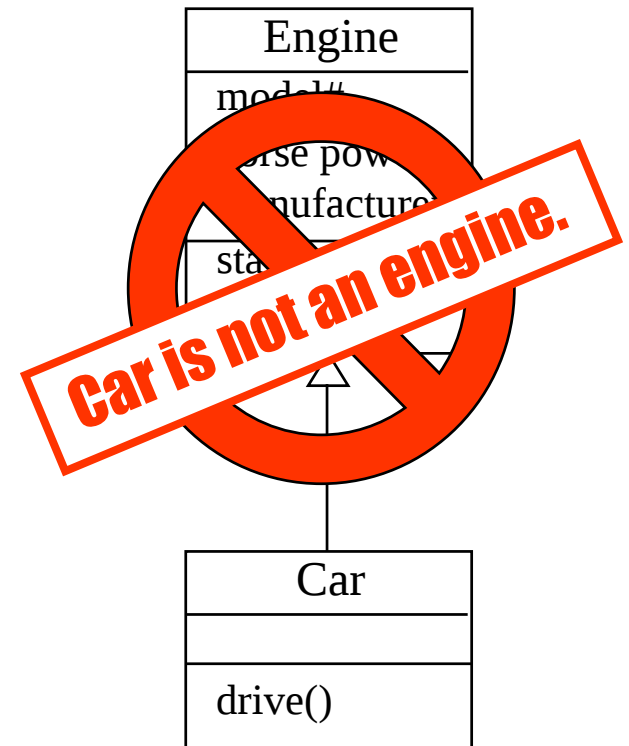
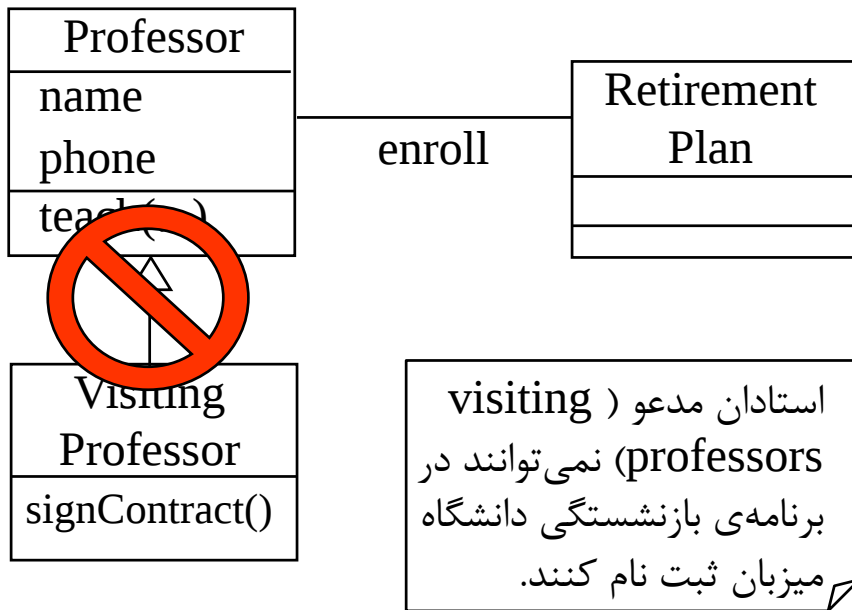
صفت است، چون «تعداد صندلی‌ها» نمی‌تواند بدون ارجاع به یک ماشین، هواپیما یا کلاس درس، همچون «تعداد صندلی‌های یک ماشین»، «تعداد صندلی‌های یک کلاس درس» وجود داشته باشد.

شیء و صفت

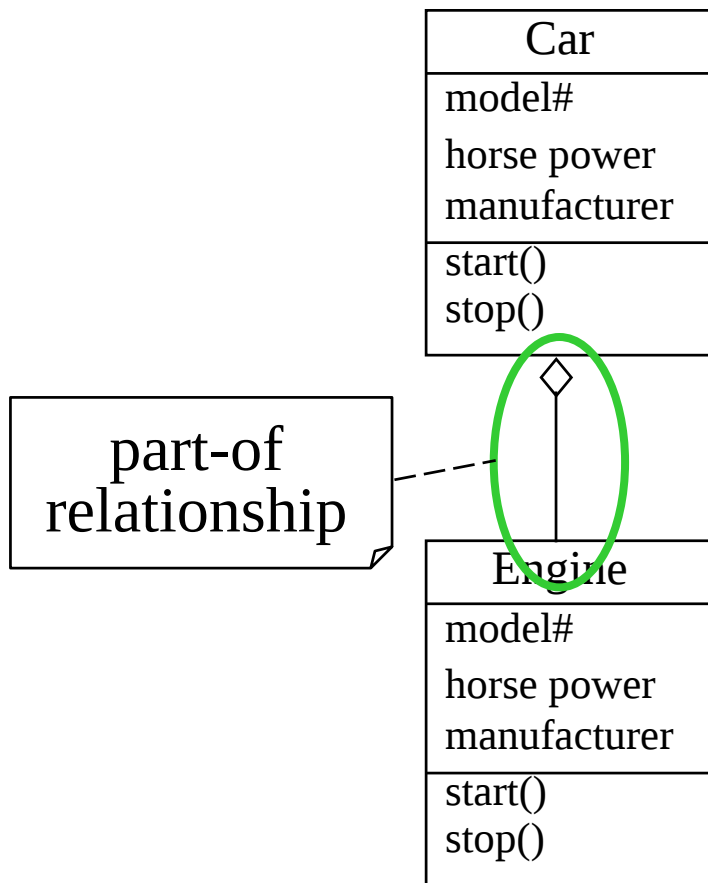
- قواعدی که قابل اعمال هستند عبارتند از:
 - صفات، توصیف اشیاء بوده یا اطلاعات حالت اشیاء را ذخیره می‌نمایند.
 - صفات از نوع اسکالر هستند همچون int، double، string، char و boolean؛ اشیاء این‌گونه نیستند.
 - می‌توانید مقادیر صفات را از طریق صفحه‌کلید وارد کنید، اما نمی‌توانید این کار را در مورد اشیاء انجام دهید.
 - اشیاء بایستی با امضای یک سازنده (صریح یا ضمنی) ایجاد شوند.

دو آزمون برای ارث‌بری

- آزمون IS-A: هر نمونه از یک زیر کلاس، یک نمونه از کلاس مافوق هم هست.
- آزمون هم‌خوانی (conformance): روابط یک کلاس مافوق، روابط زیر کلاس‌ها هم هستند.



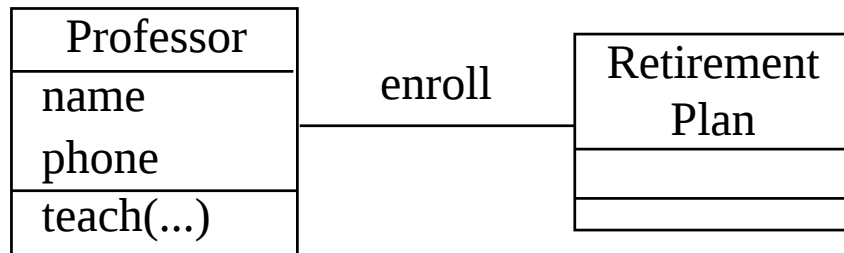
رابطه‌ی تجمیع (aggregation)



- این حقیقت را بیان می‌کند که یک شیء، بخشی از شیء دیگر است.
- مثال: موتور، بخشی از یک ماشین است.
- رابطه‌ی part-of (بخشی از) نیز نامیده می‌شود.

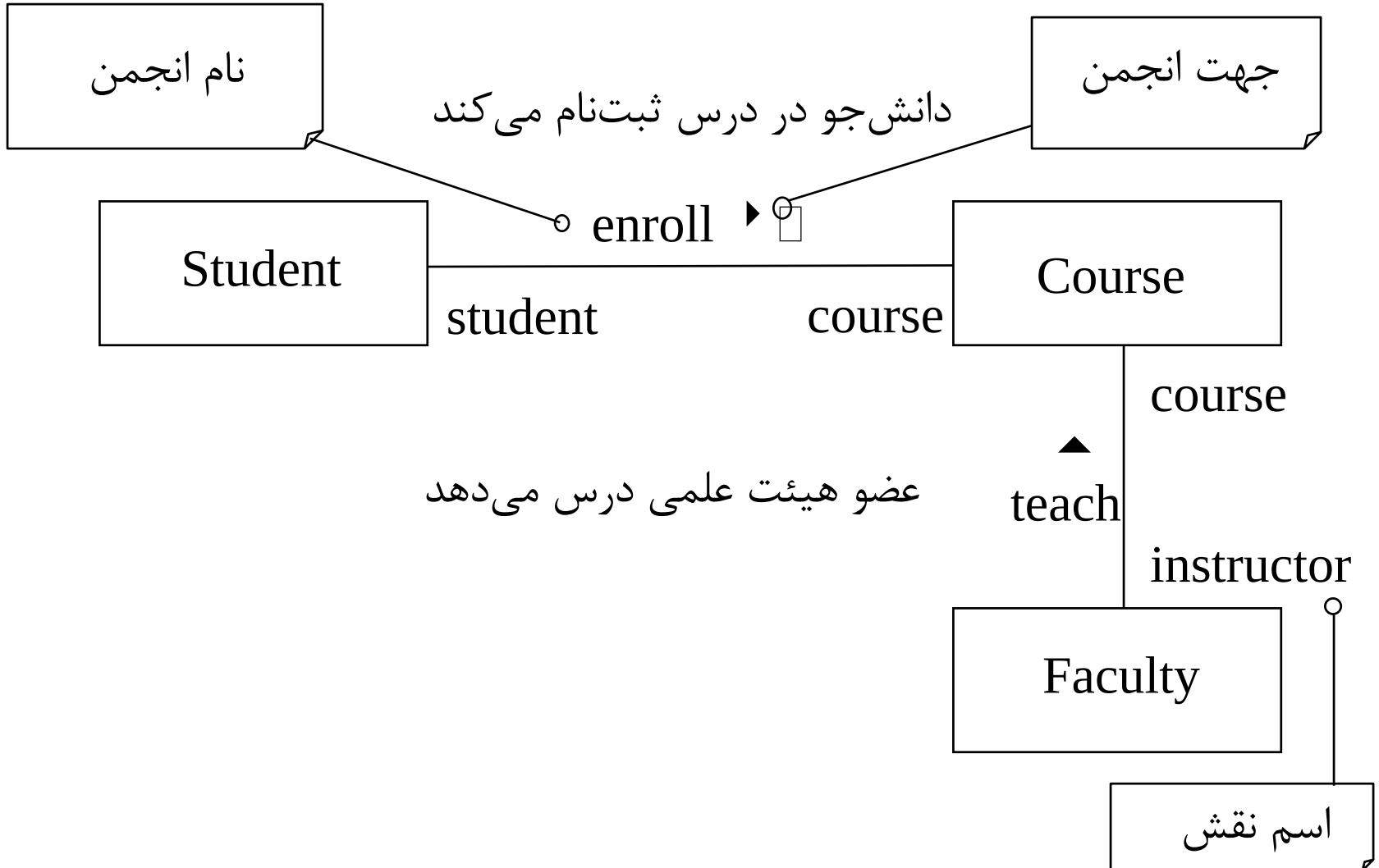
رابطه‌ی انجمنی (association)

- یک رابطه‌ی کلی، غیر از ارث‌بری و تجمیع را بیان می‌کند.
- این‌ها می‌توانند روابط خاص کاربرد، میان دو مفهوم باشند.
- مثال: «مربی درس می‌دهد»، «کاربر حساب دارد».

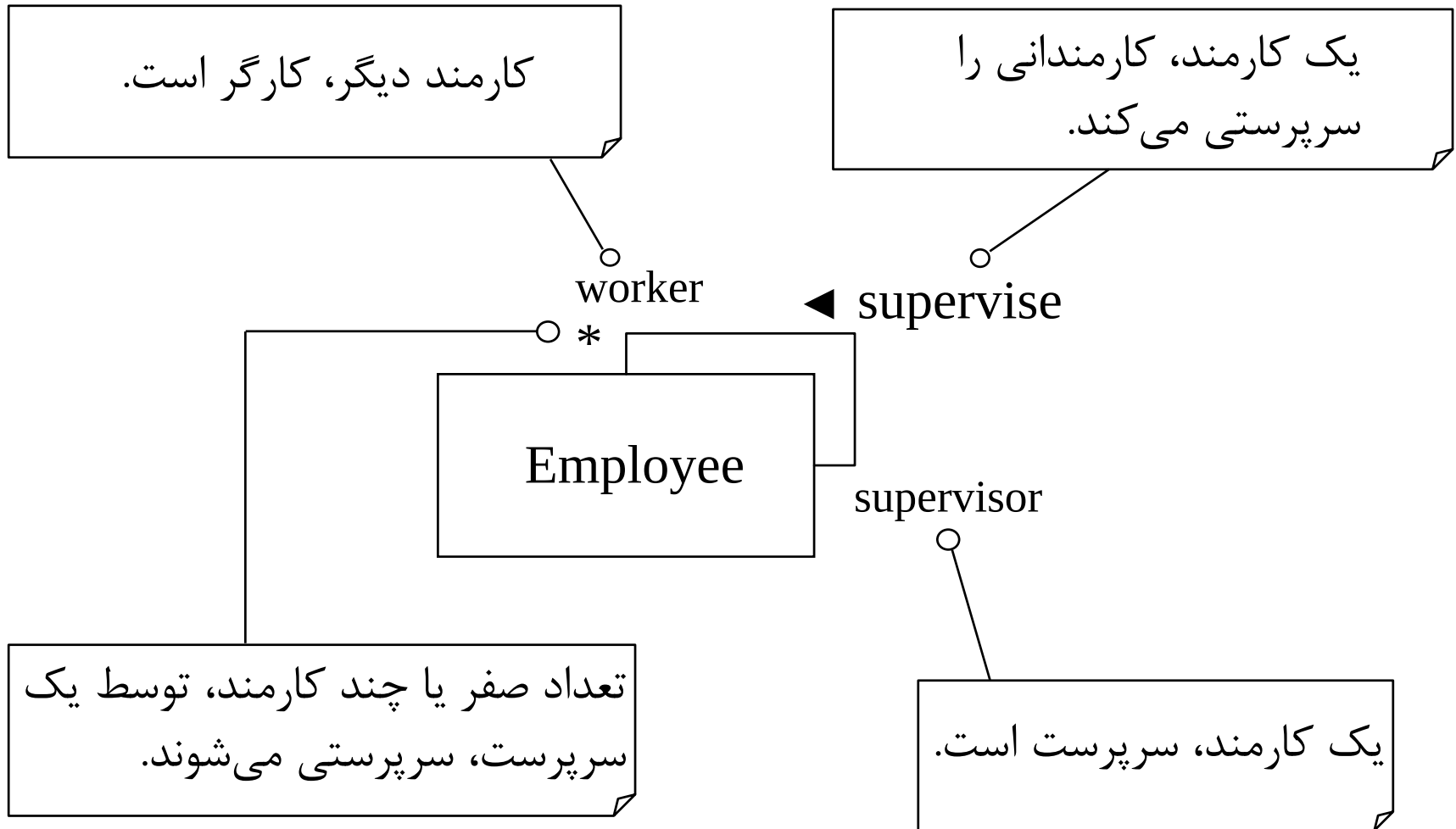


enroll (ثبت‌نام) یک رابطه‌ی ارث‌بری یا تجمیع نیست.

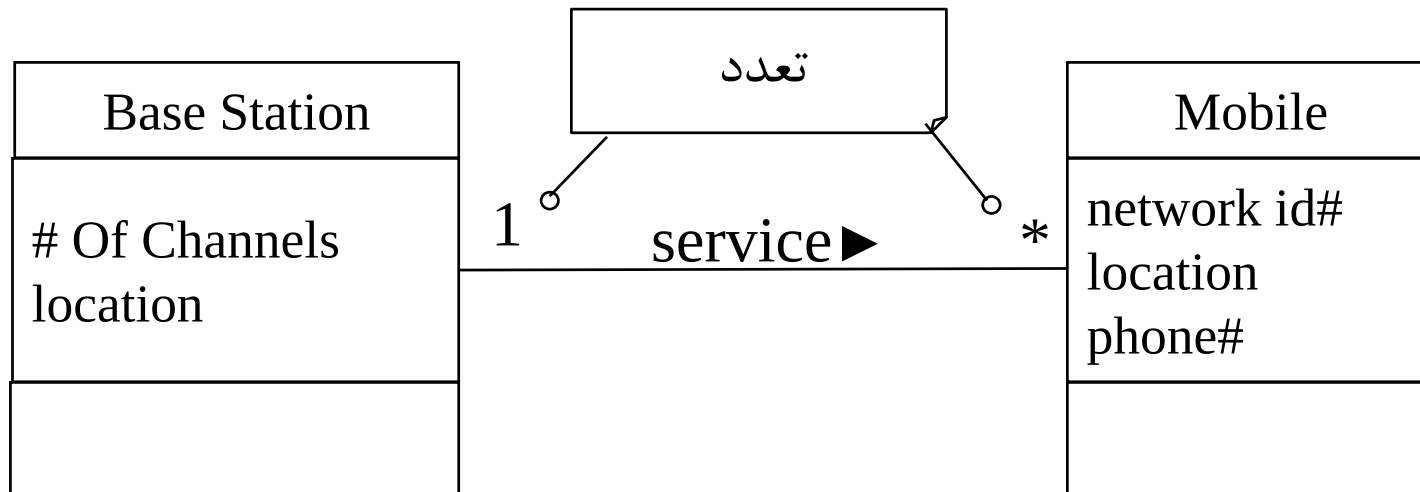
نقش و جهت انجمن



نقش و تعدد



ادعا (assertion) / قيد (constraint) تعدد

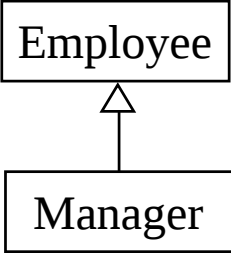
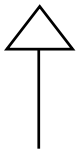
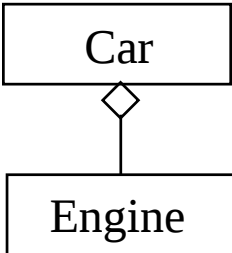
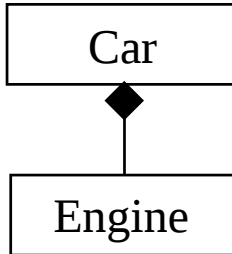


یک ایستگاه، صفر یا چند موبایل را سرویس داده و هر موبایل دقیقاً توسط یک ایستگاه سرویس داده می‌شود.

سایر قیود تعدد عبارتند از:

1	exactly one (default)	1..*	one or more
0..1	zero or one	m..n	m to n
, 0..	zero or more	n	exactly n

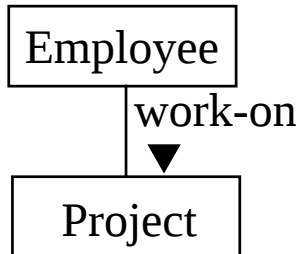
خلاصه: روابط نمودار کلاس UML

مثال	نماد	مفهوم
		ارث‌بری (میان کلاس‌ها) رابطه‌ی IS-A
		تجمیع (میان کلاس‌ها) رابطه‌ی بخشی از
		ترکیب (میان کلاس‌ها) تجمیع انحصاری مالک جزء است

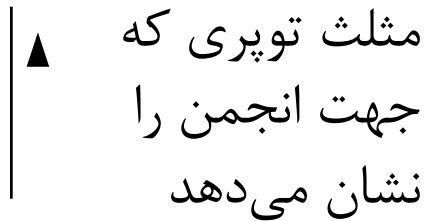
تجمیع با لوزی توپر (ترکیب) به‌ندرت استفاده می‌شود.

روابط نمودار کلاس UML

مثال



نماد

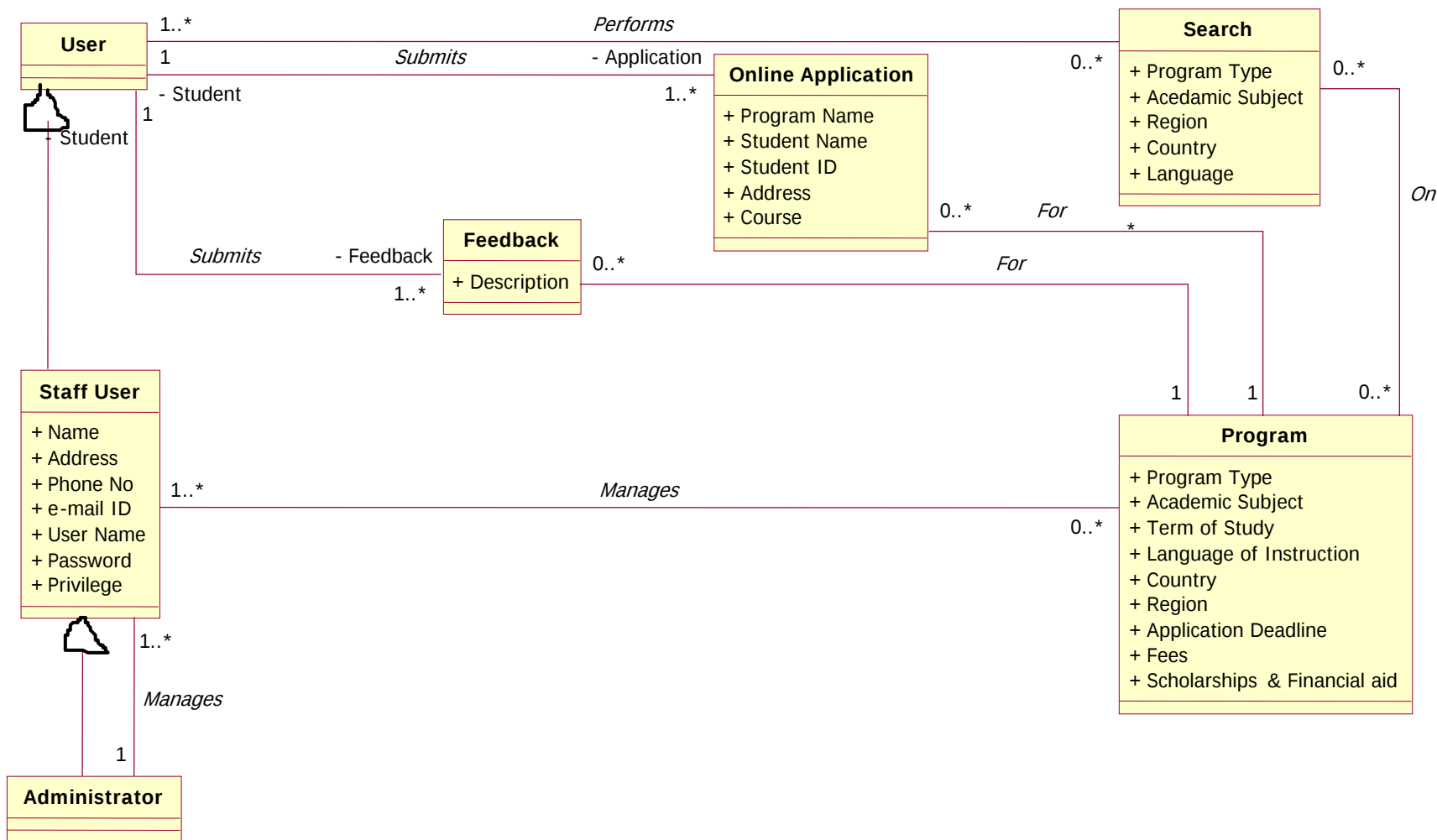


مفهوم

انجمن (میان کلاس ها)
رابطه ی عمومی

نکته: روابط بیشتری در UML هستند اما ما برای مدل سازی دامنه فقط به این سه نیاز داریم.

مدل دامنهی نمونه



تمرین کلاسی

- کارهای زیر را در پروژه‌ی تیمی یا ماشین فروش (اسلاید بعدی) انجام دهید.
- مفاهیمی که در دامنه‌ی کاربرد وجود دارند شناسایی کنید.
- مفاهیم را با تعابیر زیر رده‌بندی کنید:

– کلاس‌ها

– صفات کلاس‌ها

– روابط میان کلاس‌ها

- ارث‌بری

- تجمیع و

- انجمن

تمرین کلاسی: ماشین فروش

ماشین فروش دارای یک نمایشگر، یک صفحه کلید الفباعددی، یک شکاف درج سکه و یک تحویل دهنده‌ی جنس دارد. نمایشگر، اجناس فروشی را مثل شکلات، شیرینی، چیپس، زغال، CD کارتون و الی آخر نشان می‌دهد. هر نوع جنس دارای یک قیمت و برچسبی شامل یکی از حروف A، B، C، ... و یک رقم 1، 2، ... می‌باشد. مشتری، سکه‌ای را از طریق شکاف آن وارد می‌کند و LCD مقدار کل پول را نشان می‌دهد. مشتری بعد از این که به تعداد کافی سکه گذاشت، می‌تواند با فشار دادن یک حرف و یک رقم انتخاب داشته باشد. اگر مقدار کل پول مورد نیاز، بیشتر یا مساوی قیمت جنس انتخابی بود، ماشین فروش، جنس را تحویل داده و بقیه‌ی پول را به مشتری بر می‌گرداند. مشتری می‌تواند نظرش را عوض کرده و با فشار دادن دکمه‌ی بازگشت، سکه‌هایش را درخواست نماید.

اعمال اصول چابک

۱- از نزدیک با مشتری و کاربران کار کنید تا کاربرد و دامنه‌ی کاربرد آن‌ها را درک کنید.

۲- تنها در صورتی مدل‌سازی دامنه انجام دهید که لازم است. ساده شروع کنید و به‌صورت پله‌ای آن را گسترش دهید.

۳- مدل‌سازی دامنه ممکن است به‌صورت همزمان با مدل‌سازی تعامل کنش‌گر-سیستم، مدل‌سازی تعامل اشیاء، مدل‌سازی حالت اشیاء و مدل‌سازی فعالیت انجام شود.

گام‌های مدل‌سازی دامنه

- گام‌های مدل‌سازی دامنه

– اگر برنامه‌ی درسی، تحلیل و طراحی OO داشته باشد، گام‌های مدل‌سازی دامنه می‌تواند در آن درس تدریس شود.