

فصل ششم

مدل سازی نیازمندی ها

سناریو ها، اطلاعات و کلاس های تحلیل

تحلیل نیازمندی ها

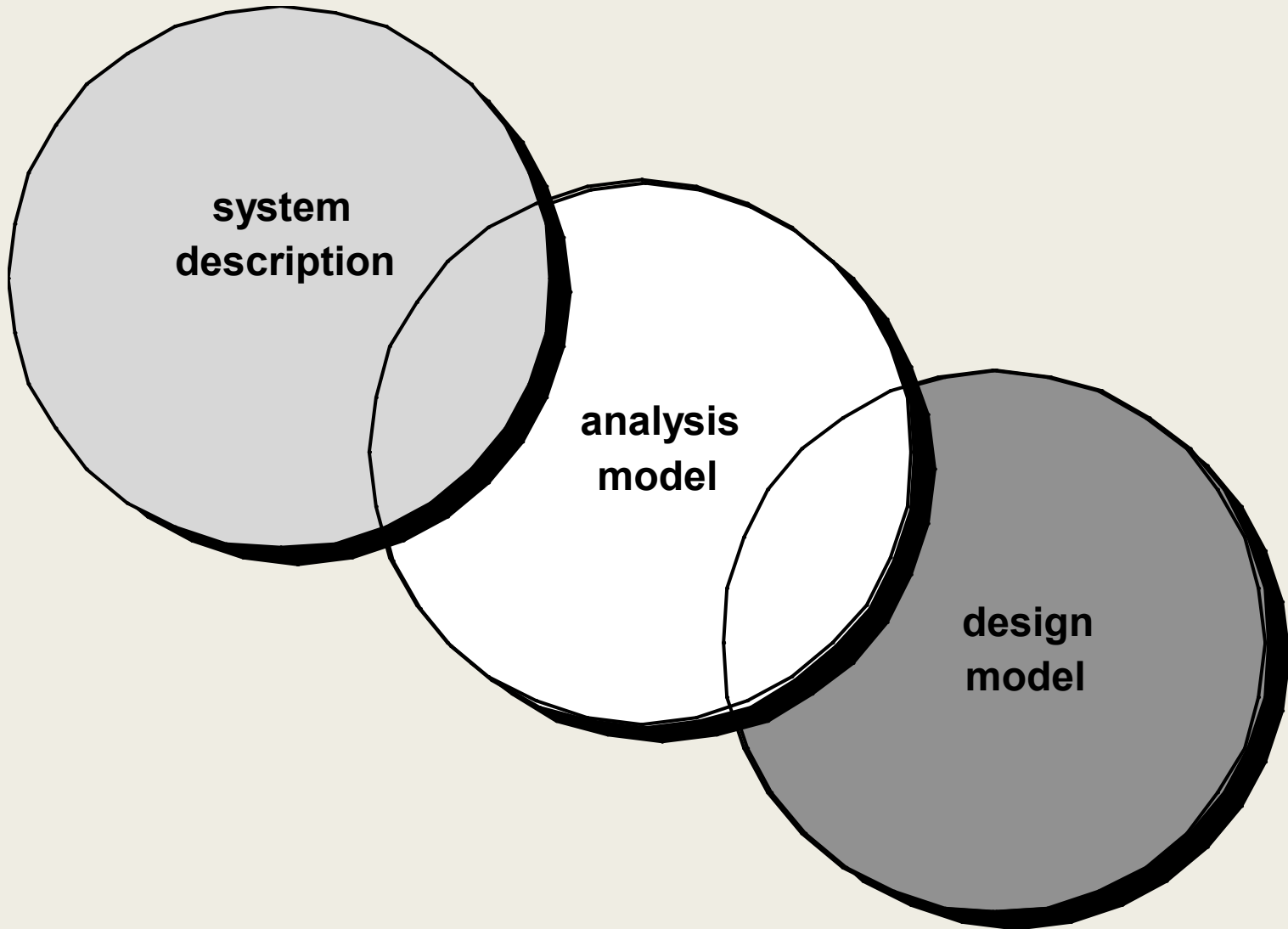
■ تجزیه نیازمندی ها

- ویژگی های عملیاتی نرم افزار را مشخص می کند
- رابط نرم افزار را با دیگر عناصر سیستم نشان می دهد
- محدودیت هایی را که نرم افزار باید بر آورده کند، برقرار می کند.

■ تحلیل نیازمندی ها اجازه می دهد تا مهندس نرم افزار (در این نقش تحلیلگر

یا مدل کننده نامیده می شود) :

- در مورد نیازمندی های اساسی مقرر در طی وظایف مهندسی نیازمندی های اولیه، شرح دهد
- مدل هایی بسازد که سناریوهای کاربر، فعالیت های عملکردی، کلاس های مشکل و روابط آنها، رفتار کلاس و سیستم و جریان داده به گونه ای که تبدیل می شود را شرح دهد.



قواعد اولیه

- مدل باید بر نیازهایی که در دامنه مشکل یا کسب و کار قابل مشاهده هستند تمرکز کند. سطح انتزاع باید نسبتاً بالا باشد.
- هر عنصر مدل تحلیل باید به درک کلی از نیازمندی های نرم افزار بیافزاید و بینشی در دامنه اطلاعات، عملکرد و رفتار سیستم را فراهم کند.
- در نظر گرفتن زیرساخت ها و سایر مدل های غیر عملکردی را تا طراحی به تعویق افتد.
- اتصال در سراسر سیستم را کاهش دهید.
- مطمئن شوید که مدل تحلیل، ارزشی برای همه ذینفعان فراهم می کند.
- مدل را در حد امکان ساده نگه دارید.

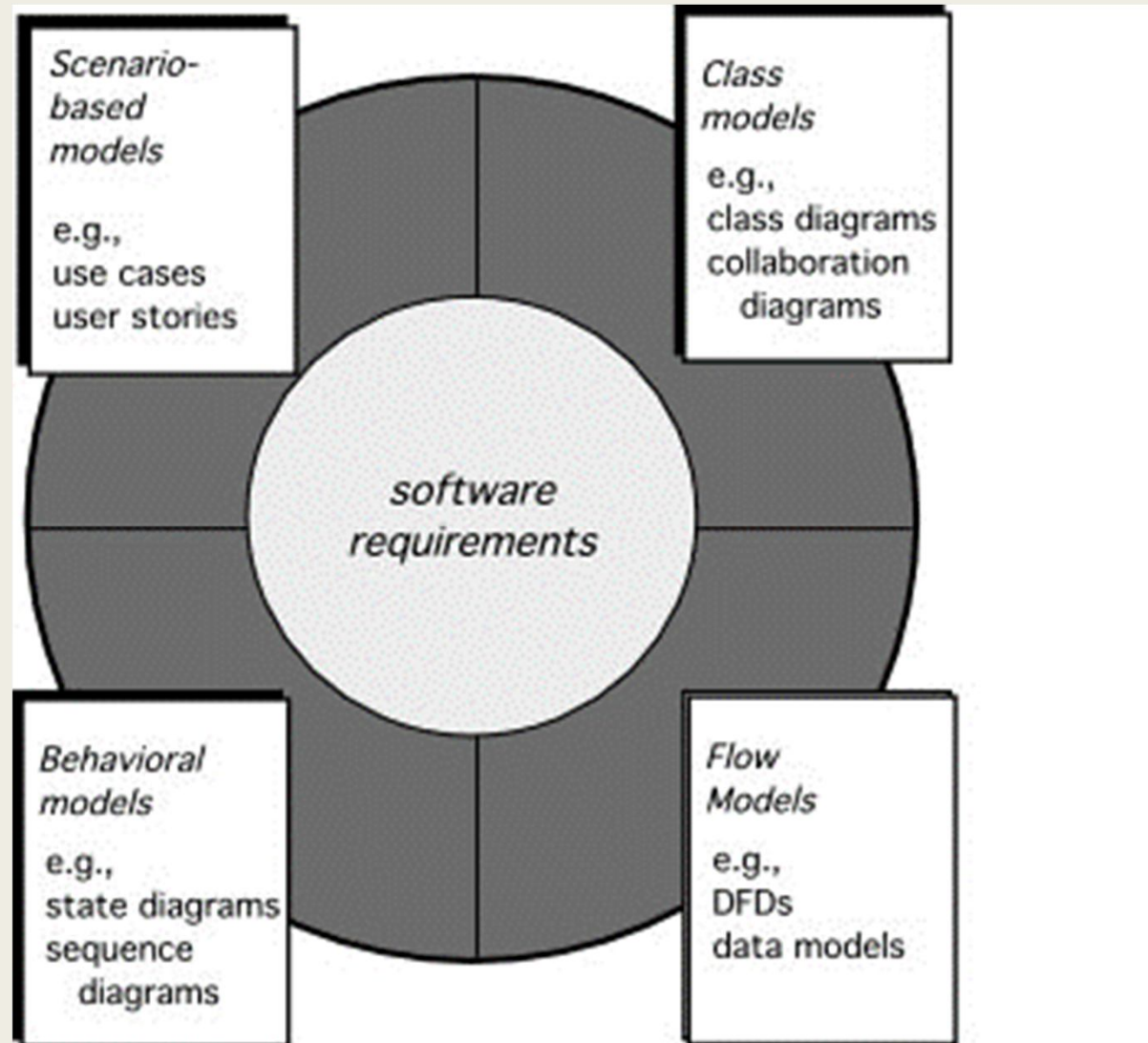
تحلیل دامنه

■ تجزیه و تحلیل دامنه نرم افزار شناسایی، تحلیل و مشخص کردن نیازمندی های مشترک از یک دامنه کاربرد خاص، به طور معمول برای استفاده مجدد در پروژه های متعدد در دامنه کاربرد آن است...

■ [تحلیل دامنه شی گرا] شناسایی، تجزیه و تحلیل و مشخص کردن قابلیت های مشترک و قابل استفاده مجدد در یک دامنه کاربرد خاص از نظر اشیاء، کلاس ها، زیر مجموعه ها و چارچوب های مشترک است...

Donald Firesmith

اجزای تحلیل دامنه



مدل سازی مبتنی بر سناریو

■ "[Use-Caseها] به طور ساده کمکی به تعریف آنچه خارج از سیستم وجود دارد (بازیگران) و آنچه که باید توسط سیستم انجام شود (Use-Caseها) است." *Ivar Jacobson*

(۱) در چه موردی باید بنویسیم؟

(۲) در مورد آن چقدر باید بنویسیم؟

(۳) توضیحات خود را چگونه ما باید شرح دهیم؟

(۴) چگونه باید توضیحات را سازماندهی کنیم؟

در چه مورد بنویسیم؟

- **اكتساب و استنباط** – اطلاعاتی را که برای آغاز نوشتن usecaseها نیاز دارید فراهم می کند.
- **جلسات گردآوری نیازمندی ها، اسقرار تابع کیفی (QFD) و دیگر مکانیسم های مهندسی نیازمندی ها** به کار می رود تا
 - ذینفعان شناسایی شوند.
 - دامنه مشکل تعریف شود
 - اهداف عملیاتی کلی مشخص شود
 - اولویت ها تعیین شوند
 - کلیه نیازمندی های کارکردی شناخته شده تعیین شوند و
 - توصیف چیزهایی (اشیاء) که توسط سیستم دستکاری خواهند شد.
- برای شروع توسعه مجموعه ای از useCaseها، **عملکردها یا فعالیت های انجام شده** توسط یک بازیگر خاص را لیست کنید.

چقدر درباره آن بنویسیم؟

- با پیشرفت مکالمه های بیشتر با ذینفعان، تیم جمع آوری نیازمندی ها، Usecase ها را برای هر عملکرد مشخص شده، توسعه می دهند.
- به طور کلی، Usecase ها ابتدا به شکل داستانی غیر رسمی نوشته شده است.
- اگر رسمیت بیشتر نیاز باشد، همان Usecase با استفاده از قالب ساخت یافته شبیه به مورد پیشنهاد شده، بازنویسی می شود.

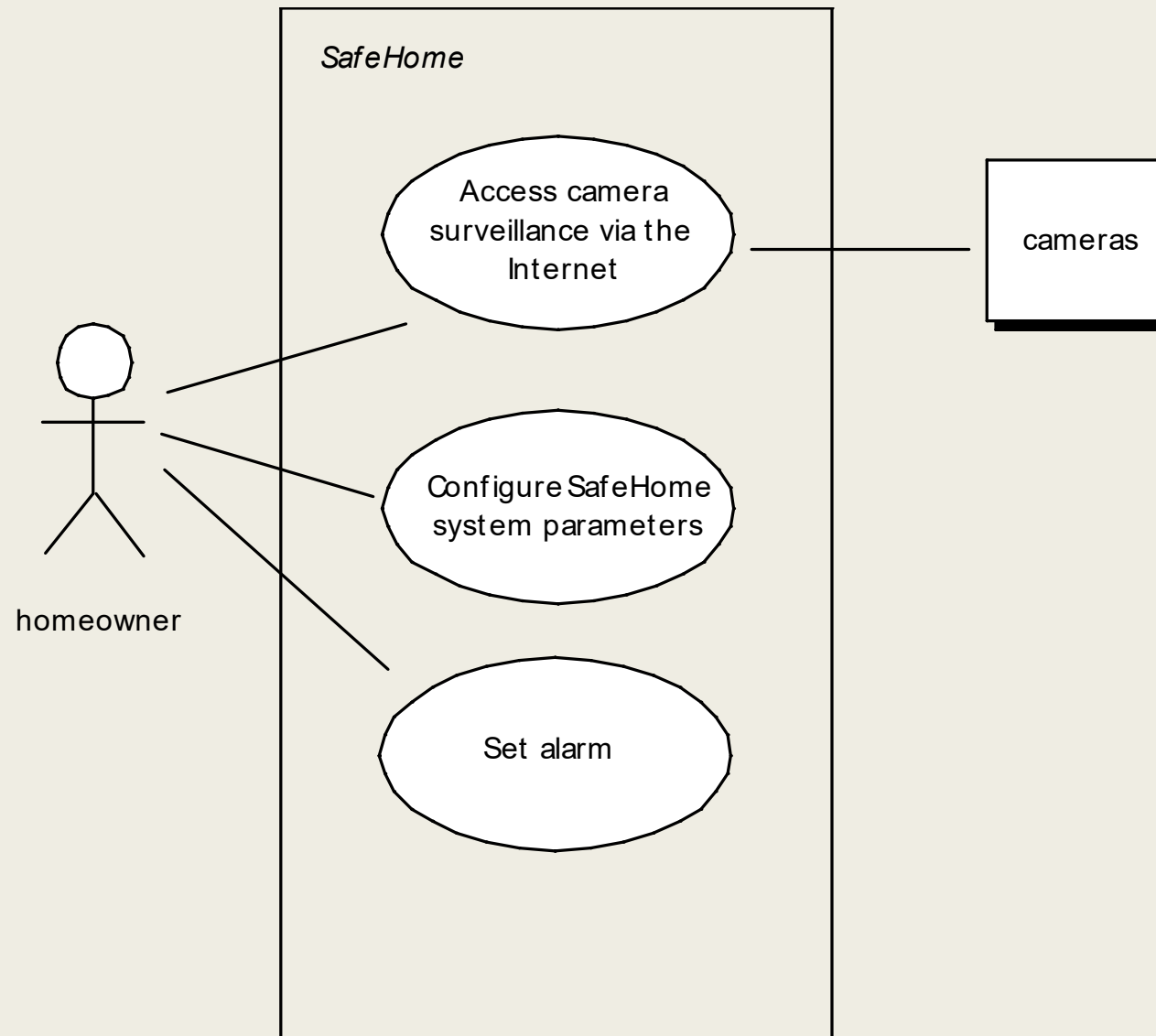
Use-Case ها

- یک سناریو که " موضوع استفاده " را برای یک سیستم توصیف می کند
- **بازیگران (actor)** نقش افراد یا دستگاه ها را به عنوان عملکرد سیستم نشان می دهند
- **کاربران** می توانند نقش های مختلفی را برای یک سناریوی خاص بازی کنند

ساخت Use-Case

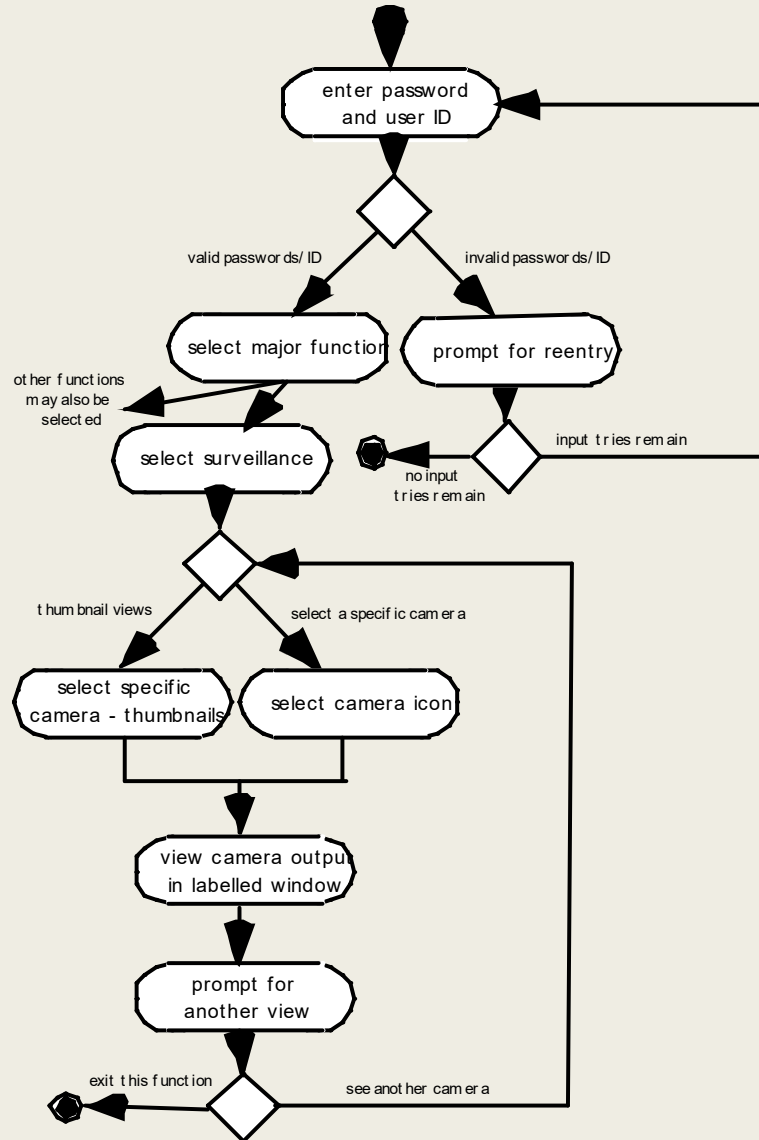
- **وظایف یا عملکردهای اصلی که توسط بازیگر انجام می شود چیست؟**
- **چه اطلاعات سیستمی، بازیگر به دست خواهد آورد، تولید خواهد کرد یا تغییر خواهد داد؟**
- **آیا بازیگر باید سیستم را در مورد تغییرات محیط خارجی مطلع کند؟**
- **بازیگر چه اطلاعاتی از سیستم می خواهد؟**
- **آیا بازیگر می خواهد در مورد تغییرات غیر منتظره مطلع شود؟**

نمودار Use-Case

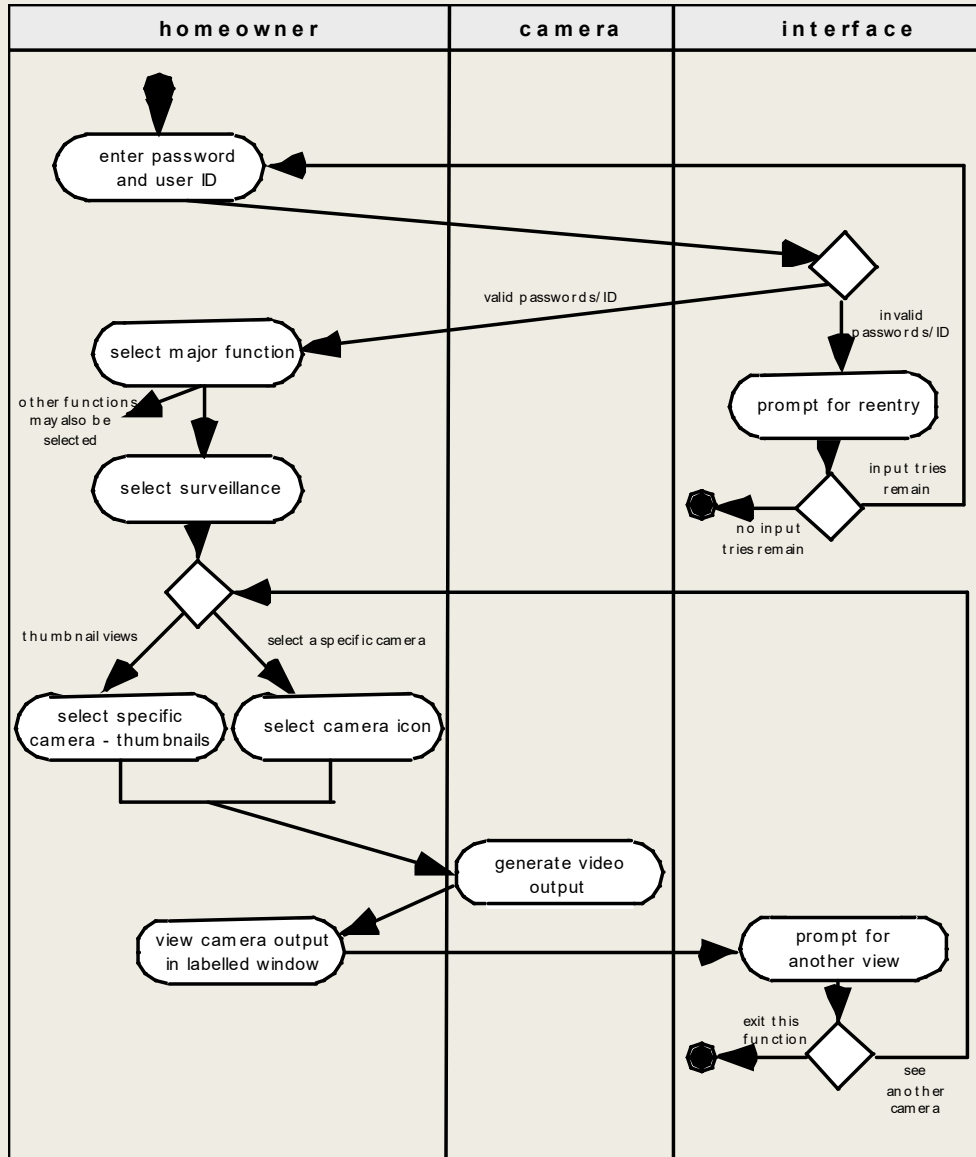


نمودار Activity

با ارائه یک نمایش گرافیکی
جریان تعامل در یک سناریوی
خاص، UseCase را تکمیل
می کند



نمودار Swimlane



به مدل کننده اجازه می دهد تا جریان فعالیت های توصیف شده در Usecase را نشان دهد و در عین حال نمایش دهد چه بازیگری (اگر چندین بازیگر در یک UseCase وجود داشته باشند) یا کلاس تحلیلی مسئولیت اقدامی را که توسط یک مستطیل مشخص شده، دارد

مدل سازی داده

- اشیاء داده را مستقل از پردازش بررسی می کند
- بر دامنه داده تمرکز می کند
- یک مدل در سطح انتزاعی مشتری ایجاد می کند
- نشان می دهد چگونه اشیاء داده مربوط به یکدیگر هستند

شی داده چیست؟

- نمایش تقریبا هر اطلاعات ترکیبی که باید توسط نرم افزار درک شود.
- اطلاعات مرکب - چیزی است که تعدادی از خواص یا ویژگی های مختلف دارد
- می تواند یک موجود خارجی باشد (به عنوان مثال هر چیزی که اطلاعات را تولید می کند یا مصرف می کند)، یک چیز (مثلا یک گزارش یا صفحه نمایش)، رخداد (مثلا یک تماس تلفنی) یا رویداد (مانند یک زنگ هشدار)، فروشنده، یک واحد سازمانی (به عنوان مثال، بخش حسابداری)، یک مکان (به عنوان مثال یک انبار) یا یک ساختار (به عنوان مثال یک فایل).
- توضیح داده شیء شامل شی داده و تمام ویژگی های آن است.
- یک شیء داده تنها اطلاعات را محاسبه می کند - هیچ مرجع در یک شیء داده به عملیات هایی که بر روی داده ها عمل می کنند وجود ندارد.

اشیاء داده و صفات

- یک شیء داده شامل مجموعه ای از صفات است که به عنوان یک جنبه، کیفیت، مشخصه یا توصیفگر شی عمل می کند

object: automobile

attributes:

make

model

body type

price

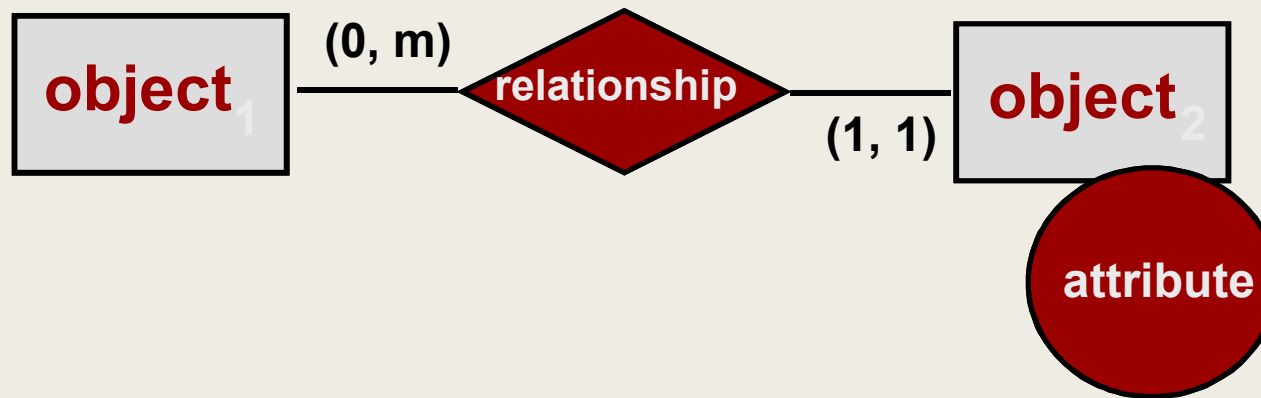
options code

رابطه چیست؟

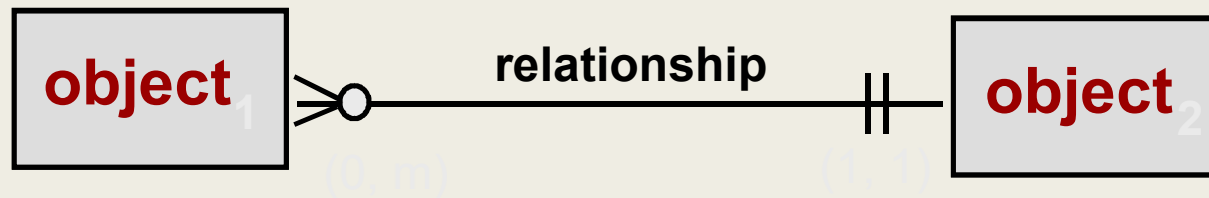
- اشیاء داده ای به شیوه های مختلف به یکدیگر متصل می شوند.
- ارتباط بین **شخص** و **ماشین** برقرار است زیرا دو موضوع مرتبط هستند.
- یک فرد مالک یک ماشین است
- یک فرد برای رانندگی یک خودرو بیمه شده است
- روابط **مالک بودن** و **بیمه رانندگی شدن** ارتباطات مربوطه بین **شخص** و **ماشین** را تعریف می کند.
- چندین نمونه از رابطه می تواند وجود داشته باشد
- اشیاء می توانند به روش های مختلفی مرتبط باشند

نمایش ERD

■ یک شکل رایج:



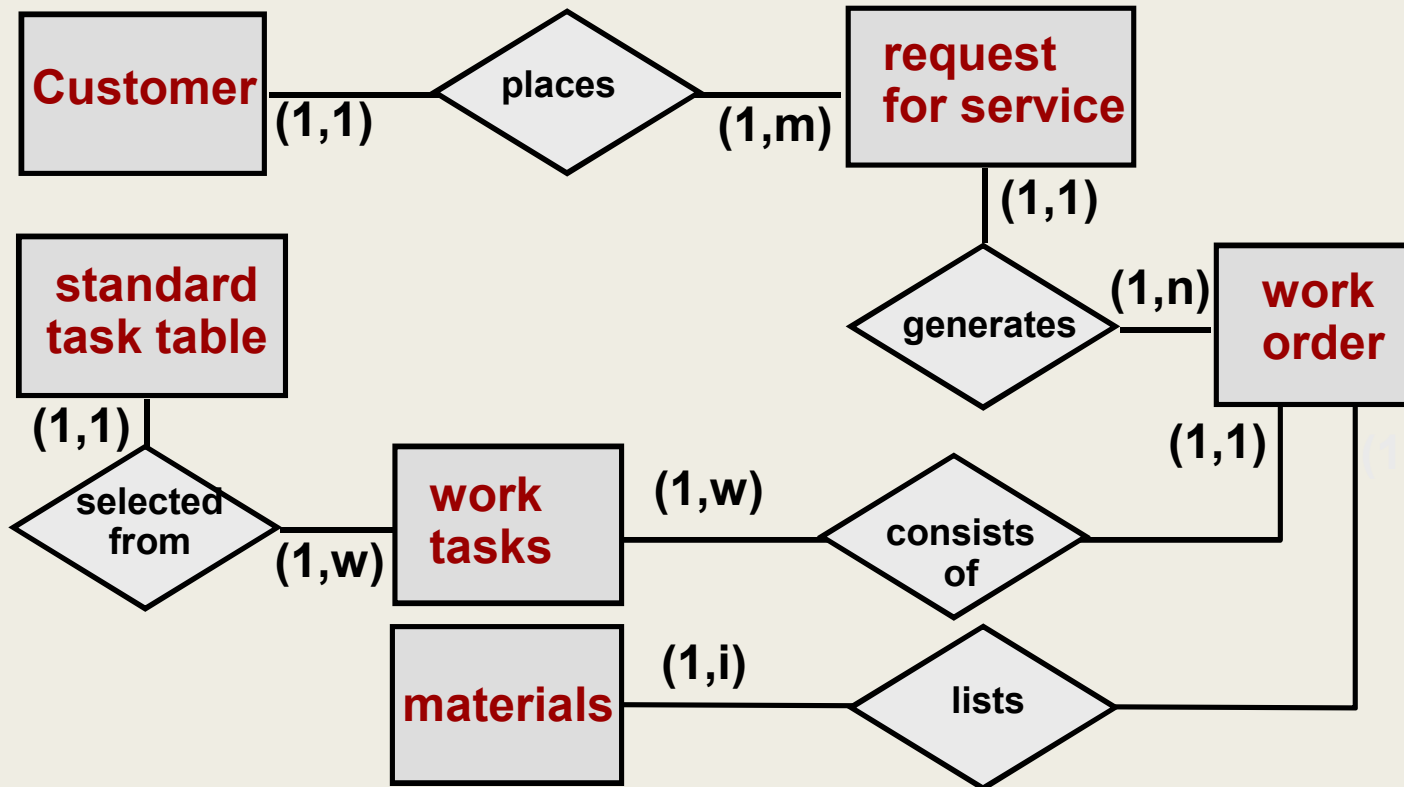
■ شکل رایج دیگر:



ساخت ERD

- **سطح ۱** – مدل کردن تمام اشیاء داده‌ها (موجودیت‌ها) و "ارتباطات" آنها به یکدیگر
- **سطح ۲** – مدل کردن تمام موجودیت‌ها و روابط
- **سطح ۳** – مدل تمام موجودیت‌ها، روابط و صفاتی است که عمق بیشتری را ارائه می‌دهند

مثالی از ERD



مدل سازی بر اساس کلاس

■ مدل سازی مبتنی بر کلاس، موارد زیر را نمایش می دهد:

– **اشیاء** که سیستم دستکاری خواهد کرد

– **عملیات** (روش ها یا خدمات نیز نامیده می شوند) است که در

اشیاء برای تاثیر دستکاری به کار می روند

– **روابط** (برخی سلسله مراتبی) بین اشیاء

– **همکاری ها** که بین کلاس های تعریف شده رخ می دهند.

■ عناصر یک مدل مبتنی بر کلاس عبارتند از کلاس ها و اشیاء،

صفات، عملیات، مدل های CRC، نمودار های همکاری و بسته ها.

شناسایی کلاس های تحلیل

- بررسی سناریوهای کاربرد توسعه داده شده به عنوان بخشی از مدل نیازمندی ها و انجام "تجزیه دستوری"
- کلاس ها با تفکیک اسم یا عبارات اسمی و وارد کردن آن در یک جدول ساده تعیین می شوند.
- به مترادف ها باید دقت شود.
- اگر کلاس (اسم) برای اجرای یک راه حل لازم است، بخشی از فضای راه حل است؛ در غیر این صورت اگر کلاس فقط برای توصیف راه حل ضروری باشد، بخشی از فضای مشکل است.
- اما زمانی که همه اسم ها مجزا باشند چه باید بکنیم؟

چگونگی شناسایی کلاس های تحلیل

■ کلاس های تجزیه و تحلیل خود را در یکی از روش های زیر نشان میدهد:

- موجودیت های خارجی (مثلا سیستم ها، دستگاه ها، افراد دیگر) که اطلاعات تولید یا مصرف می کنند

- چیزها (مثلا گزارش ها، نمایش ها، نامه ها، علائم) که بخشی از دامنه اطلاعات برای مشکل هستند

- حوادث یا وقایع (به عنوان مثال انتقال مالکیت یا تکمیل یک سری از حرکات ربات) که در حین عملیات سیستم رخ می دهد

- نقش ها (مثلا مدیر، مهندس، فروشنده) که توسط افرادی که با سیستم در تعامل هستند، بازی می شود

چگونگی شناسایی کلاس های تحلیل

■ کلاس های تجزیه و تحلیل خود را در یکی از روش های زیر نشان
میدهد:

- واحدهای سازمانی (به عنوان مثال بخش، گروه، تیم) که مرتبط با به یک کاربرد
هستند

- اماکن (به عنوان مثال، طبقه تولید یا بندر بارگیری) که زمینه مشکل و عملکرد کلی را
ایجاد می کنند

- ساختارها (به عنوان مثال، حسگرها، وسایل نقلیه چهار چرخ یا کامپیوترها) که کلاسی
از اشیاء یا کلاس های مرتبط با اشیاء را تعریف می کنند

کلاس های بالقوه

- **اطلاعات حفظ شده** کلاس بالقوه در تجزیه و تحلیل فقط در صورتی مفید خواهد بود که اطلاعات مربوط به آن باید به یاد سپرده شود تا سیستم بتواند کار کند.
- **خدمات مورد نیاز** کلاس بالقوه باید مجموعه ای از عملیات قابل شناسایی را داشته باشد که می تواند ارزش های صفات آن را به نحوی تغییر دهد.
- **صفات چندگانه** در طی تحلیل نیازمندی ها، تمرکز باید بر روی اطلاعات "عمده" باشد؛ یک کلاس با یک صفت تنها ممکن است فقط در حین طراحی مفید باشد، اما بهتر است به عنوان صفتی از کلاس دیگری در طول فعالیت تحلیل نمایش داده شود.
- **ویژگی های مشترک** مجموعه ای از صفات می تواند برای کلاس بالقوه تعریف شود و این ویژگی ها برای همه نمونه های کلاس اعمال می شود.
- **عملیات مشترک** مجموعه ای از عملیات را می توان برای کلاس بالقوه تعریف کرد و این عملیات روی همه نمونه های کلاس اعمال می شود.
- **نیازمندی های اساسی** موجودیت های خارجی که در فضای مشکل هستند و اطلاعاتی را تولید یا مصرف می کنند که برای عملیات هر راه حل سیستم ضروری است، تقریباً همیشه به عنوان کلاس در مدل نیازمندی ها تعریف خواهد شد.

تعریف صفات

■ صفات کلاسی را توصیف می کنند که برای شامل شدن در مدل تحلیل برگزیده شده است.

- ایجاد دو کلاس مختلف برای بازیکنان حرفه ای بیس بال

- برای نرم افزار آمار بازی : نام، موقعیت، متوسط ضربه، درصد در اختیار داشتن زمین، سال های بازی و تعداد بازی ها می تواند مرتبط باشند
- برای نرم افزار صندوق بازنشستگی: متوسط حقوق، اعتبار به عنوان واگذاری کامل، گزینه های انتخاب شده طرح بازنشستگی، آدرس پستی و غیره.

تعریف عملیات

■ تجزیه دستوری یک روایت در حال پردازش را انجام دهید و به افعال دقت کنید

■ عملیات می تواند به چهار دسته کلی تقسیم شود:

(۱) عملیاتی که به نوعی داده ها را دستکاری می کنند (به عنوان مثال،

اضافه کردن، حذف، مرتب سازی مجدد، انتخاب)

(۲) عملیات که محاسباتی را انجام می دهند

(۳) عملیات که درباره حالت یک شی مورد بررسی می کند

(۴) عملیاتی که یک شی را برای وقوع یک رخداد کنترلی نظارت می کنند.

مدل های CRC

■ مدل سازی کلاس - مسئولیت - همکار (CRC) ابزار ساده ای برای شناسایی و سازماندهی کلاس هایی است که مربوط به نیازهای سیستم یا محصول می باشد. Ambler مدل سازی CRC را اینگونه توصیف می کند:

■ یک مدل CRC واقعا مجموعه ای از کارت های شاخص استاندارد است که کلاس ها را نشان می دهند. کارت ها به سه بخش تقسیم می شوند. در بالای کارت نام کلاس را می نویسید. در بدنه کارت، مسئولیت های کلاس در سمت چپ و همکاران در سمت راست لیست می شوند.

مدل سازی CRC

Class: FloorPlan	
Description:	
Responsibility:	Collaborator:
defines floor plan name/type	
manages floor plan positioning	
scales floor plan for display	
scales floor plan for display	
incorporates walls, doors and windows	Wall
shows position of video cameras	Camera

انواع کلاس

- **کلاس های موجودیت**، که کلاس های مدل یا کسب و کار نیز نامیده می شوند، مستقیماً از متن مشکل استخراج می شوند (به عنوان مثال، نقشه طبقه و حسگر).
- **کلاس های مرزی** برای ایجاد واسط (مثلاً صفحه نمایش تعاملی یا گزارش های چاپی) استفاده می شود که کاربر با استفاده از نرم افزار آن را می بیند و با آن تعامل دارد.
- **کلاس های کنترل کننده** یک "واحد کار" را از ابتدا تا انتها مدیریت می کنند. به این معنا که کلاس های کنترل کننده را می توان برای مدیریت موارد زیر طراحی شوند:
 - ایجاد یا به روز رسانی اشیاء موجودیت؛
 - نمونه سازی اشیاء مرزی زیرا اطلاعات را از اشیاء موجودیت به دست می آورند؛
 - ارتباط پیچیده بین مجموعه ای از اشیاء؛
 - اعتبار سنجی داده های ارتباطی میان اشیاء یا کاربر و برنامه

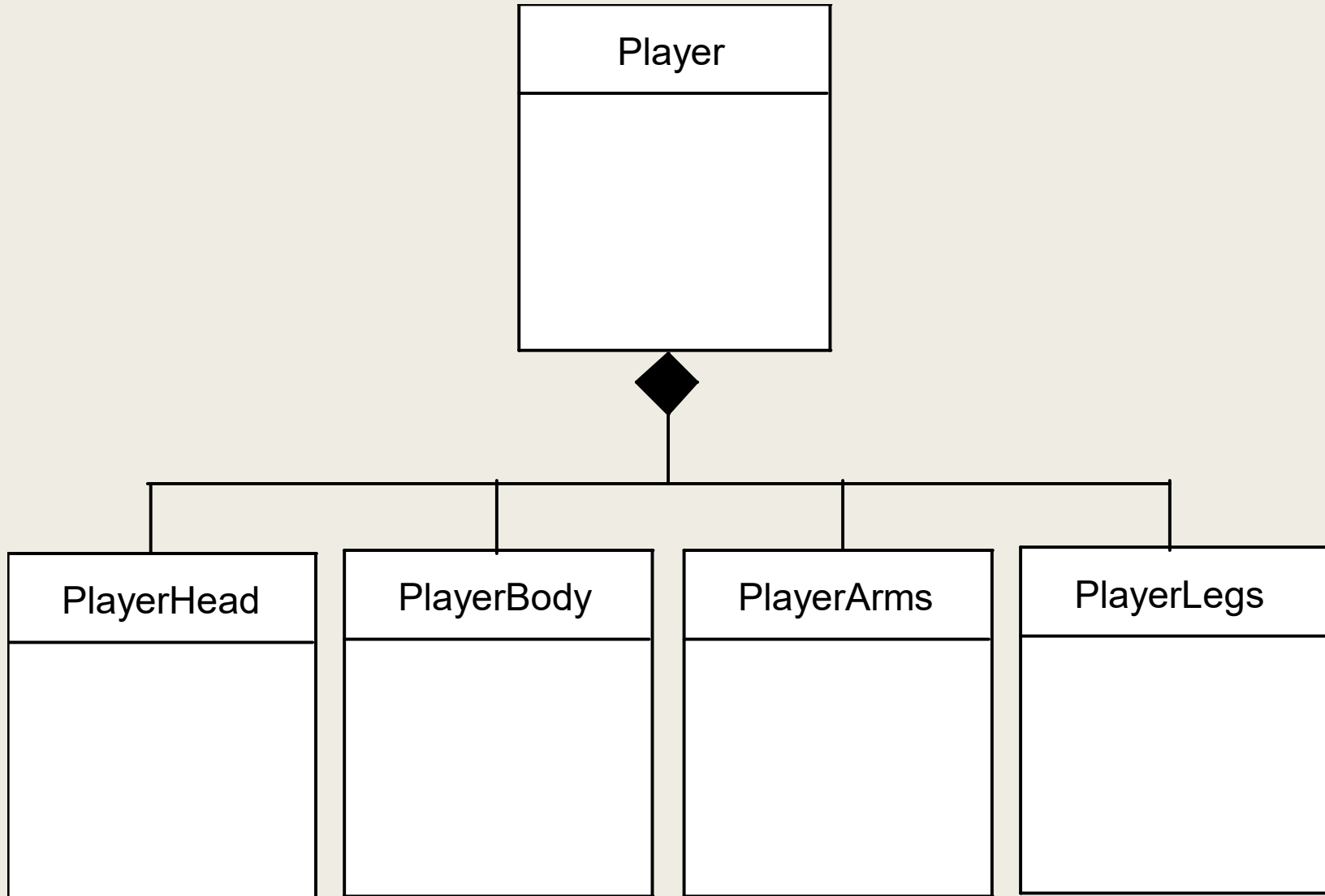
مسئولیت ها

- هوش سیستم باید در میان کلاس ها توزیع شود تا بتواند به بهترین وجه به نیازهای مشکل پاسخ دهد
- هر مسئولیتی باید تا حد امکان به طور کلی بیان شود
- اطلاعات و رفتار مربوطه باید در یک کلاس قرار بگیرند
- اطلاعات در مورد یک چیز باید در یک کلاس واحد متمرکز شود، نه در میان کلاس های مختلف توزیع شود.
- در صورت لزوم، مسئولیت ها باید در میان کلاس های مرتبط به اشتراک گذاشته شود.

همکاری ها

- کلاسها مسئولیت های خود را به یکی از این دو روش انجام می دهند:
 - یک کلاس می تواند از عملیات خود برای دستکاری صفات خود استفاده کند، در نتیجه یک مسئولیت خاص را انجام می دهد، یا
 - یک کلاس می تواند با کلاس های دیگر همکاری کند.
- همکاری ها روابط بین کلاس ها را مشخص می کنند
- همکاری ها با تعیین اینکه آیا یک کلاس می تواند هر مسئولیت خود را برآورده کند، شناسایی می شود
- سه روابط عمومی متفاوت بین کلاس ها:
 - رابطه بخشی از (is-part-of)
 - رابطه دانشی دارد از (has-knowledge-of)
 - رابطه بستگی دارد به (depends-upon)

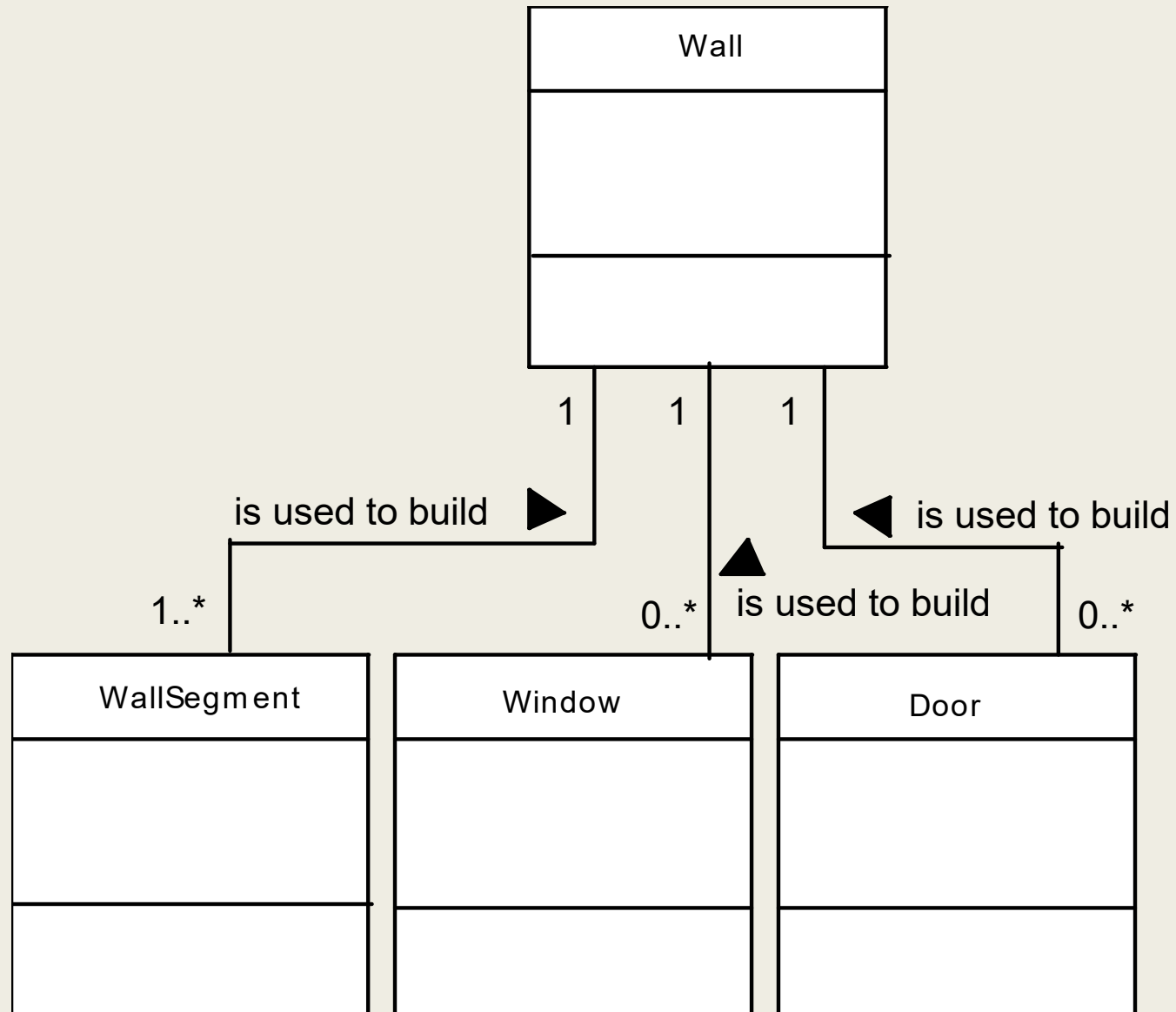
کلاس تجمعی مرکب



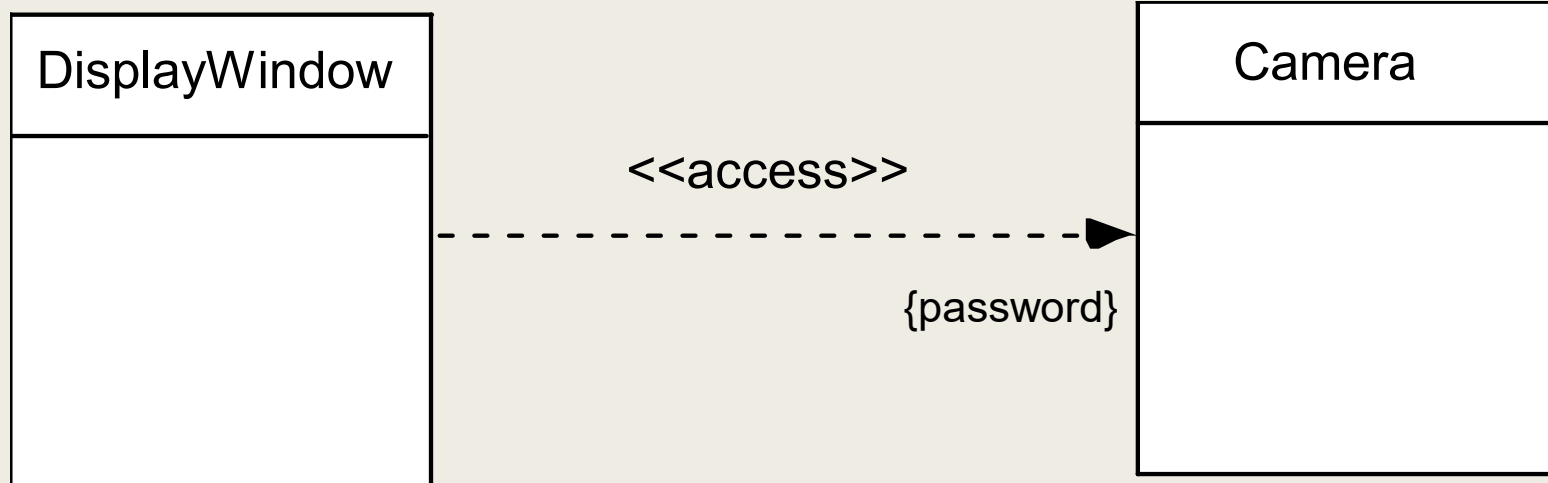
اتحاد ها و وابستگی ها

- دو کلاس تحلیل اغلب به نوعی به یکدیگر متصل هستند
 - در UML این روابط **اتحادها** نامیده می شوند
 - اتحاد ها را می توان با نشان دادن **چندگانگی** (اصطلاح **چندی** (**cardinality**)) در مدل سازی داده ها استفاده می شود
- در بسیاری از موارد، یک رابطه سرویس گیرنده-سرویس دهنده بین دو کلاس تحلیل وجود دارد.
 - در چنین مواردی، کلاس سرویس گیرنده به نوعی به کلاس سرویس دهنده بستگی دارد و یک **رابطه وابستگی** ایجاد می شود

چندگانگی



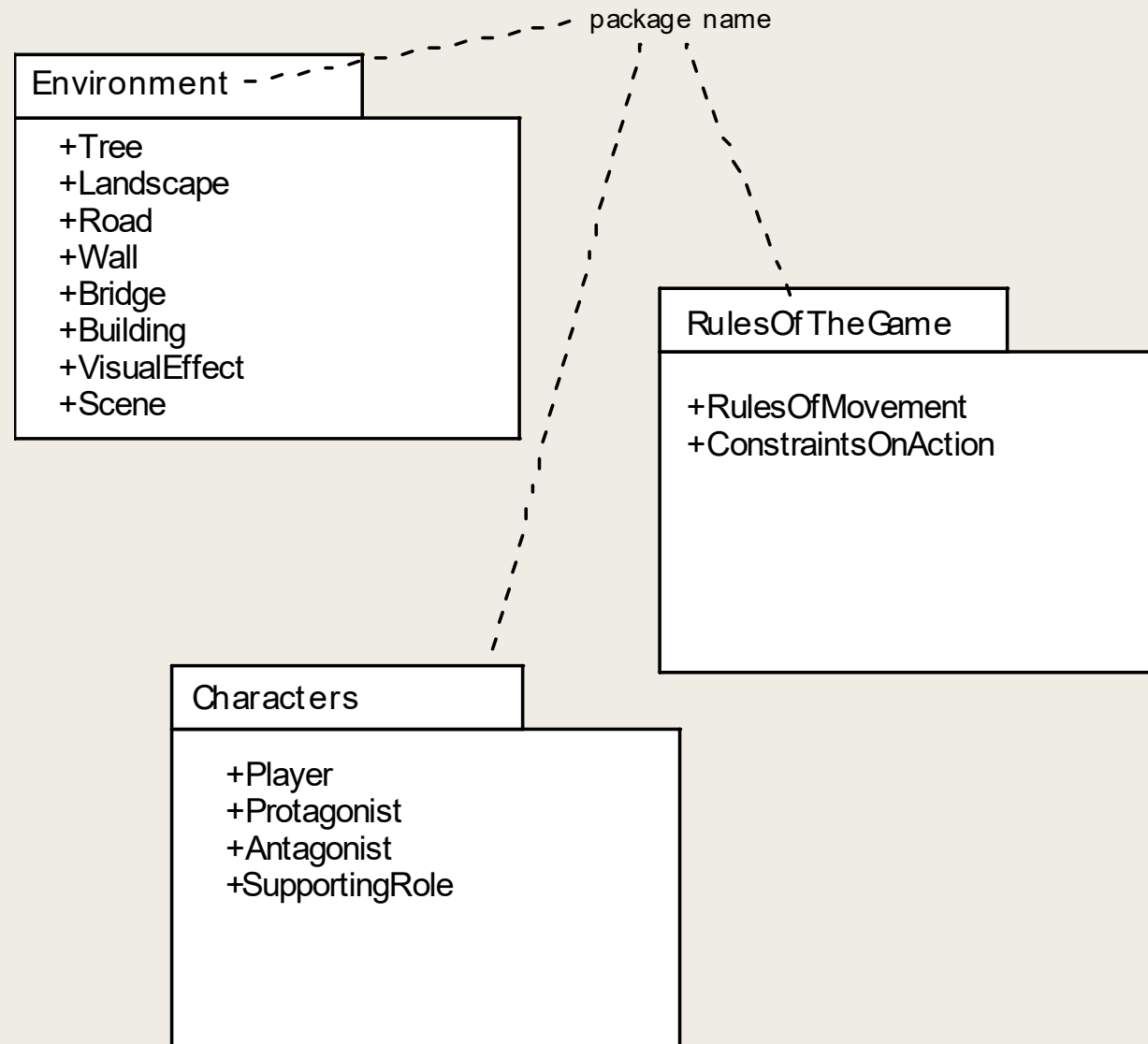
وابستگی‌ها



بسته های تحلیل

- عناصر مختلفی از مدل تحلیل (مانند UseCase ها، کلاس های تحلیل) به شیوه ای دسته بندی می شوند که آنها را به عنوان گروه شده بسته بندی می کند
- علامت مثبت قبل از نام کلاس تحلیل در هر بسته نشان می دهد که کلاس ها دارای دید عمومی و در نتیجه از بسته های دیگر قابل دسترسی هستند.
- علامت های دیگری می توانند قبل یک عنصر از یک بسته باشند. علامت منفی نشان می دهد که یک عنصر از همه بسته های دیگر پنهان شده است و نماد # نشان می دهد که یک عنصر فقط برای بسته های موجود در بسته مفروض در دسترس است.

بسته های تحلیل



بررسی مدل CRC

- به همه شرکت کنندگان در بررسی (مدل CRC) زیر مجموعه ای از کارت های شاخص مدل CRC داده می شوند.
- کارت هایی که همکاری می کنند باید جدا شوند (به عنوان مثال، هیچ بررسی کننده ای نباید دو کارت داشته باشد که همکاری می کنند).
- تمام سناریوهای UseCase (و نمودارهای UseCase مربوطه) باید در طبقه بندی ها سازماندهی شوند.
- سرپرست بررسی به طور عمدی UseCase را می خواند.
- همانطور که سرپرست بررسی به یک شیء خاص می رسد، یک نشانه (Token) را به شخصی که کارت شاخص کلاس مربوطه را دارد منتقل می کند.

بررسی مدل CRC

■ هنگامی که نشانه منتقل می شود، از دارنده کارت کلاس خواسته می شود که مسئولیت های ذکر شده در کارت را شرح دهد.

- گروه تعیین می کند که آیا یکی (یا بیشتر) مسئولیت ها نیازمندی های UseCase را برآورده می کند.

■ اگر مسئولیت ها و همکاری هایی ذکر شده در کارت های شاخص با UseCase تطبیق نیابد، تغییرات در کارت ها صورت می گیرد.

- این ممکن است شامل تعریف کلاس های جدید (و کارت های شاخص CRC مربوطه) یا مشخص کردن مسئولیت ها یا همکاری های جدید یا بازبینی شده در کارت های موجود باشد.