

# فصل ششم – مدل سازی نیازمندیها

(بخش دوم)

---

These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 7/e*, by Roger S. Pressman

Presented by: Hassan Tanabi

Hassan.Tanabi@Gmail.com

## تعریف ویژگیهای کلاس

- ویژگیها کلاسی که برای مدل تحلیل انتخاب شده اند را توصیف می کنند.
- توصیف کلاس در حوزه مسئله
- ساخت دو کلاس متفاوت برای بازیکنان بیس بال:
  - **برای بازیکنان نرم افزار آماری:**  
نام، پست، تعداد سال بازی، تعداد بازیها، ...
  - **برای نرم افزار حقوق و دستمزد:**  
میانگین حقوق دریافتی، حقوق بازنشستگی، ...

## تعریف کارها

- کارها رفتار یک شی را مشخص می کنند.
- برای مشخص کردن کارها مجددا سناریو را بررسی و این بار بدنبال **افعال** می گردیم.



verbs

- کارها را می توان به چهار دسته تقسیم نمود:
  1. کارهایی که داده ها را به طریق مختلفی تغییر می دهند. مانند: جمع، حذف، ...
  2. کارهایی که پردازشی را مهیا می کنند.
  3. کارهایی که وضعیت یک شی را مورد بررسی قرار می دهند.
  4. کارهایی که یک شی را به منظور کنترل اتفاق افتادن یک رویداد مانیتور می کنند.

## مدل های CRC

- تشخیص و سازماندهی کلاس هایی که مرتبط با سیستم یا نیازمندیهای آن هستند.
- یک مدل CRC در واقع مجموعه ای از کارت های اندیس گذاری شده استاندارد است که کلاس ها را نمایش می دهد. هر کارت به سه قسمت تقسیم می شود، بالای کارت نام کلاس. در میانه آن سمت چپ وظایف کلاس و در سمت راست کارت شریک های آن کلاس

By: *Amblar*

<b>Class: FloorPlan</b>	
Description:	
<b>Responsibility:</b>	<b>Collaborator:</b>
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 ها

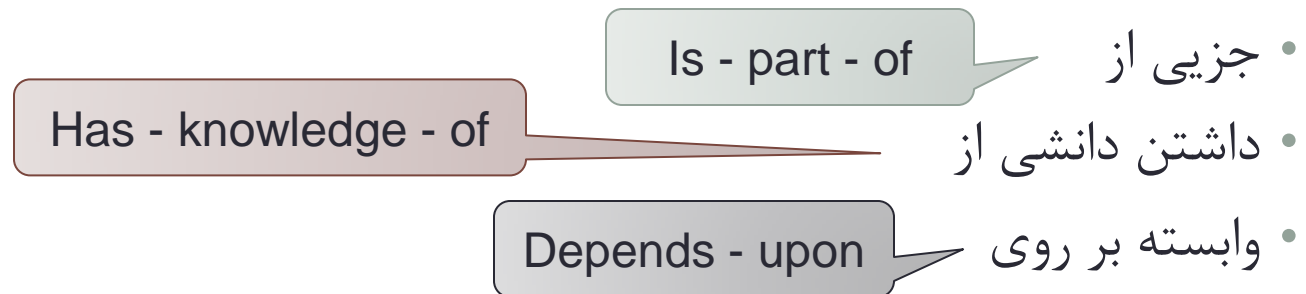
- **کلاس های موجودیت**، که کلاس های مدل یا تجارت هم نامیده می شوند.
- مستقیماً از بیان مسئله استخراج می شوند. مانند: حس گر، نقشه طبقه، ...
- **کلاس های مرزی**، برای ایجاد واسطی استفاده می شوند که کاربر آنها را می بیند یا برای استفاده از سیستم با آنها تعامل دارد.
- **کلاس های کنترلی**، که یک “واحد از کار” را از ابتدا تا انتها مدیریت می کند.

## وظایف

- هوشمندی سیستم باید در میان کلاس ها توزیع شود.
  - به منظور ارضای نیازمندیها
- هر وظیفه تا حد ممکن باید کلی بیان شود.
- اطلاعات و رفتار مرتبط با آن باید در همان کلاس ذکر شوند.
  - برای رسیدن به هدف کپسوله سازی در شی گرایي
- اطلاعات مربوط به یک چیز باید در یک کلاس ذکر شود و نباید در میان چندین کلاس توزیع شود.
  - در غیر این صورت نگهداری و تست نرم افزار دشوار می شود.
- وظایف در مواقع نیاز باید در میان کلاس های مرتبط به اشتراک گذاشته شود.
  - بروزرسانی و نمایش حرکت در یک بازی

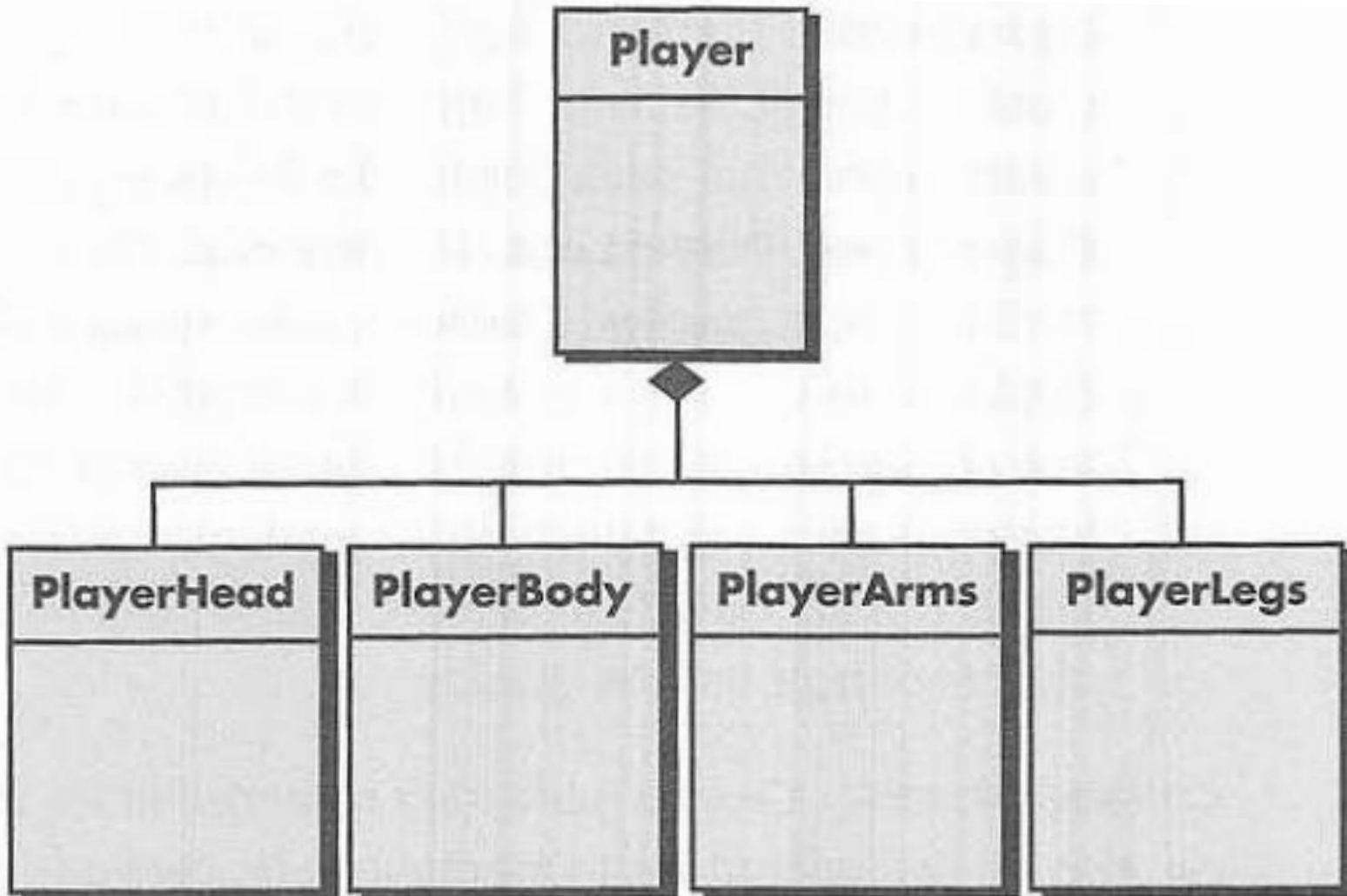
## همکاری ها

- کلاس ها از دو طریق وظایف خود را انجام می دهند:
- یک کلاس می تواند با استفاده از کارهای خود ویژگیهای خود را تغییر دهد
- و یا می تواند با دیگر کلاس ها همکاری داشته باشد.
- همکاری ها وابستگی های بین کلاس ها را مشخص می کند.
- همکاری ها مشخص می کند که آیا یک کلاس به تنهایی از عهده وظیفه اش بر می آید یا نه
- سه نوع وابستگی کلی بین کلاس ها وجود دارد:





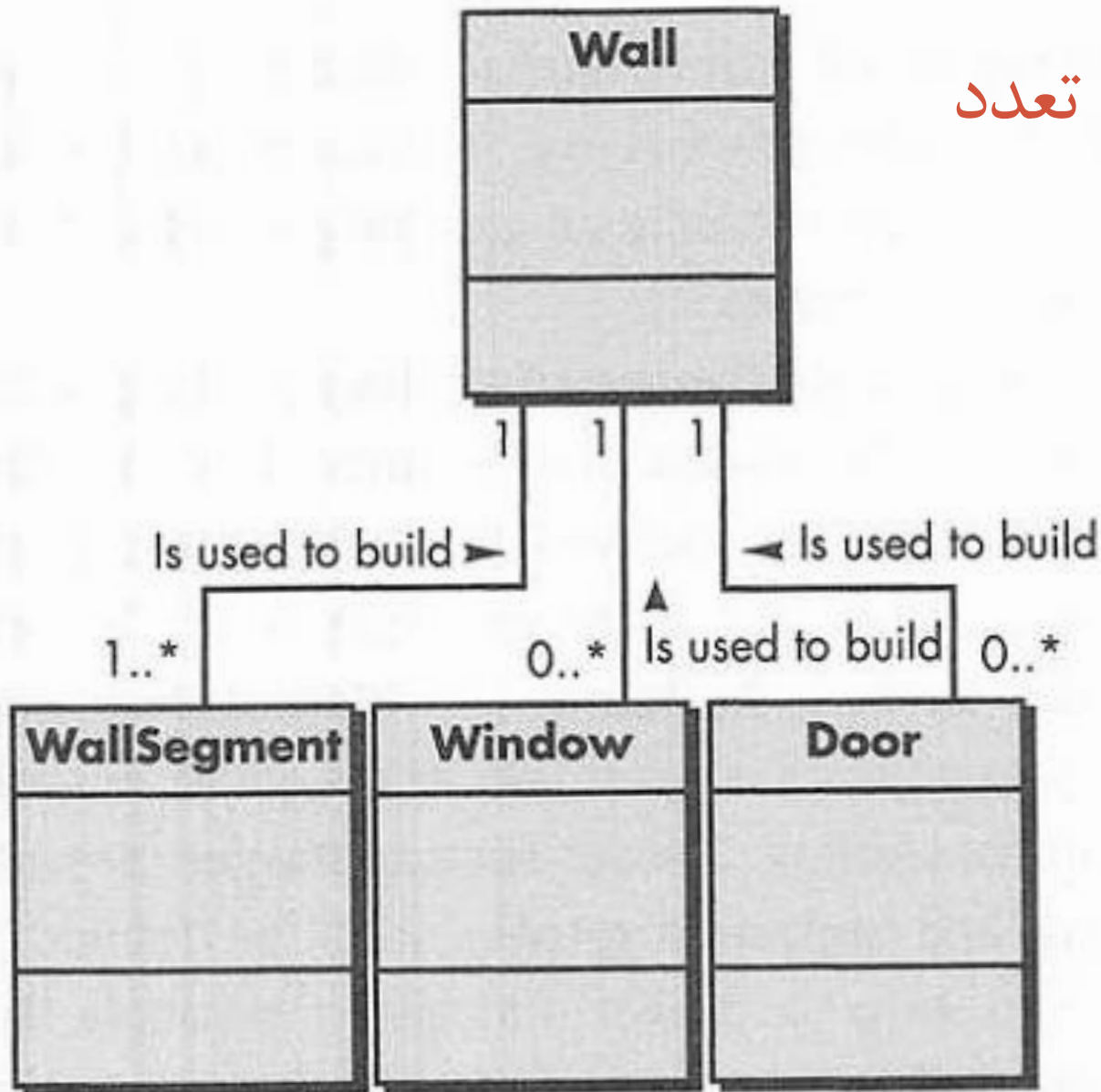
## نمونه ای از رابطه تجمیع



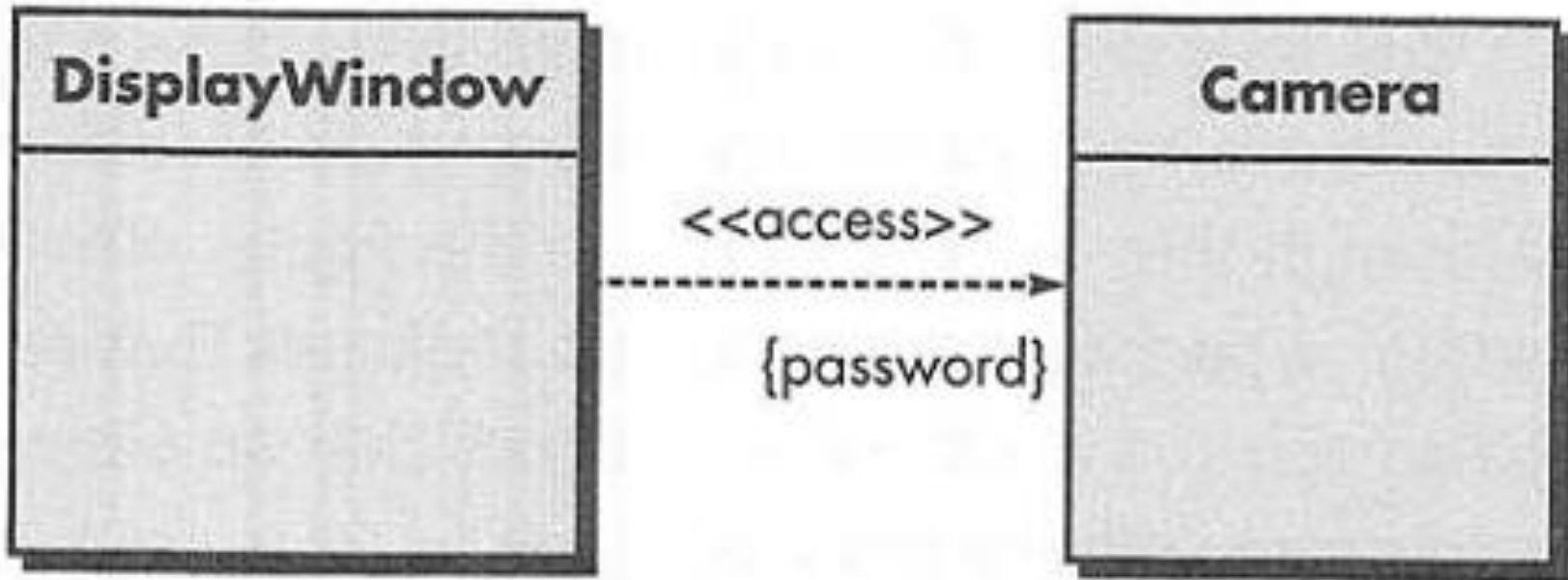
## روابط انجمنی و وابستگی ها

- دو کلاس تحلیل به صورت های مختلفی با یکدیگر در ارتباط هستند
  - در UML این ارتباط **انجمنی** نامیده می شود
  - این مفهوم با در نظر گرفتن **تعدد** مجددا تعریف می شود
- در خیلی از موارد، روابط **Client-Server** بین دو کلاس تحلیل وجود دارد.
- در این حالت ها یک کلاس سرویس گیرنده وابسته به سرویس دهنده می باشد و رابطه **وابستگی** ایجاد می شود.

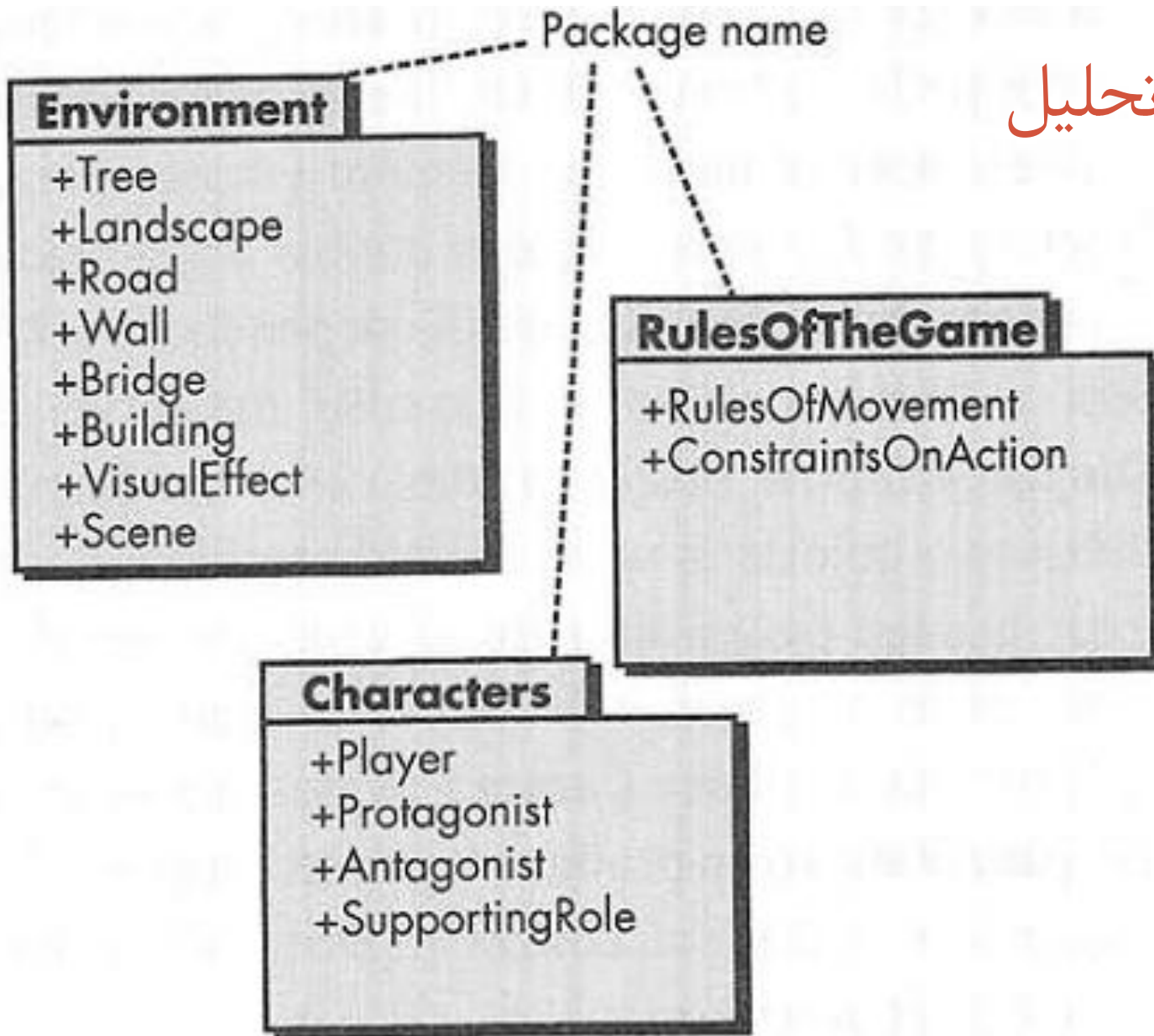
## رابطه انجمنی و تعدد



## رابطه وابستگی



## بسته های تحلیل



خسته نباشید

---

