

بسمه تعالی

درس مهندسی نرم افزار پیشرفته

فصل یازدهم

بسته‌ها، مدلسازی مولفه‌ها، و مدلسازی استقرار

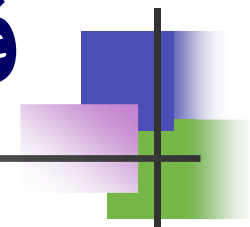
دکتر فریدون شمس

اهداف جلسه



- آشنائی با **بسته‌ها** و مفاهیم اساسی آن
- آشنائی با مفاهیم اساسی و روش ایجاد **نمودار مولفه‌ها**
- آشنائی با مفاهیم اساسی و روش ایجاد **نمودار استقرار**

فهرست مطالب

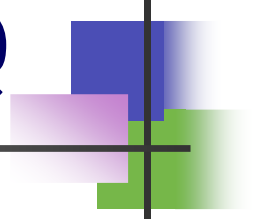


■ بسته‌ها

■ مدلسازی مولفه‌ها

■ مدلسازی استقرار

بسته‌ها



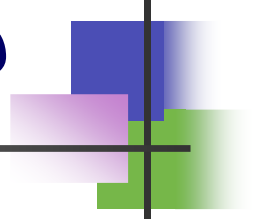
- هنگام مدلسازی سیستم‌های نرم‌افزاری بزرگ به روش شی‌گرا، معمولاً با مدل‌هایی که حاوی تعداد بسیار زیادی از عناصر مدلسازی مانند کلاس‌ها هستند، روبرو خواهیم شد
- درک و مدیریت صحیح مدل‌ها مستلزم سازماندهی عناصر این مدل‌ها

در گروه‌های بزرگتر است

مفهوم بسته‌ها



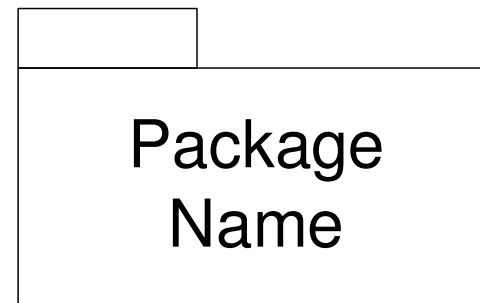
مفاهیم بسته‌ها



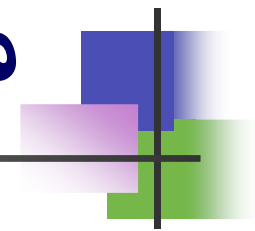
■ بسته (*Package*)

■ مکانیزمی عمومی برای سازماندهی عناصر مدلسازی در گروه‌های بزرگتر

کلاس‌ها، واسط‌ها، مولفه‌ها،
گره‌ها، تعامل‌ها، موارد
کاربری، نمودارها و
بسته‌های دیگر



مفاهیم بسته‌ها (ادامه)



■ قابلیت دید (دستیابی) (*Visibility*)

■ عمومی (*Public*)

■ محافظت شده (*Protected*)

■ خصوصی (*Private*)

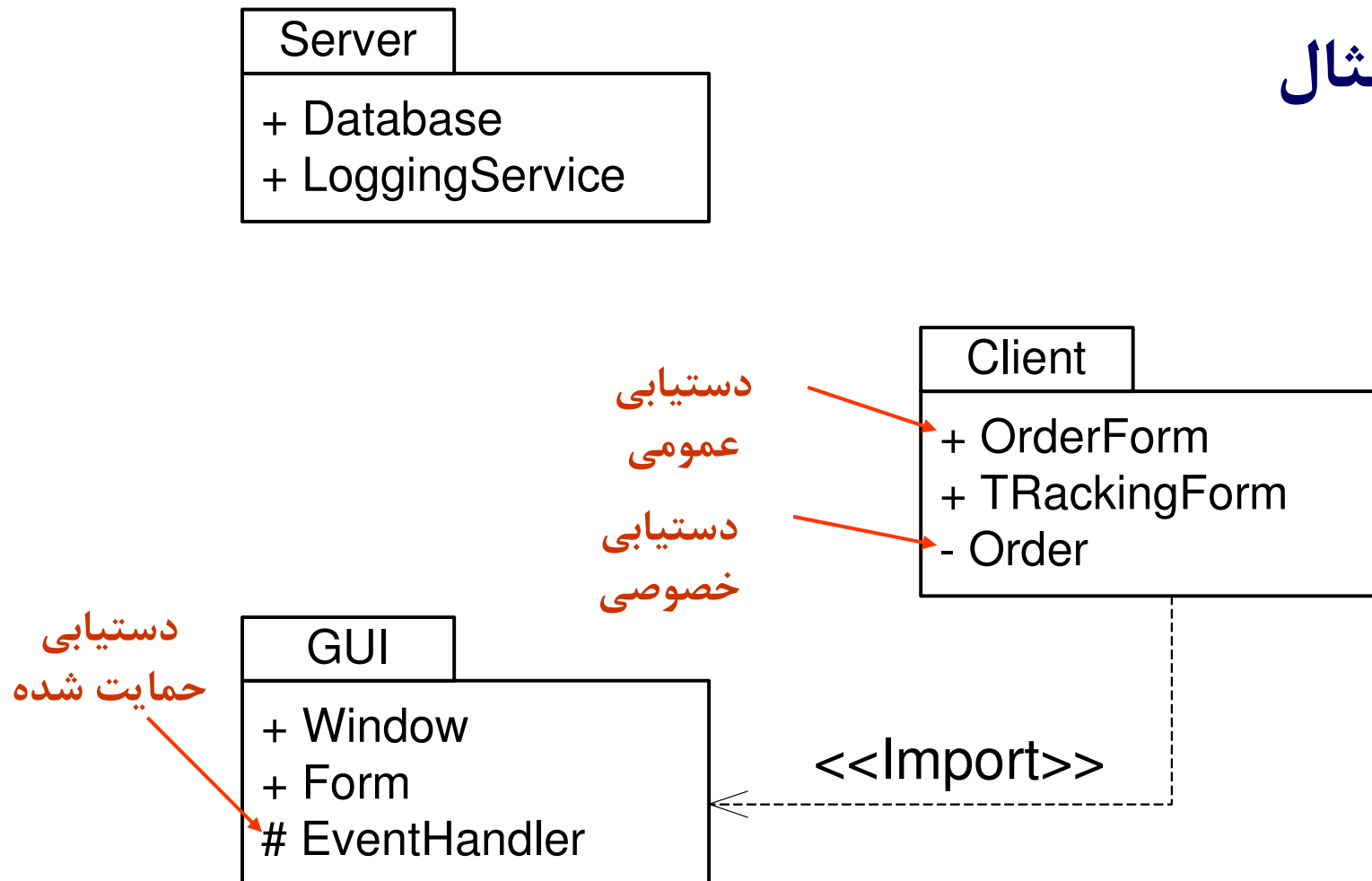
■ وارد کردن و صادر کردن (*Importing & Exporting*)

■ فرض کنید دو کلاس الف و ب در دو بسته جداگانه وجود داشته باشند و هرکدام قابلیت دید «عمومی» را دارند.

■ برای اینکه کلاس الف به کلاس ب دسترسی داشته باشد، باید بسته حاوی کلاس ب توسط بسته حاوی کلاس الف «وارد» شود

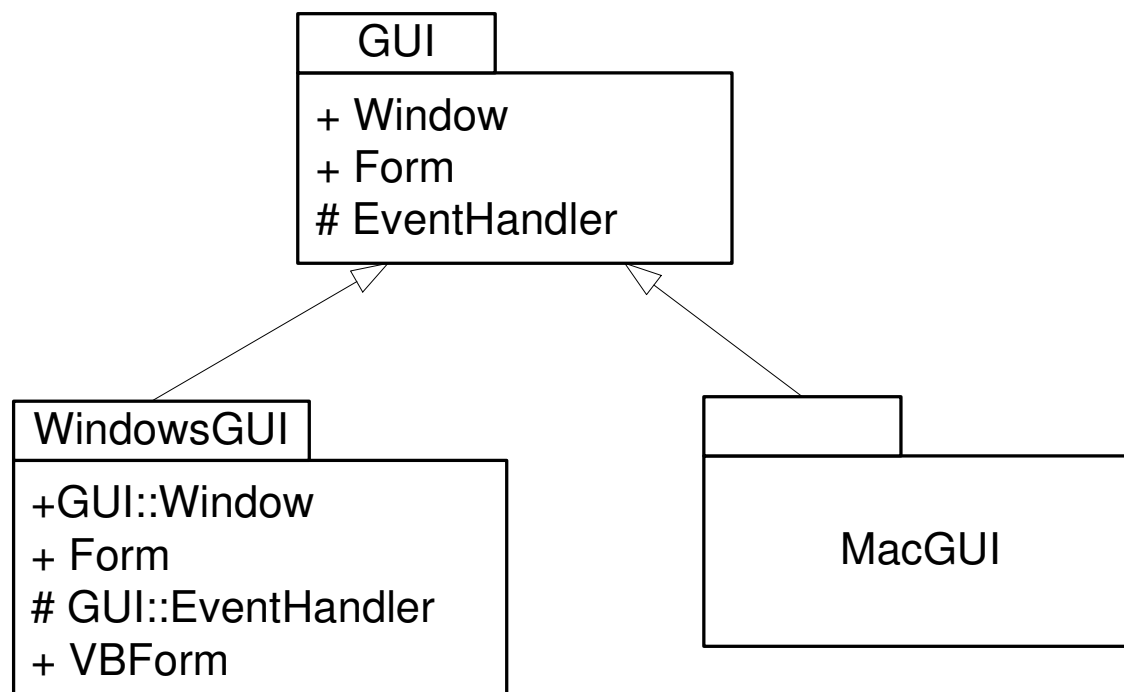
مفاهیم بسته‌ها (ادامه)

مثال

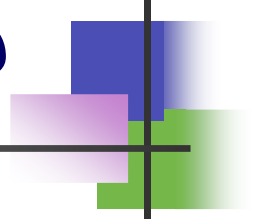


مفاهیم بسته‌ها (ادامه)

رابطه عام/خاص



موارد استفاده از بسته‌ها



(۱) مدلسازی گروه‌هایی از عناصر

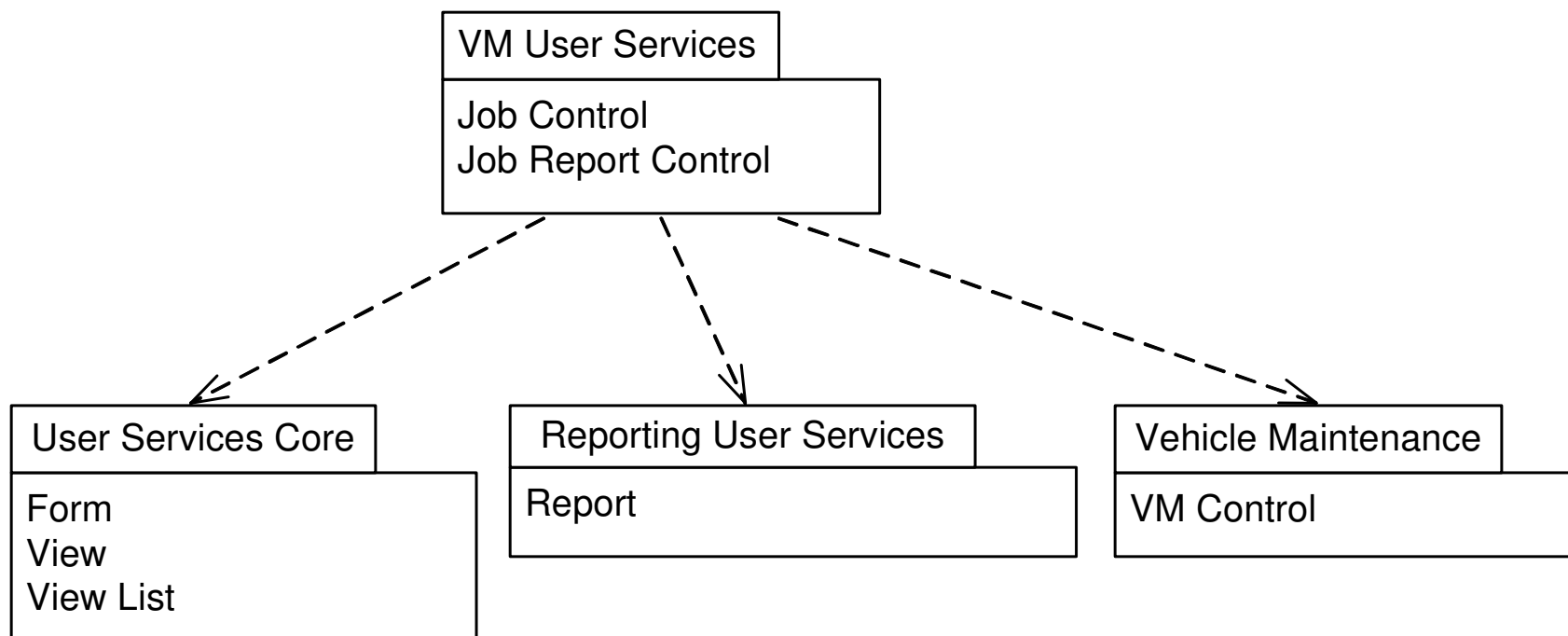
- رایجترین استفاده از بسته‌ها در گروه‌بندی عناصر مدلسازی منطقاً مرتبط است به صورتیکه بتوان با هر بسته بعنوان **یک واحد** رفتار کرد
- بسته‌ها با کلاس‌ها تفاوت دارند

(۲) مدلسازی دیدهای معماری

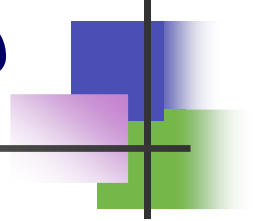
- نمایش عناصر موجود در دید معماری
 - مانند نمایش لایه
- نمایش برخی دیدهای معماری توسط بسته‌ها آسان‌تر خواهد شد
 - مانند دید لایه‌بندی

موارد استفاده از بسته‌ها (ادامه)

مثال: تعمیرگاه



مدلسازی مولفه‌ها



■ در سیستم‌های

■ بسیار ساده: نیاز به مدلسازی منطقی وجود ندارد

■ واقعی: نقش مدلسازی منطقی بسیار کلیدی است

در UML موجودیت‌های فیزیکی سیستم بوسیله مفهوم مولفه مدلسازی می‌شوند

«چگونه می‌توان یک سیستم بزرگ را به سیستم‌های کوچکتر تقسیم کرد؟»

متدولوژی‌های OO مطرح می‌کنند



مفهوم کلاس

مدلسازی مولفه‌ها (ادامه)



■ ولی بنظر می‌رسد که

■ در سیستم‌های نرم‌افزاری، بسیاری از مفاهیم کلاس به عنوان

بدین علت برنامه‌نویسی شی‌گرا (OOP)، تا حدودی، در مجسم کردن
ایده مدارات مجتمع (IC) نرم‌افزاری **ناموفق** عمل کرده است

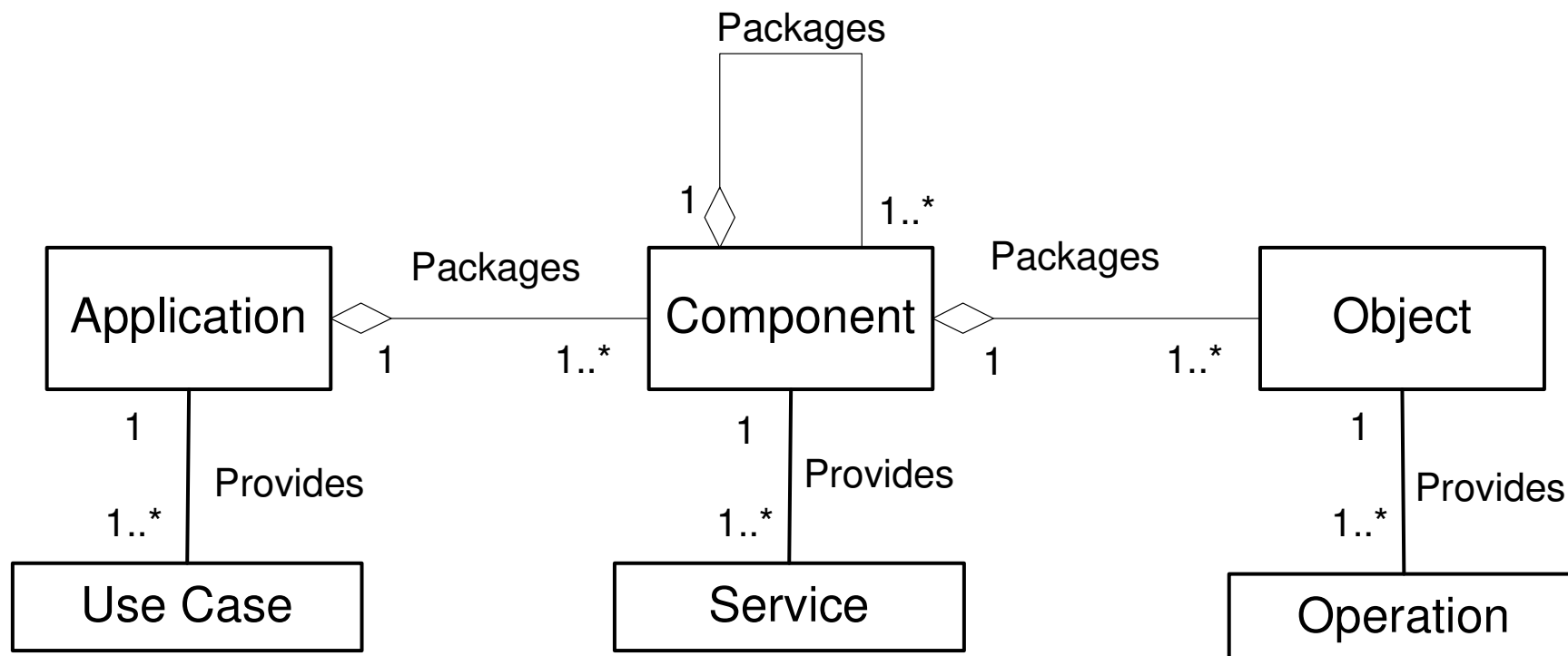
■ مکانیزم‌های استفاده مجدد که در زبان‌های برنامه‌سازی شی‌گرا مطرح

است **محدود** به برنامه‌های خود زبان می‌باشند

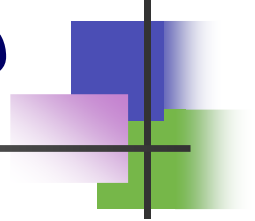
مدلسازی مولفه‌ها (ادامه)



■ مثال: رابطه میان موارد کاربری، مولفه‌ها و اشیاء تشکیل دهنده یک برنامه کاربردی



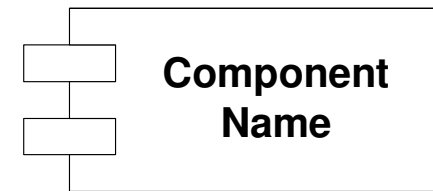
مفاهیم مدل‌سازی مولفه‌ها



■ مولفه (Components)

- یک قطعه پیچیده، تقریباً مستقل و جایگزین‌پذیر از سیستم که دسترسی به آن تنها بوسیله یک واسطه خوش‌تعریف امکان‌پذیر است.

تکنولوژی‌های EJB،
COM+، .NET و CORBA
یکی از مصادیق این تعریف
هستند

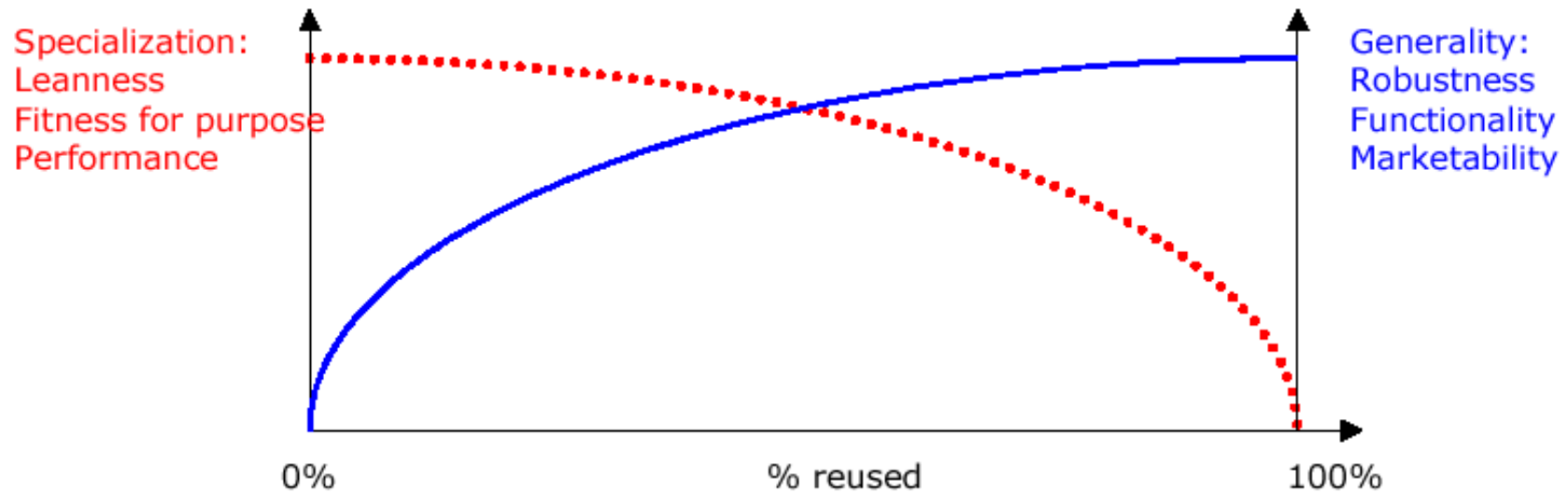


اهداف مدلسازی مولفه‌ها



■ هدف تکنولوژی مولفه‌ها

■ رسیدن به یک ترکیب متعادل از استفاده مجدد (*Reuse*) و استفاده (*Use*)

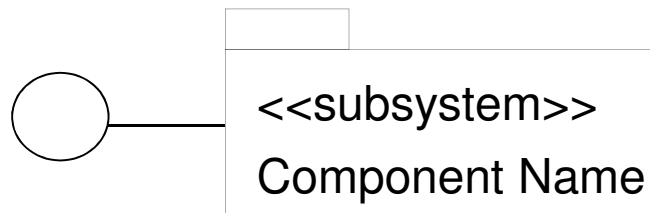


ویژگیهای مدلسازی مولفه‌ها

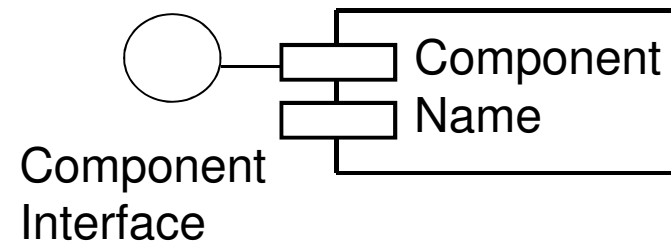
■ ساختار پیاده‌سازی

- بواسطه آن یک یا چند واسط پیاده‌سازی می‌گردند
- زیرسیستم‌ها (*Subsystems*) در طراحی به مولفه‌ها در پیاده‌سازی تبدیل می‌گردند

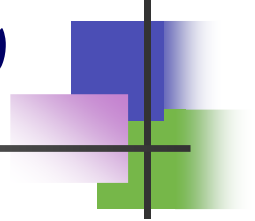
Design Model



Implementation Model



شی‌گرا و مولفه‌گرا



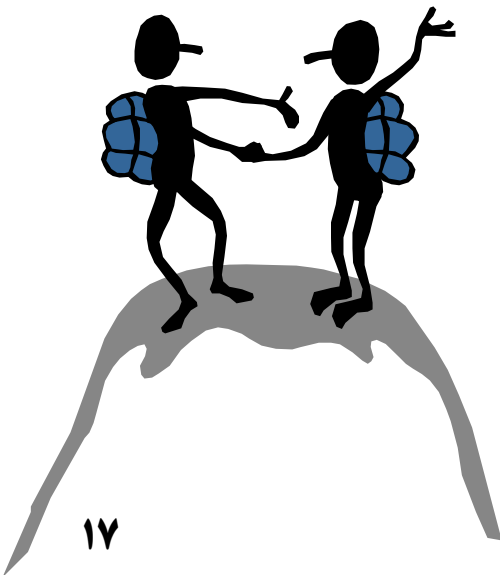
■ OOP در مقابل CBD

■ هدف OOP ایجاد یک سیستم نرم‌افزاری متشکل از اشیاء مرتبط است،

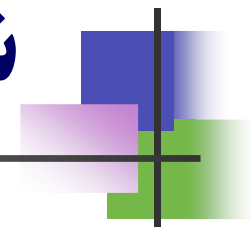
■ در حالیکه

■ هدف *Component-based Development* ایجاد یک سیستم متشکل از

تعدادی مولفه مرتبط است



شی‌گرا و مولفه‌گرا (ادامه)



■ ارتباط اشیاء با مولفه‌ها

■ هر مولفه در پایین‌ترین سطح، شامل مجموعه‌ای از اشیاء است

■ ویژگیهای مشترک

■ دارای اسم هستند

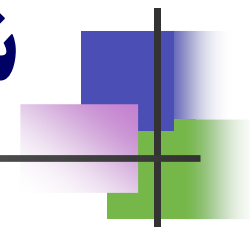
■ می‌توانند مجموعه‌ای از واسط‌ها را پیاده‌سازی نمایند

■ می‌توانند در روابط وابستگی، تعمیم‌پذیری و تجمعی شرکت نمایند

■ قابلیت تو در تو بودن دارند

■ می‌توانند در تعاملات شرکت کنند

شی گرا و مولفه گرا (ادامه)



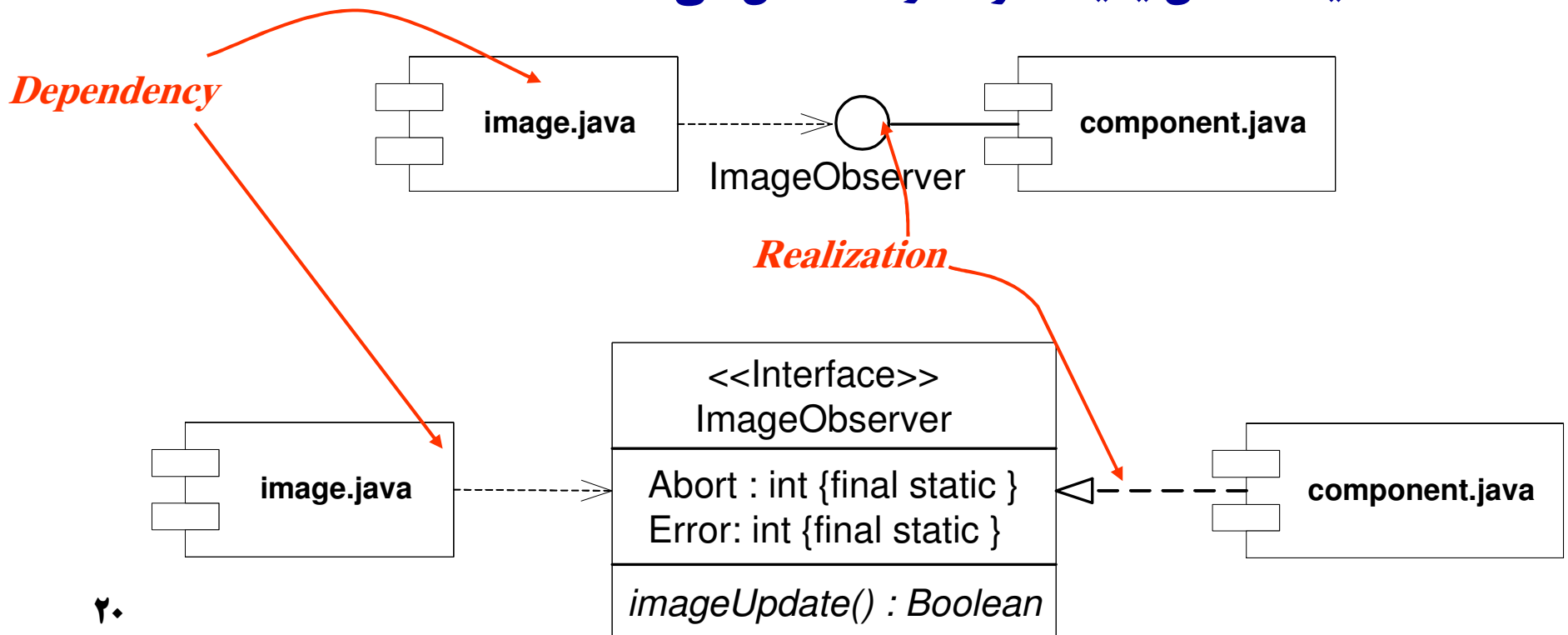
■ تفاوت‌ها

- کلاس‌ها ساختار منطقی سیستم را تشکیل داده در حالیکه مولفه‌ها ساختار فیزیکی سیستم را تشکیل می‌دهند
- مولفه‌ها سطح تجرید بالاتری از کلاس‌ها را نمایش می‌دهند
- یک کلاس می‌تواند دربردارنده صفات و اعمالی که برای سرویس گیرندگان آن، به صورت مستقیم دسترسی پذیرند باشد، در حالیکه ساختار داخلی یک مولفه کاملاً کپسوله است

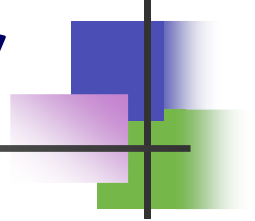
واسط در مدل سازی مولفه ها

■ واسط (Interface)

■ عبارتست از مجموعه ای از اعمال که سرویس های ارائه شده بوسیله یک کلاس یا یک مولفه را مشخص می کند



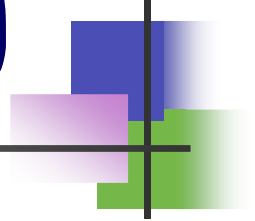
جایگزینی در مولفه‌ها



■ جایگزین‌پذیری باینری (*Binary Replaceability*)

- هدف بیشتر ابزارهای مبتنی بر مولفه‌ها فراهم نمودن محیطی است که در آن یک سیستم نرم‌افزاری بوسیله تجمیع تعدادی از **قطعات** باینری **جایگزین‌پذیر** ایجاد گردد

انواع مولفه و سازماندهی آنها



■ انواع مولفه‌ها

■ مولفه‌های استقرار (*Deployment Components*)

■ مولفه‌های محصول کار (*Work Product Components*)

■ مولفه‌های اجرایی (*Execution Components*)

■ مولفه‌ها را می‌توان در بسته‌ها گروه‌بندی نمود

■ در این صورت، از روابط وابستگی، تعمیم‌پذیری و تجمعی میان مولفه‌ها

استفاده می‌شود

روش‌های مدل‌سازی مولفه‌ها



■ تکنیک‌های مدل‌سازی

- مدل‌سازی فایل‌های اجرایی و کتابخانه‌ای
- مدل‌سازی جداول پایگاه داده‌ها، فایل‌های داده‌ای و مستندات
- مدل‌سازی API‌ها استفاده شده
- مدل‌سازی کد منبع

روش‌های مدلسازی مولفه‌ها (ادامه)

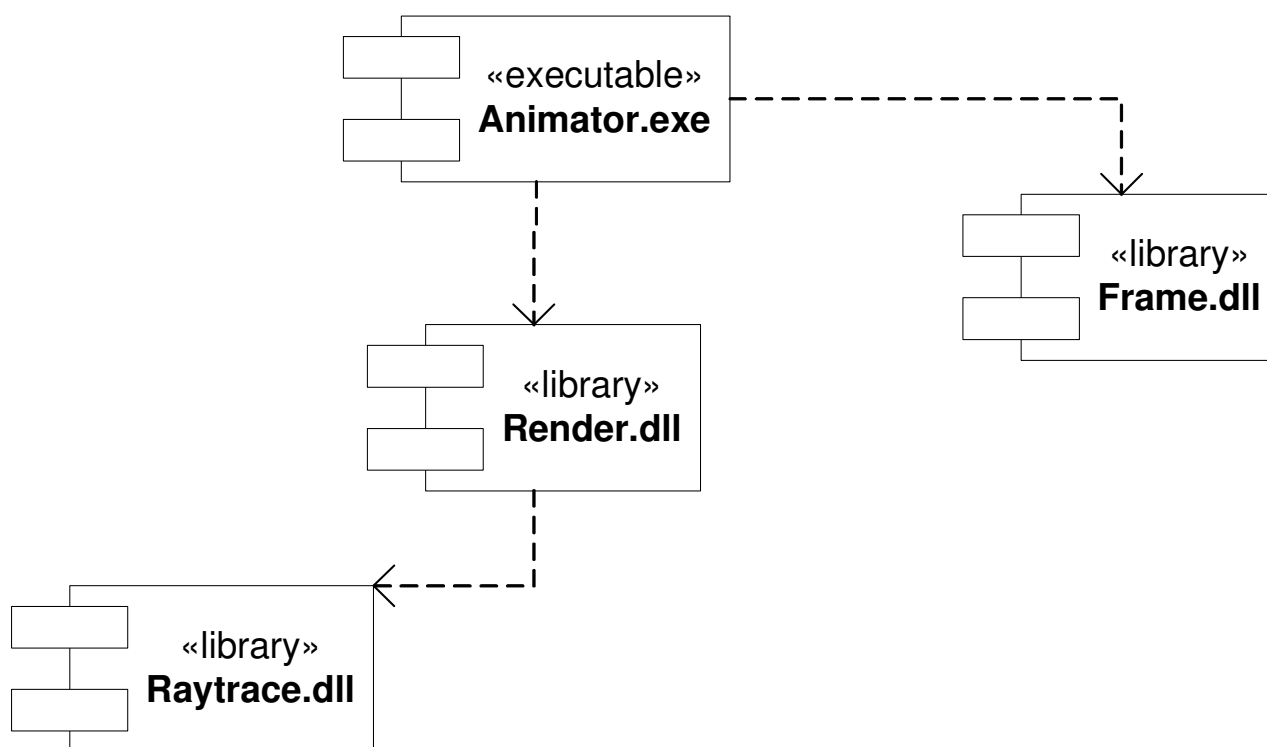


■ مدلسازی فایل‌های اجرایی و کتابخانه‌ای

- نحوه تقسیم‌بندی سیستم فیزیکی را با توجه به فاکتورهای تکنیکی، مدیریت پیکربندی و قابلیت استفاده مجدد را مشخص نمایید
- هر مولفه اجرایی یا کتابخانه را به صورت یک مولفه مدلسازی نمایید
- در صورت لزوم واسط‌های کلیدی سیستم را معین کنید
- روابط موجود بین مولفه‌های اجرایی، کتابخانه‌ها و واسط‌ها را مدلسازی نمایید

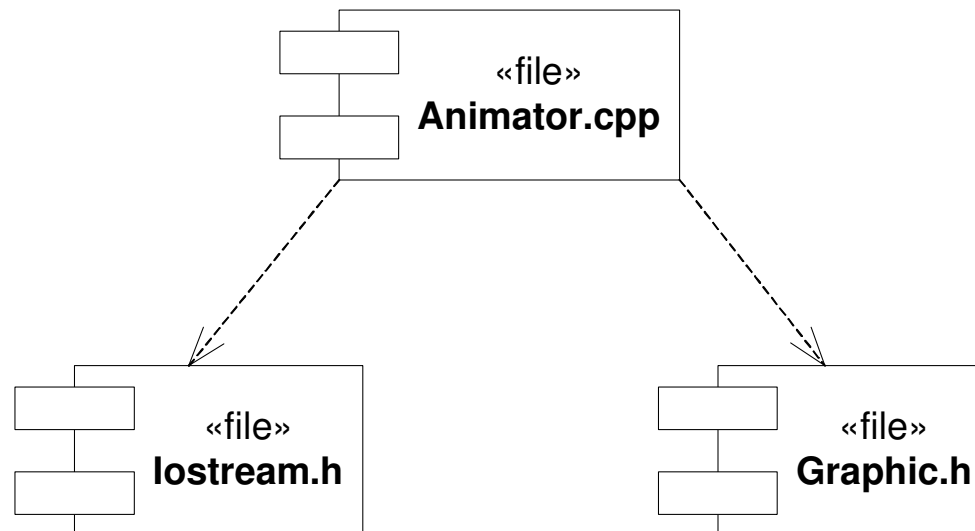
روش‌های مدلسازی مولفه‌ها (ادامه)

مثال:



روش‌های مدلسازی مولفه‌ها (ادامه)

مثال:



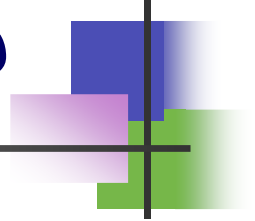
مدلسازی استقرار



■ در UML نداشت مولفه‌های نرم‌افزاری به مولفه‌های سخت‌افزاری و ارتباطات میان مولفه‌های سخت‌افزاری در سیستم مورد نظر بوسیله **نمودار استقرار** نمایش داده می‌شود

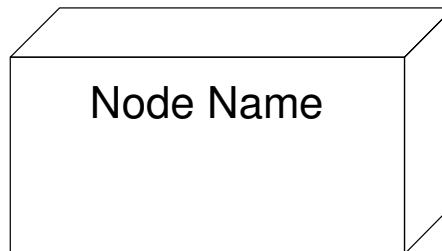


مفاهیم مدل سازی استقرار



گره (*Node*) ■

- عنصر فیزیکی که در زمان اجرا وجود داشته و یک منبع محاسباتی را نمایش داده و به صورت کلی دارای حافظه و قابلیت پردازش است



مفاهیم مدلسازی استقرار (ادامه)



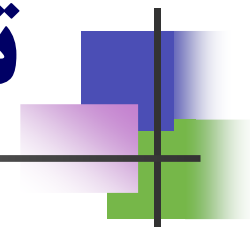
■ سازماندهی گره‌ها

- می‌توان گره‌ها را در بسته‌ها گروه‌بندی نمود
- می‌توان از روابط وابستگی، تعمیم‌پذیری و تجمعی میان آنها استفاده کرد

■ روابط

- رایجترین رابطه میان گره‌ها، رابطه انجمنی است که معمولاً نمایش دهنده یک ارتباط فیزیکی میان گره‌ها مانند ارتباط *Ethernet*، گذرگاه مشترک یا حتی پیوند ماهواره‌ای باشد

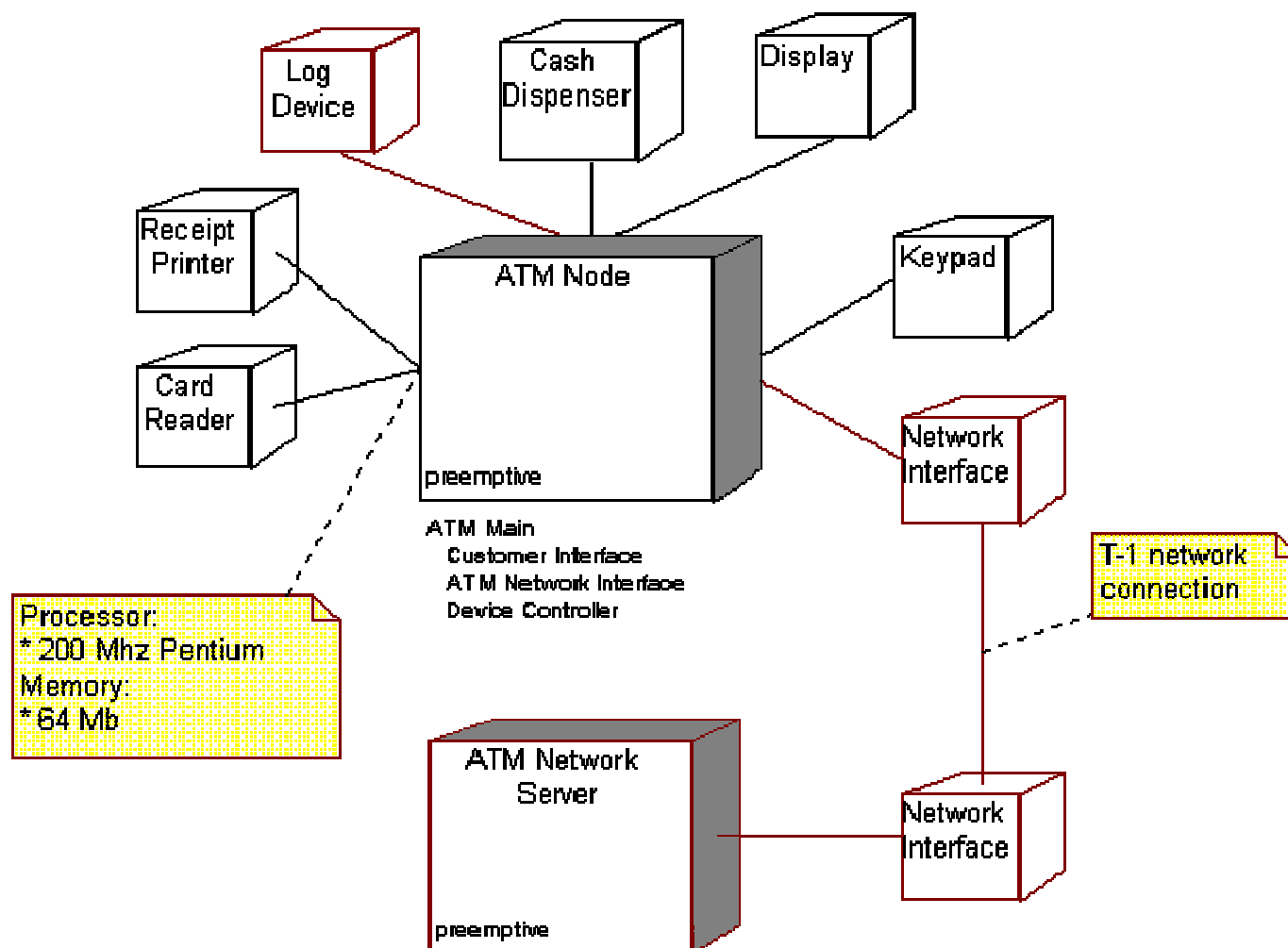
تکنیک‌های مدلسازی استقرار



■ مدلسازی پردازشگرها و دستگاه‌ها

- هرکدام از منابع محاسباتی سخت‌افزاری که دید استقرار را تشکیل می‌دهند به صورت یک **گره** مدلسازی نمایند
- اگر این منابع، پردازشگرها و یا دستگاه‌هایی مخصوصی باشند که جزئی از واژگان سیستم مورد نظر را تشکیل می‌دهند، از مکانیزم **مقوله‌بندی** برای بیان آنها استفاده نمایند
- صفات و اعمال هر گره را مشخص نمایند

مدلسازی استقرار - نمونه





پرسش و پاسخ