

بسمه تعالی

## فصل سوم

# آشنائی با مفاهیم اولیه شی گرائی

# اهداف جلسه

- آشنائی با مفاهیم اساسی شیء گرائی
- آشنائی با انواع روابط بین کلاسها

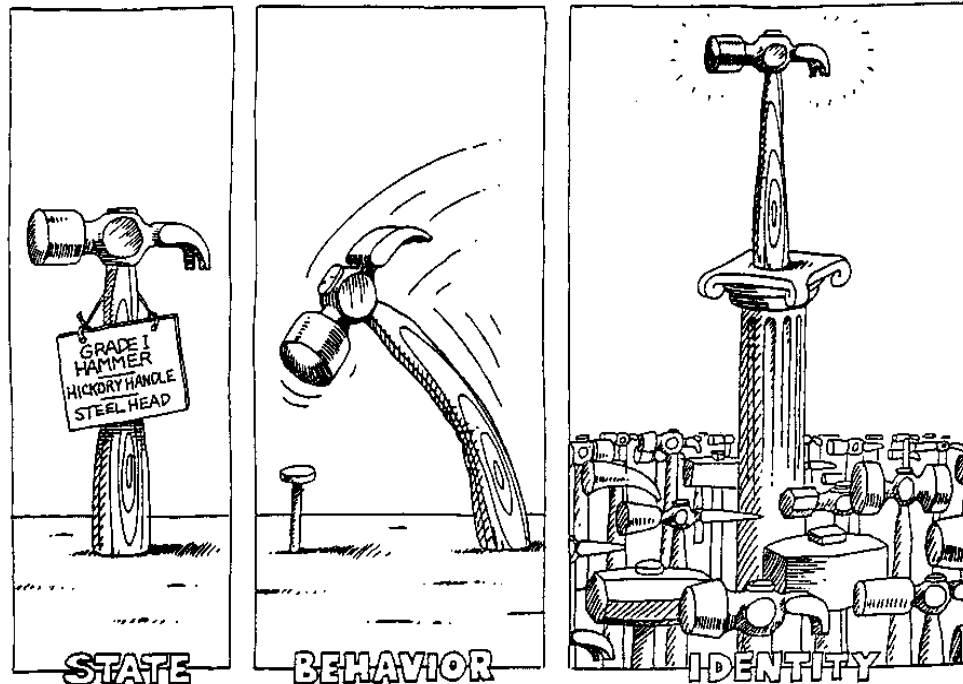
# فهرست مطالب



- شیء
- تعریف کلاس
- مکانیزم ارتباط بین اشیاء
- رابطه بین کلاس‌ها

# شیء (Object)

یک مفهوم کلی است بگونه‌ای که دارای **هویت** بوده و قادر به بروز **رفتار** و **ثبات حالات** (وضعیت) خود باشد



STATE

BEHAVIOR

IDENTITY

حالت

رفتار

هویت

# شی (ادامه)

## ■ هویت (*Identity*)

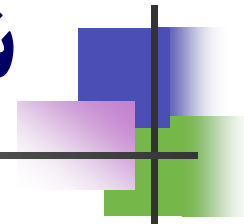
■ آن ویژگی از یک شیء است که آنرا از بقیه اشیاء متمایز می‌سازد

## ■ حالت (*State*)

■ حالت یک شیء در بردارنده تمام خواص آن شیء و مقادیر جاری آنها است

## ■ رفتار (*Behavior*)

■ چگونگی عمل و عکس‌العمل یک شیء در مقابل دریافت و یا ارسال پیام دارد



## ■ مثال‌هایی از اشیاء

- موجودیت‌های خارجی: سیستم‌های برونی، دستگاه‌ها و مردم تولیدکننده یا مصرف‌کننده اطلاعات، ...
- موجودیت‌های اطلاعاتی: انواع فایل‌ها، مستندات، ...
- مفاهیم منطقی: دانشگاه، کلاس درس، دوره آموزشی، ...
- نقش‌های انسانی: مشتری، تحلیلگر، معمار نرم‌افزار، ...

# شی (ادامه)

## ■ مثال‌هایی از اشیاء

- واحدهای سازمانی: اداره آموزش، امور مالی، ...
- مکان‌های فیزیکی: لابراتور زبان، آزمایشگاه شیمی، ...
- ابزارآلات و اسباب: موتورسیکلت، انبردست، صندلی، ...

براساس تعریف قبل مفاهیمی مانند **عمل**، **صفت** و **زمان** نمونه‌هایی از

موجودیت‌هایی هستند که یک شیء محسوب نمی‌شوند

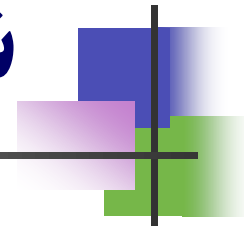
# شی (ادامه)

## ■ مثال ۱: کتاب الف

- هویت: کتاب
- حالت: اطلاعات فهرست‌نویسی، مکان نگهداری فیزیکی، وضعیت فعلی (امانت/ارزرو/آزاد)، ...
- رفتار: ثبت اطلاعات کتاب، جستجو، سفارش دادن برای خرید، ...

الگوریتم‌ها + ساختمان داده‌ها = شیء





## مثال ۲: درس‌های رشته کامپیوتر

Intro to OO (180)

Operating Systems (101)

Data Structures(200)

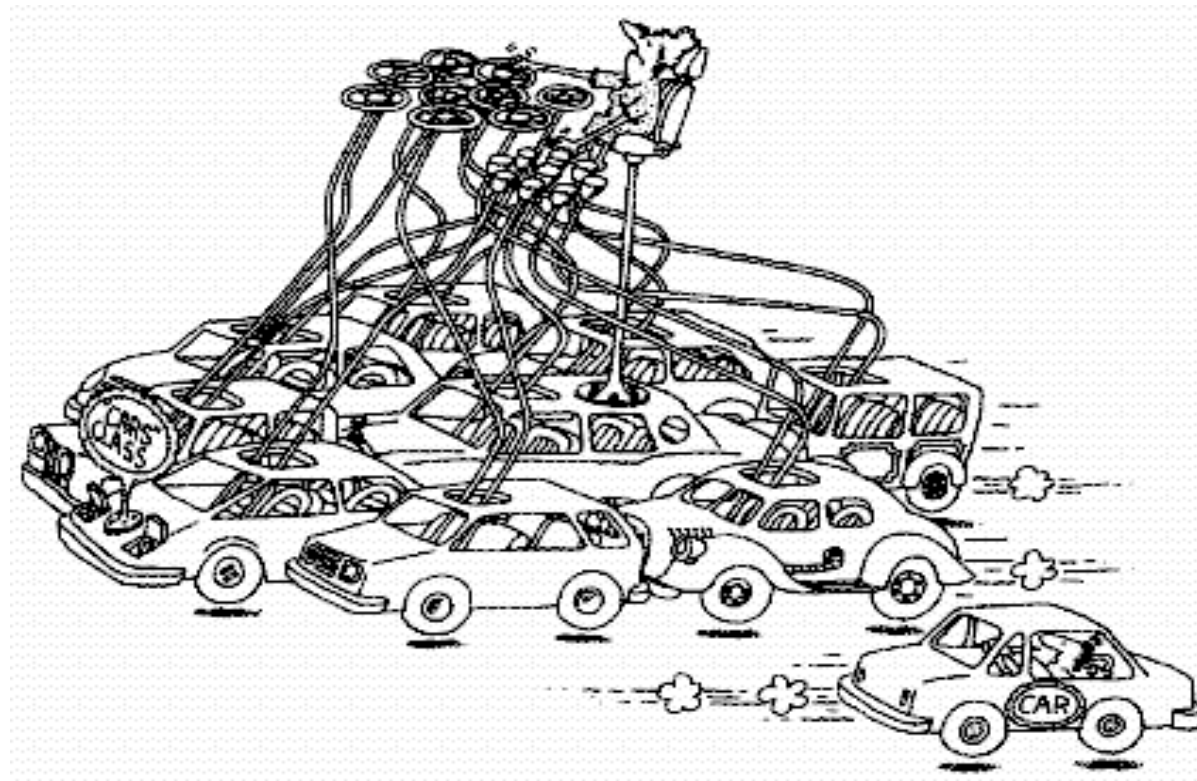
Software Engineering(110)

Database Design (110)

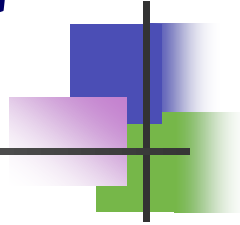
Software Engineering(110)

# کلاس (Class)

مجموعه‌ای از اشیاء که دارای ساختار و رفتار مشترک



# کلاس (ادامه)

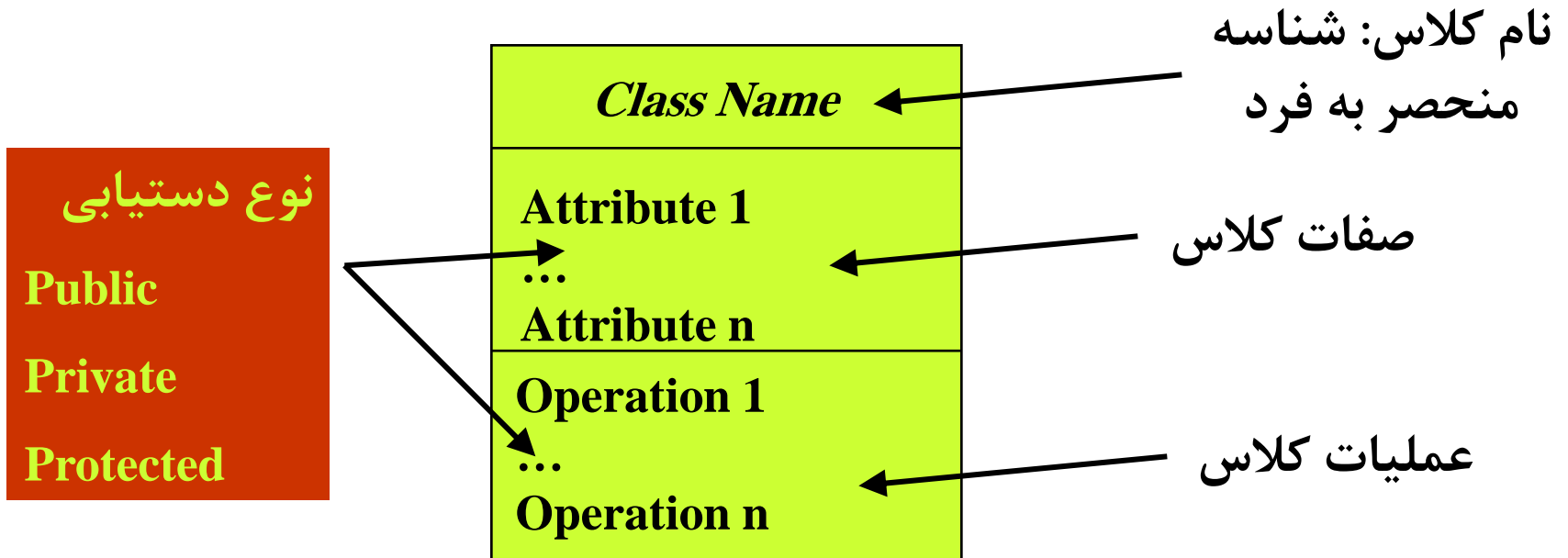


## ویژگیهای کلاس

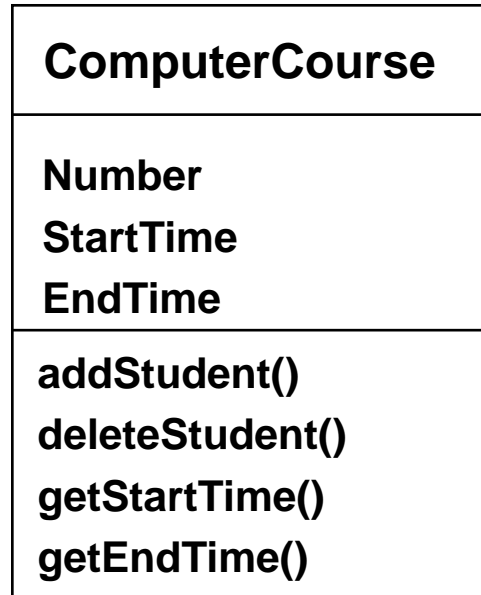
- کلاس به صورت **قالبی** که برحسب نیاز **نمونه‌های** (*Instances*) از آن برداشت می‌نمائیم، عمل می‌نماید
- گروه‌بندی اشیاء در کلاس
- مدیریت بهتر + قابلیت استفاده مجدد
- ابزار اصلی تجرید در شی‌گرا

# کلاس (ادامه)

## نمایش کلاس در UML

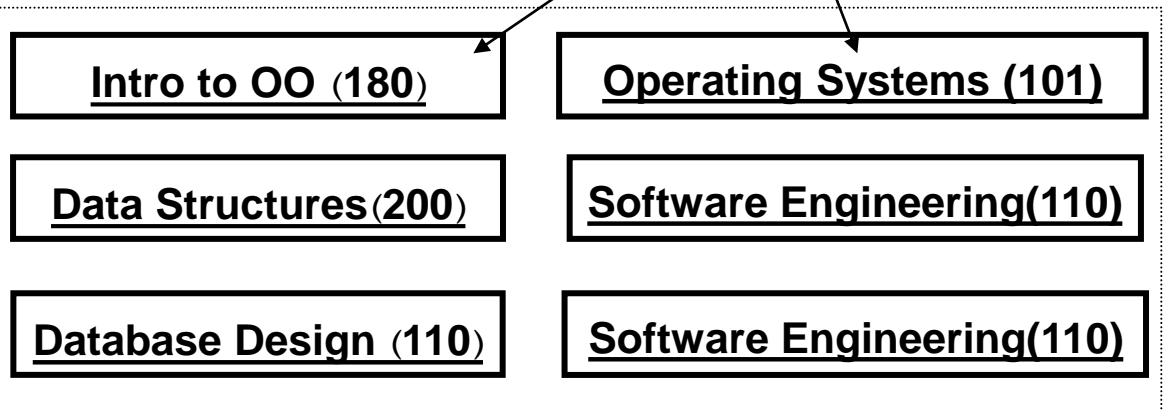


مثال ۱

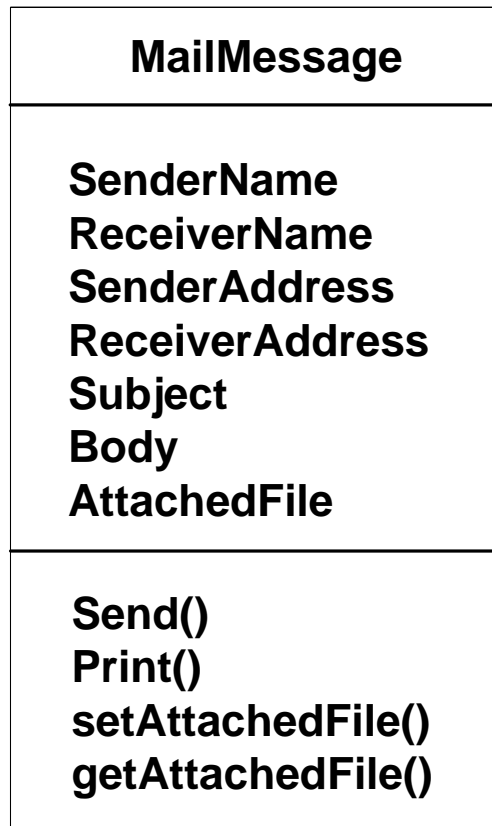


نمونه‌ها (اشياء)

Computer Course

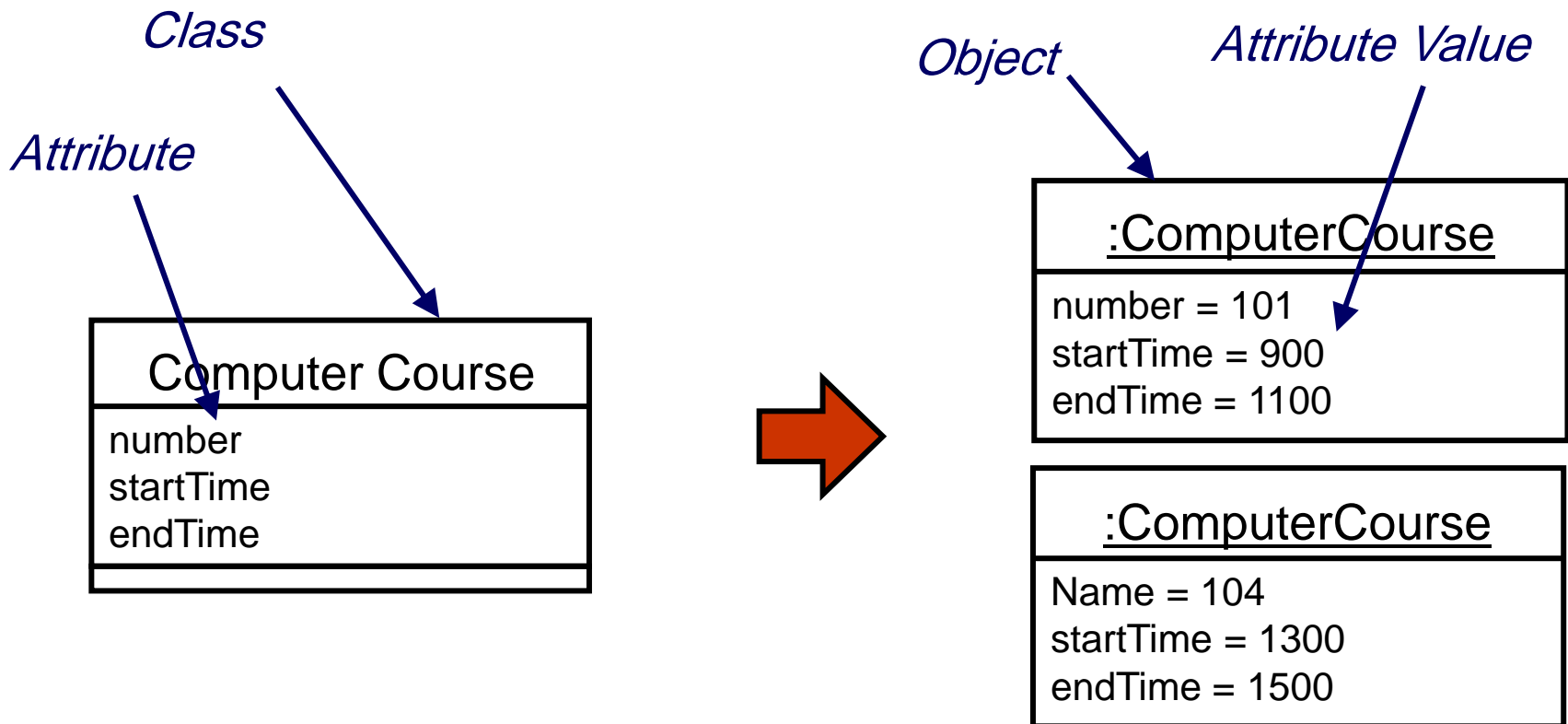


## مثال ۲: کلاس پیام الکترونیکی (MailMessage) در UML

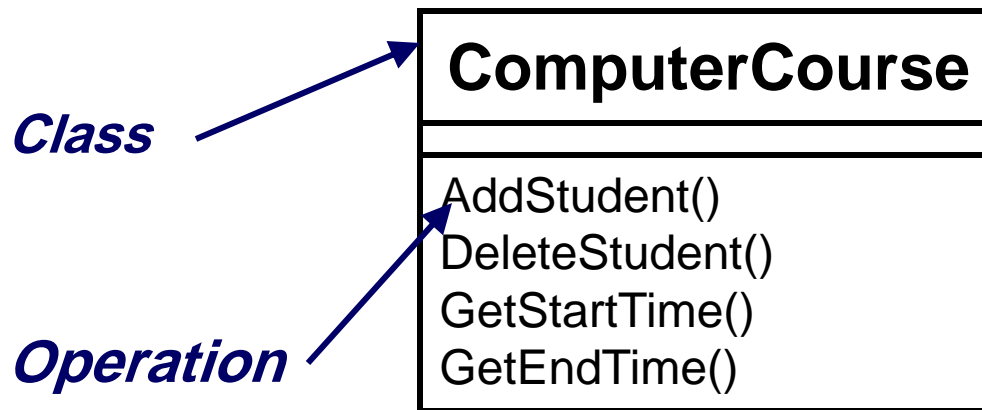


# کلاس (ادامه)

صفات (*Attributes*): ویژگی‌هایی از کلاس که نگهداری آن برای مهم است



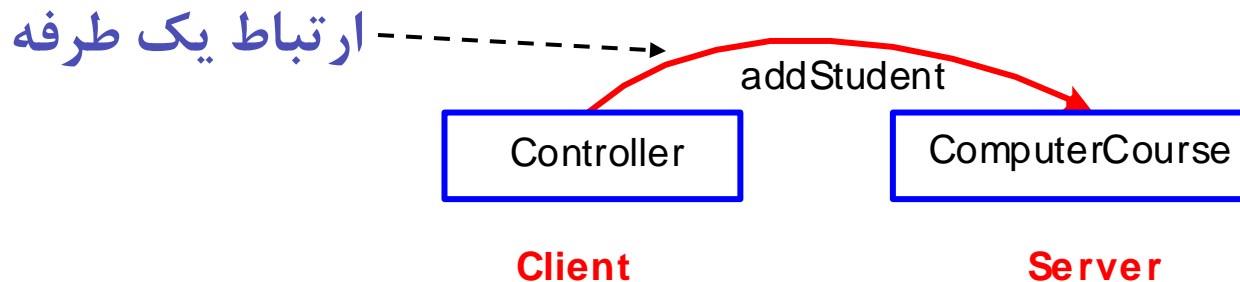
عملیات (*Operations*): رفتار کلاس در مقابل کلاس‌های دیگر





# مکانیزم ارتباط بین اشیاء

- ارتباط بین اشیاء و بهرمندی از سرویس‌های (عملیات) آنها از طریق مکانیزم تبادل پیام (*Message Passing*) صورت می‌گیرد



مقدار برگشتی (*Return value*)

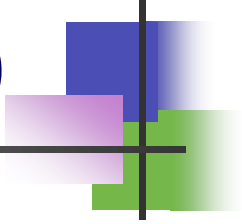
تغییر پارامترهای دریافتی

نحوه اعلام پاسخ به *Client*

# مکانیزم ارتباط بین اشیاء (ادامه)

- هر پیام عبارت از یک درخواست از طرف سرویس گیرنده به سرویس دهنده برای اجرای یک عمل (*Method Call*) است.

# رابطه بين كلاسها



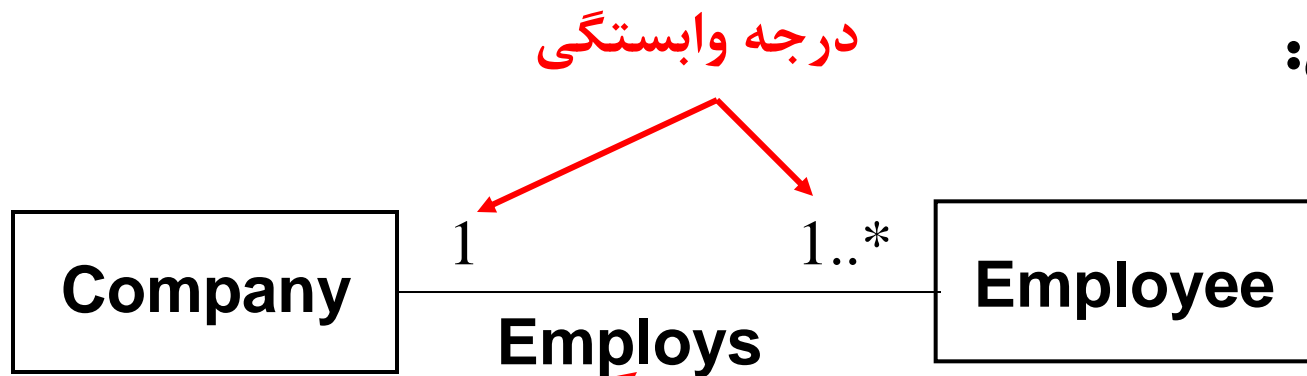
## روابط اساسی ■

- رابطه انجمنی (*Association*)
- رابطه تجمعی (*Aggregation*)
- رابطه عام/خاص (*Generalization/Specialization*)

# رابطه انجمنی

نوعی وابستگی معنایی (*Semantic Relationship*) بین نمونه‌هایی از کلاس‌های متفاوت است که با حذف این وابستگی عملاً هیچ ارتباط دیگری بین دو کلاس وجود نخواهد داشت

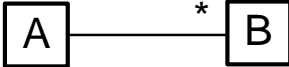
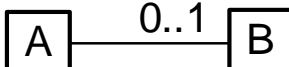
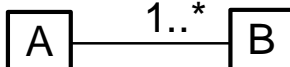
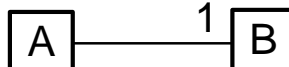
مثال:



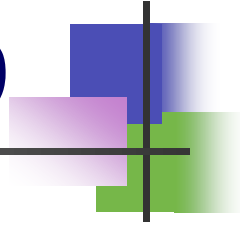
# رابطه انجمنی (ادامه)



## درجه وابستگی

یک نمونه از A همیشه با هیچ، یک نمونه، یا بیشتر از B ارتباط دارد.	یک نمونه از A همیشه با هیچ یا فقط یک نمونه از B ارتباط دارد.	یک نمونه از A همیشه با یک نمونه یا بیشتر از B ارتباط دارد.	یک نمونه از A همیشه با یک نمونه از B ارتباط دارد.
			

# رابطه تجمعی



” زمانی که یک کلاس از تلفیق تعدادی کلاس دیگر تشکیل گردد،  
این رابطه را تجمعی گویند “

**PART-OF** (پایین به بالا)

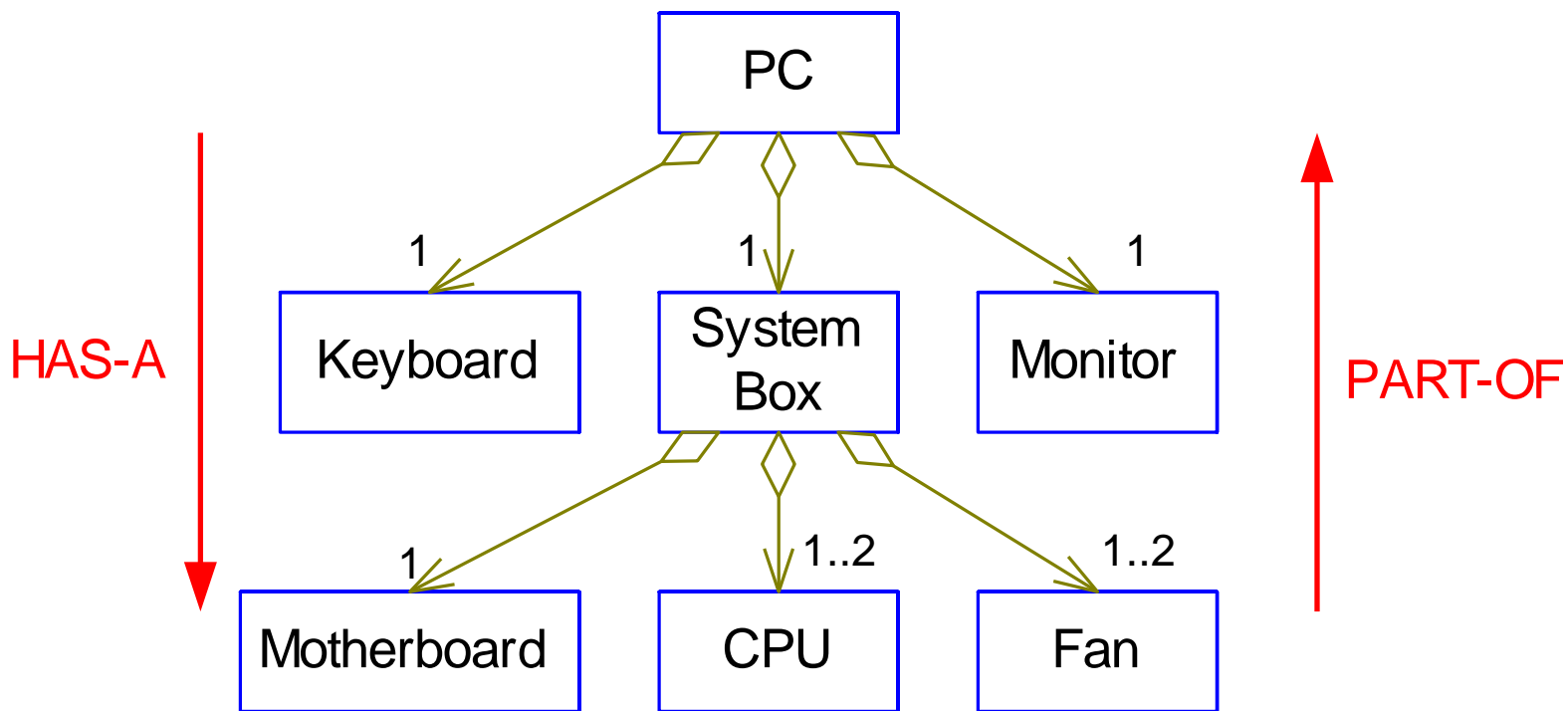
**HAS-A** (بالا به پایین)

برای مثال:

A PC **HAS-A** Keyboard

A Monitor is **PART-OF** a PC

# رابطه تجمعی (ادامه)



# رابطه عام / خاص

یک رابطه طبقه‌بندی (*Taxonomic Relationship*) بین یک عنصر عمومی‌تر و یک عنصر خاص‌تر به طوریکه عنصر خاص‌تر ساختار و/یا رفتار عنصر عمومی‌تر را به ارث برده و علاوه بر آن از ویژگیها و/یا رفتارهای اضافی برخوردار است

↑ (پایین به بالا) **IS-A / Generalized-to**

↓ (بالا به پایین) **Specialized-to**

برای مثال

The Hourly Employee **IS-A** Employee

A Full-Time Employee is **Specialized-to** Manager



# رابطه عام / خاص (ادامه)

## در وراثت

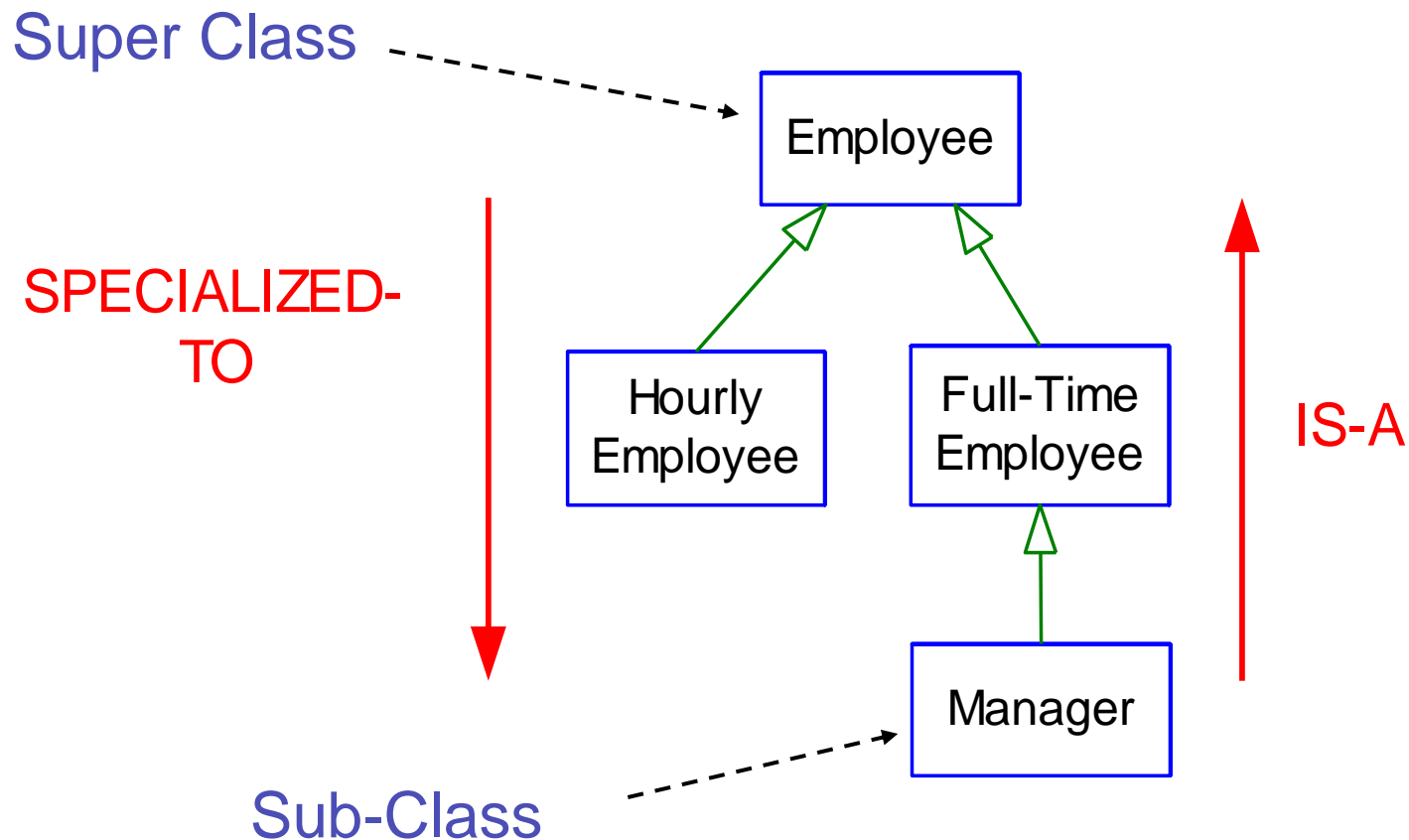
- کلاس فرزند یک تخصیص (*Specialization*) از کلاس پدر را نمایش داده، و
- همزمان کلاس پدر یک تعمیم (*Generalization*) از کلاس فرزند به حساب می آید

## انواع وراثت

- وراثت یگانه (*Single Inheritance*): برای هر زیر کلاس فقط یک کلاس پدر وجود دارد
- وراثت چندگانه (*Multiple Inheritance*): یک زیر کلاس از دو کلاس یا بیشتر ارث می برد

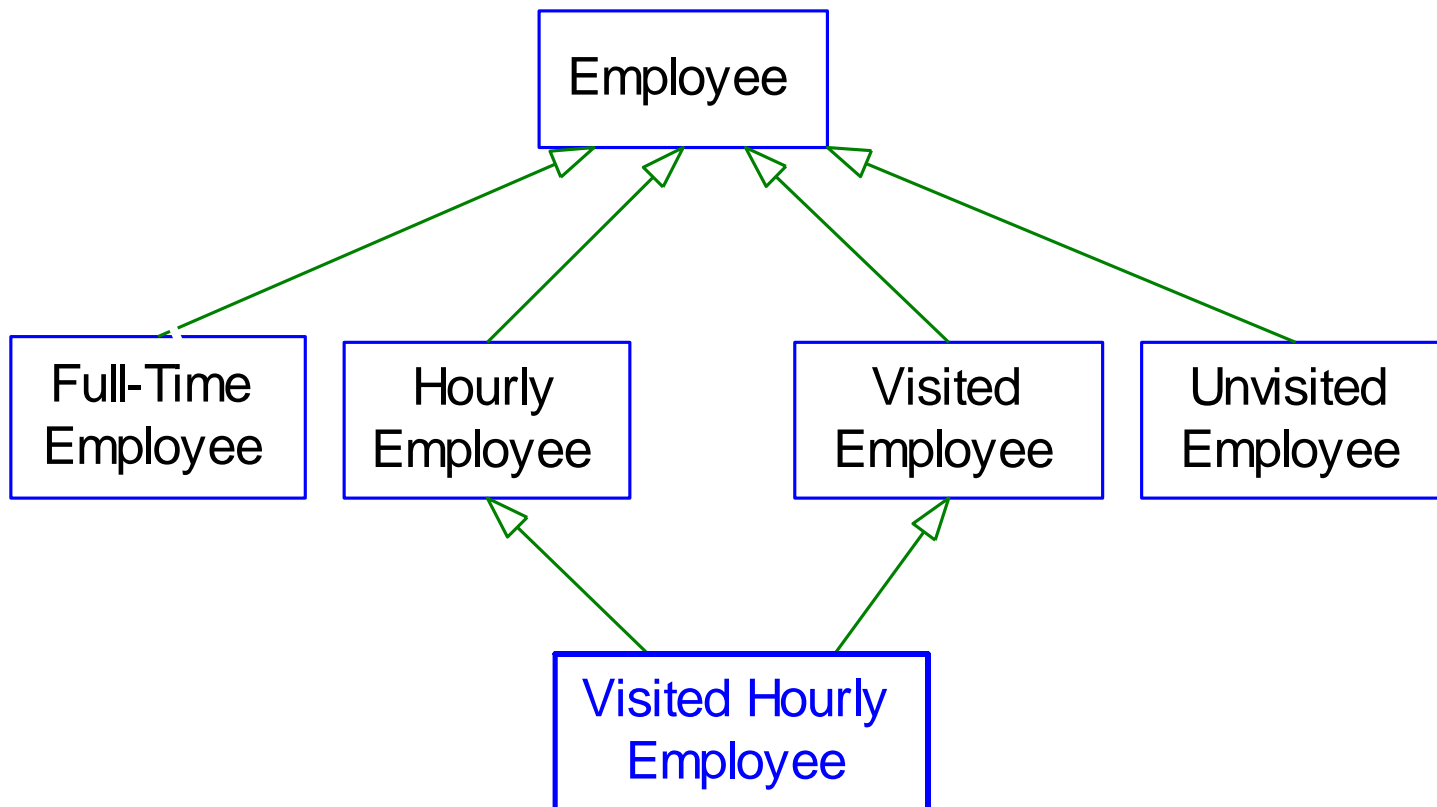
# رابطه عام / خاص (ادامه)

مثال ۱: انواع کارمند (وراثت یگانه)



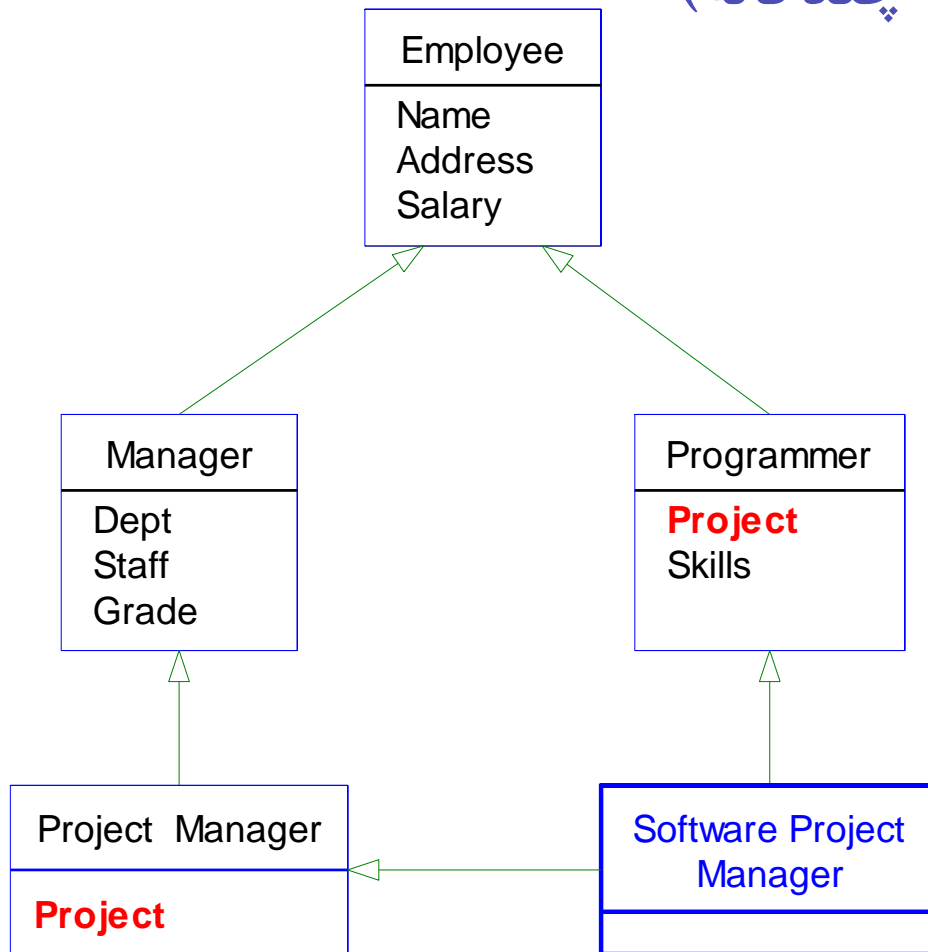
# رابطه عام / خاص (ادامه)

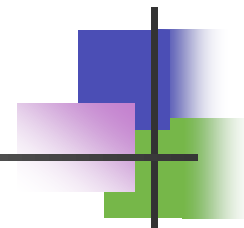
مثال ۲: انواع کارمند (وراثت چندگانه)



# رابطه عام / خاص (ادامه)

مثال ۳: انواع کارمند (وراثت چندگانه)





# پرسش و پاسخ