

بسمه تعالی

فصل سوم

آشنائی با مفاهیم اولیه شئی گرائی

مدرس: فریدون شمس

اهداف جلسه



- آشنائی با مفاهیم اساسی شیء گرائی
- آشنائی با انواع روابط بین کلاسها
- آشنائی با نماد گذاری مفاهیم اساسی در

UML

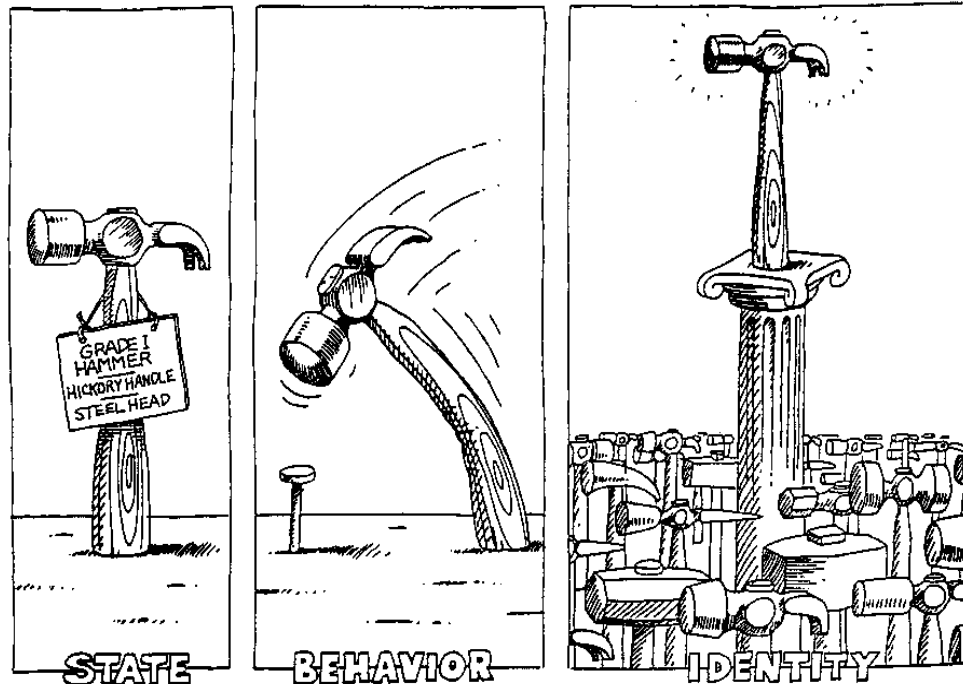


- شیء
- تعریف کلاس
- مفهوم داده مجرد
- مکانیزم ارتباط بین اشیاء
- رابطه بین کلاسها

Abstract Data Type

شیء (Object)

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



حالت

رفتار

هویت

شیء (ادامه)

- هویت (Identity): آن ویژگی از یک شیء است که آنرا از بقیه اشیاء متمایز می سازد.
- حالت (State): حالت یک شیء دربردارنده تمام خواص آن شیء و مقادیر جاری آنها.
- رفتار (Behavior): چگونگی عمل و عکس العمل یک شیء در مقابل دریافت و یا ارسال پیام.

شئی (ادامه

(

مثالهایی از اشیاء عبارتند از:

- موجودیتهای خارجی: سیستمهای برونی، دستگاه ها و مردمی که تولیدکننده یا مصرف کننده اطلاعات، ...
- موجودیتهای اطلاعاتی: انواع فایلها، مستندات، ...
- مفاهیم منطقی: دانشگاه، کلاس درس، دوره آموزشی، ...
- نقشهای انسانی : مشتری، تحلیلگر، معمار نرم افزار، ...

شئی (ادامه)



مثالهایی از اشیاء: (ادامه)

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

بر اساس تعریف قبل مفاهیمی مانند **عمل**، **صفت**، و **زمان** نمونه هایی از موجودیتهایی هستند که **یک شئی محسوب نمی شوند**.

شئ (ادامه)

(

مثال ۱: کتاب الف

الگوریتم ها + ساختمان داده ها = شئ

فیز

برای خرید...

شئ (ادامه

(



مثال ۲: درسهای رشته کامپیوتر

Intro to OO (180)

Operating Systems (101)

Data Structures(200)

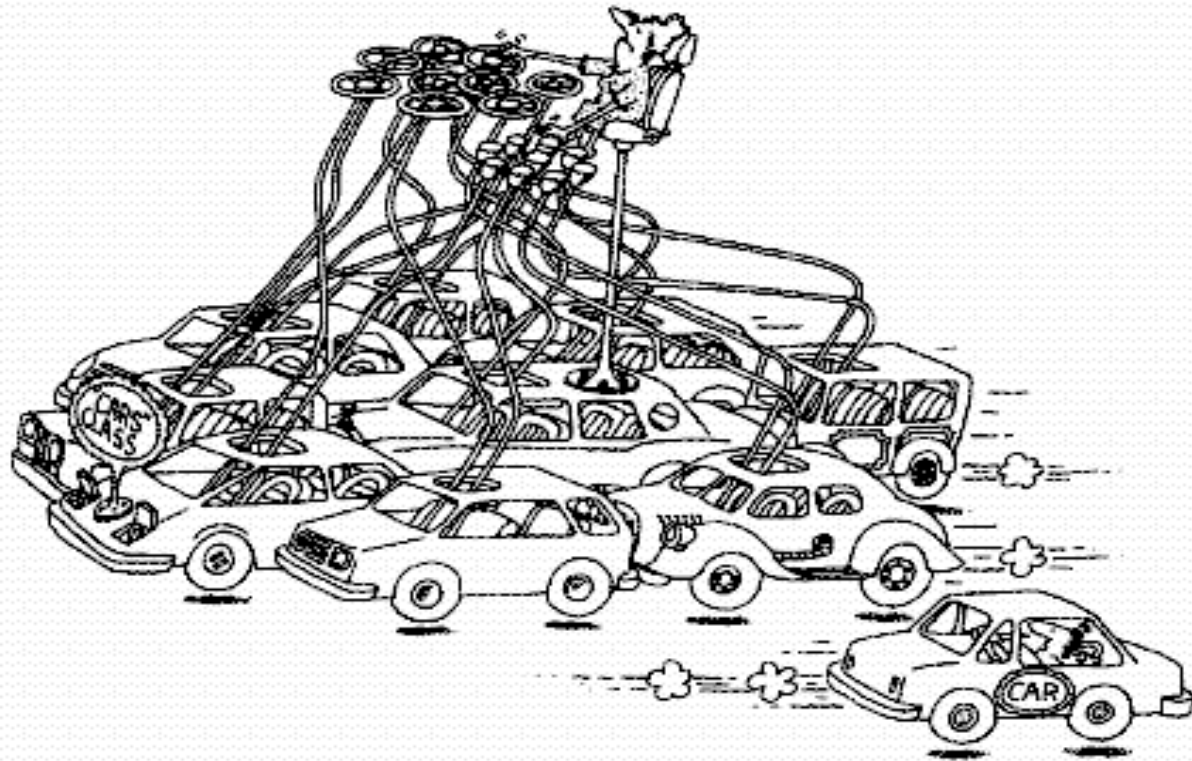
Software Engineering(110)

Database Design (110)

Software Engineering(110)

کلاس (Class)

”مجموعه ای از اشیاء که دارای ساختار و رفتار مشترک“



کلاس (ادامه)



ویژگیهای کلاس:

– کلاس به صورت **قالبی** که بر حسب نیاز نمونه هائی (Instances) از آن برداشت می نماییم، عمل می کند.



مدیریت بهتر+

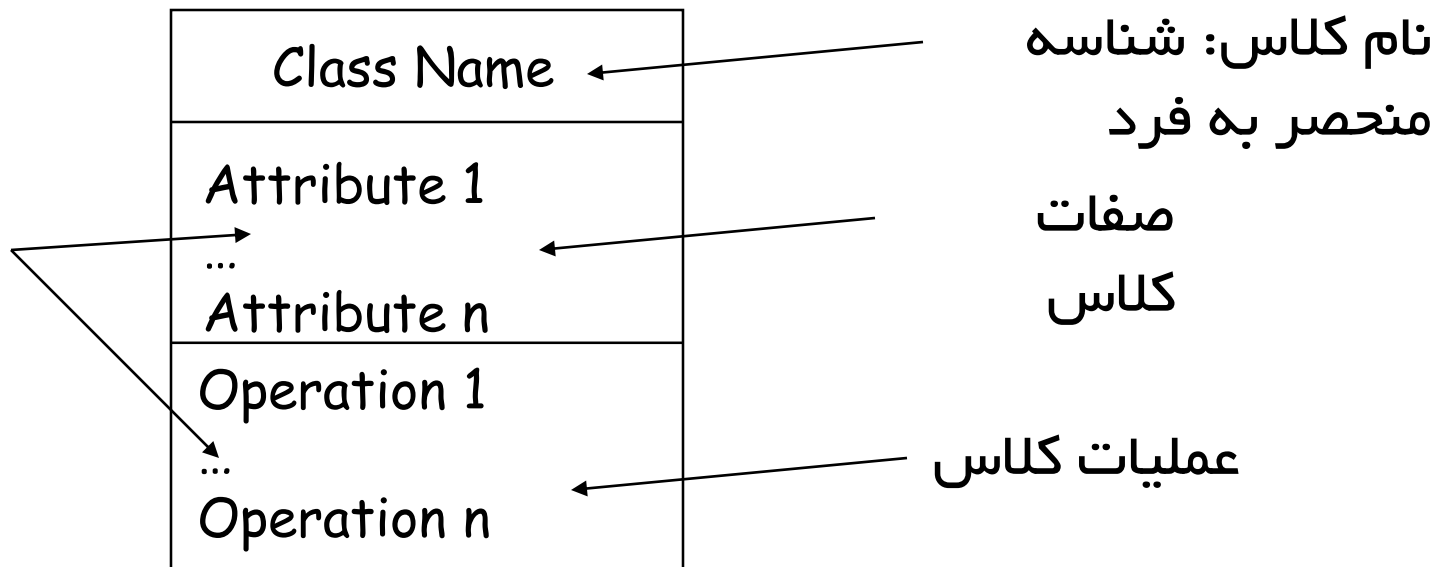
– گروه بندی اشیاء در کلاس
قابلیت استفاده مجدد

– ابزار اصلی تجرید در OO

کلاس (ادامه)



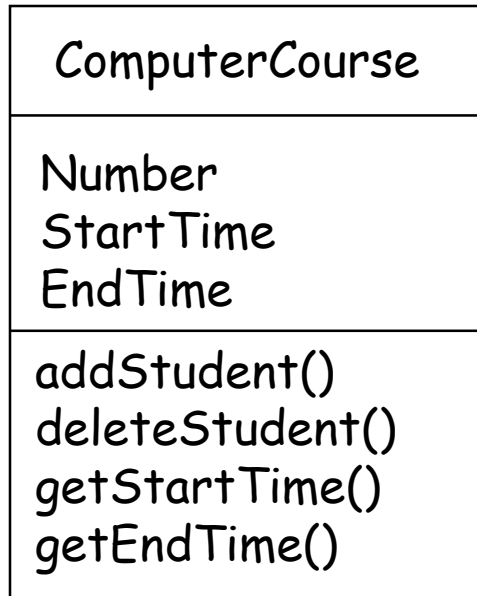
نمایش کلاس در UML:



كلاس (ادامه)



مثال ۱ :



نمونه ها (اشياء)

Intro to OO (180)

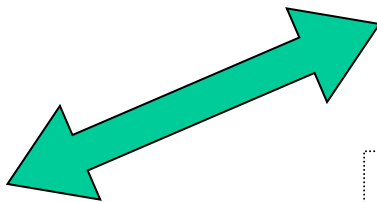
Operating Systems (101)

Data Structures(200)

Software Engineering(110)

Database Design (110)

Software Engineering(110)



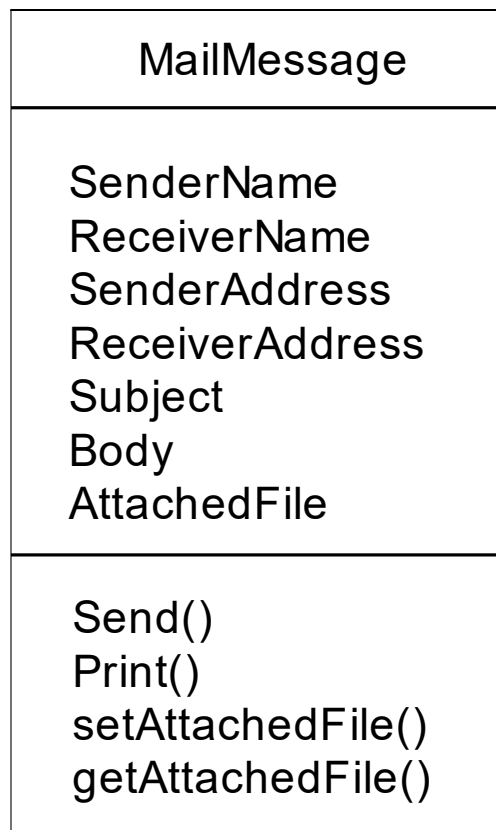
Computer Course



کلاس (ادامه)



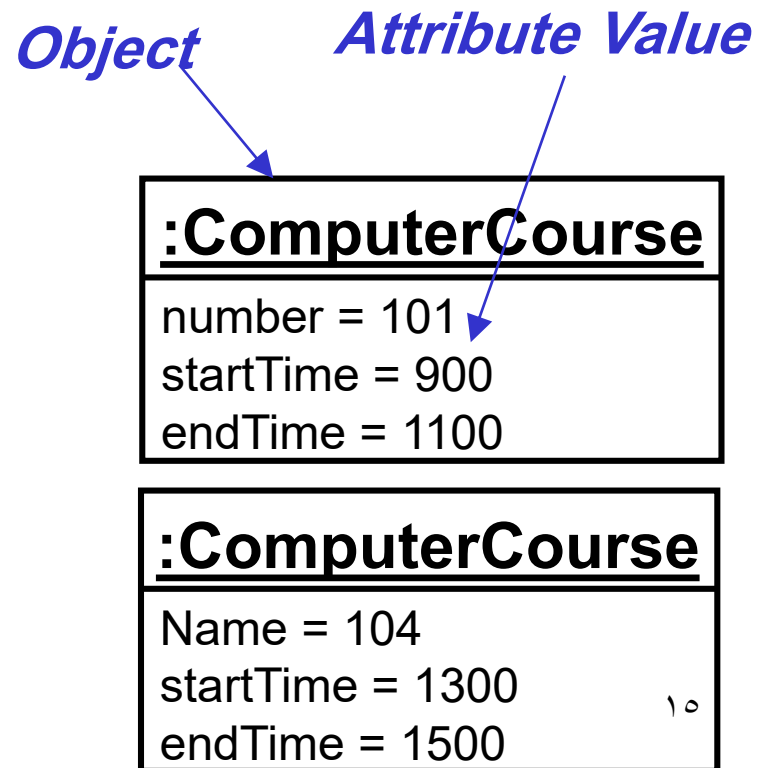
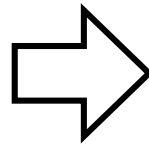
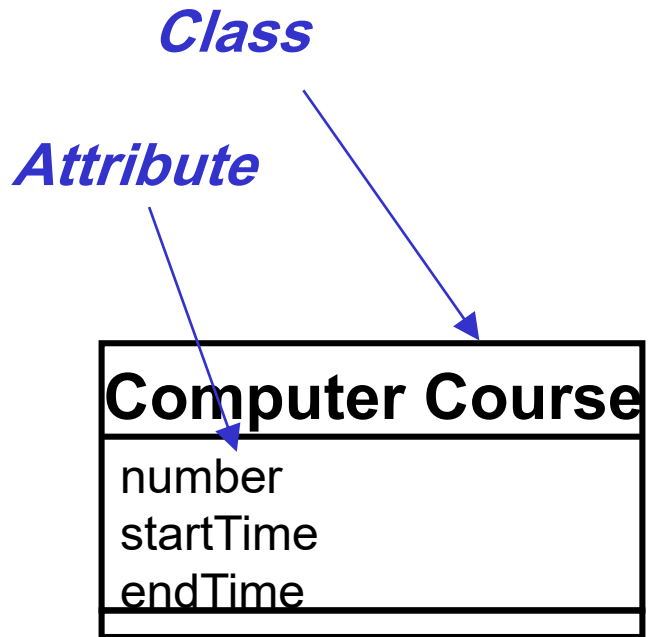
مثال ۲: کلاس پیام الکترونیکی (MailMessage) در UML



کلاس (ادامه)

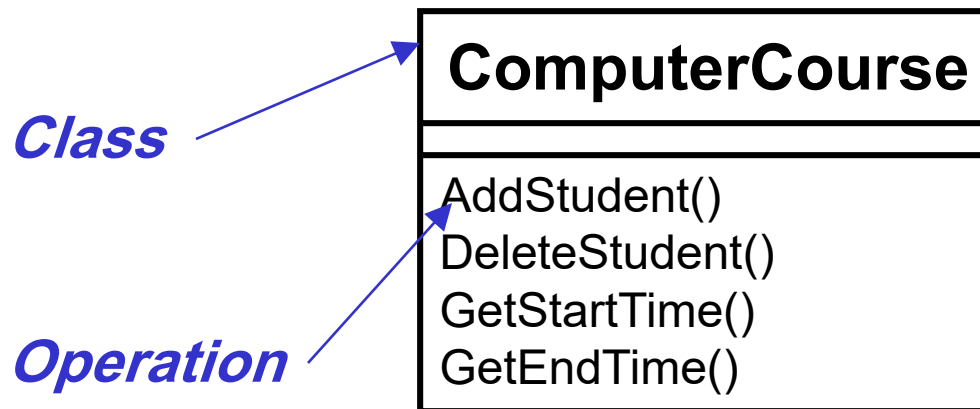


صفات (Attributes): ویژگیهایی از کلاس که نگهداری آن برای ما مهم است





عملیات (Operations): رفتار کلاس در مقابل کلاسهای دیگر



Abstract Data Type(ADT)



”ADT امکانی را فراهم می آورد که باعث شناسائی (معرفی) اشیاء از طریق بیان ساختار و رفتار آنها بدون نیاز به اطلاع از نحوه پیاده سازی آن ساختار یا رفتار باشد.“

با توجه به مفهوم ADT می توان هر کلاس را به دو قسمت زیر تقسیم کرد:

– واسط: بیان کننده آنچه کلاس را انجام می دهد ← مفهوم
ADT

– پیاده سازی

ADT (ادامه)

مزایا

← اعمال اصل Encapsulation
از نحوه استفاده از کلاس

← استقلال پیاده سازی

← پدیده انتشار تغییرات نخواهیم

← نگهداری

داشت

آسانتر



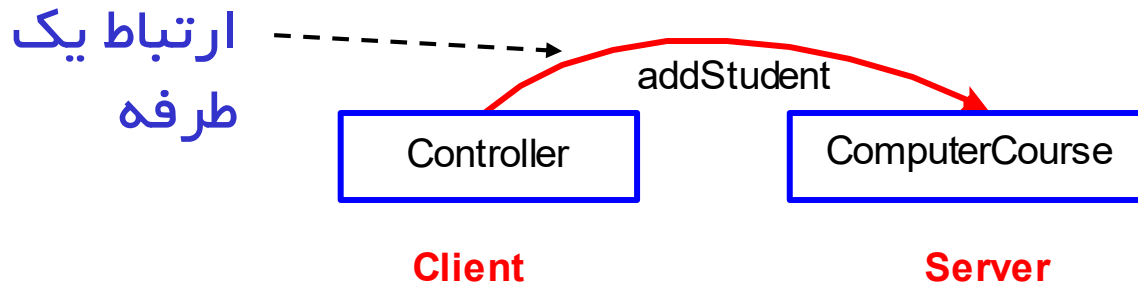
مثال: واسطی برای کلاس پیاده سازی کننده Stack (IStack) در C++

```
// Interface to Stack class
class IStack
{
// Operations (Services)
public:
// puts an integer on top of stack
virtual void push( int ) = 0;
// gets an integer from top of stack then removes it
virtual int pop( void) = 0;
// Is stack is empty?
virtual bool isEmpty() = 0;
// Is stack is full?
virtual bool isFull() = 0;
};
```

مکانیزم ارتباط بین اشیاء



”ارتباط بین اشیاء و بهر مندی از سرویس های (عملیات) آنها از طریق مکانیزم تبادل پیام (*Message Passing*) صورت می گیرد“



مقدار برگشتی (Return value)
تغیر 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. این درخواست باید مطابق با امضای عمل
(*Method Signature*) باشد“

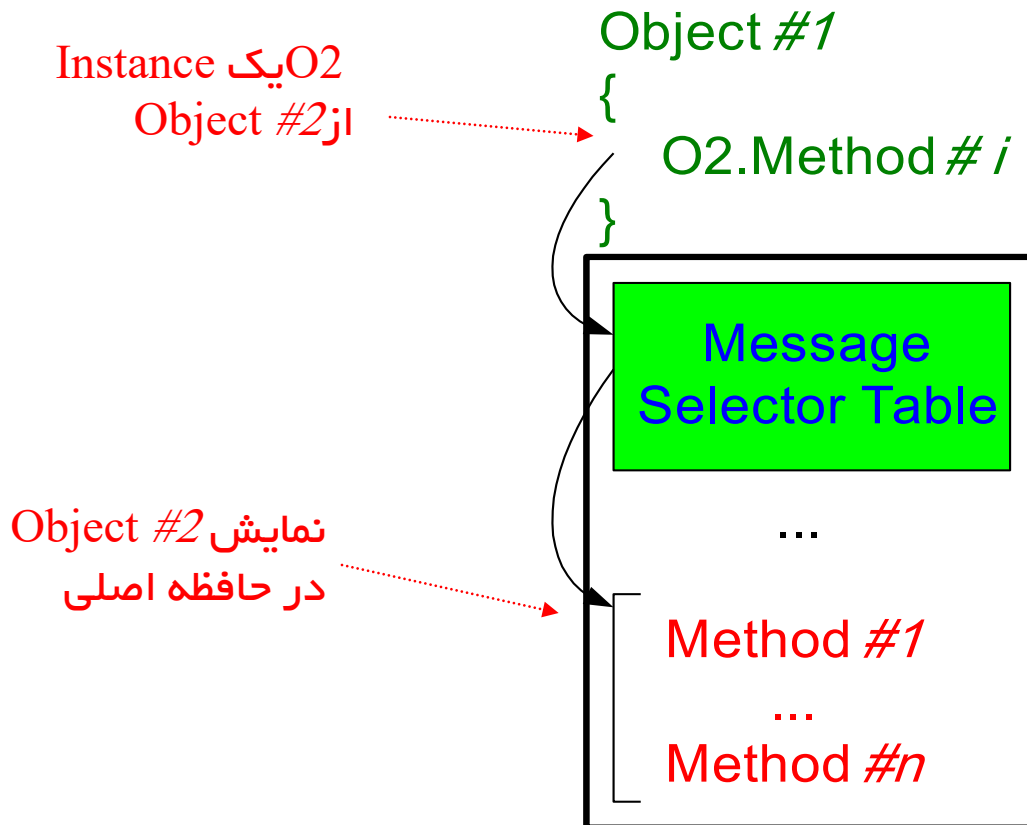
امضای عمل در بردارنده نوع، تعداد، و ترتیب پارامترها
و نوع مقدار برگشتی عمل است

مکانیزم ارتباط بین اشیاء

(ادامه)



نحوه اجرای پیام: با استفاده از انتخاب دهنده پیام (**Message** Selector)



رابطه بین

کلاسها

روابط اساسی:



Association Relationship رابطه انجمنی

Aggregation Relationship رابطه تجمعی

Generalization/Specialization Relationship رابطه عام/خاص

روابط دیگر

Composition Relationship رابطه ترکیب

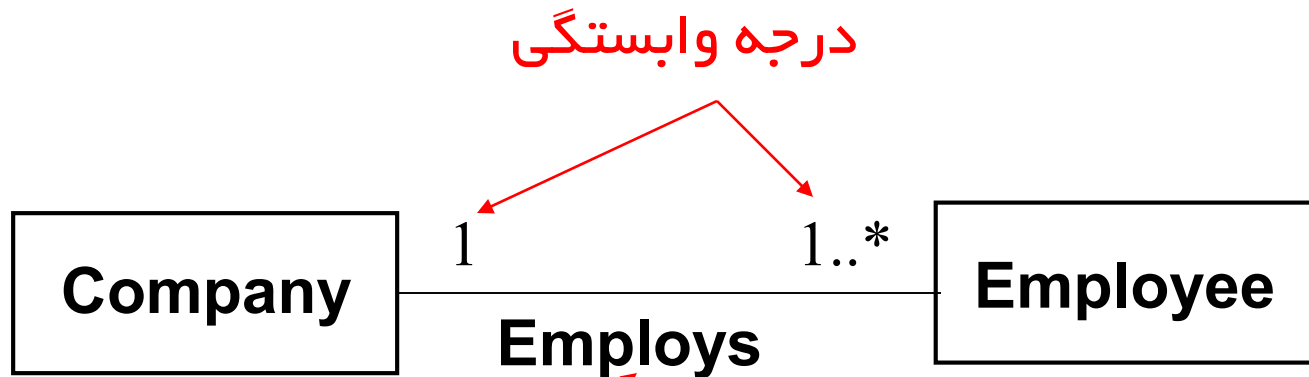
Using Relationship رابطه استفاده از

رابطه انجمنی



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

مثال:

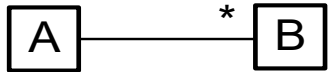
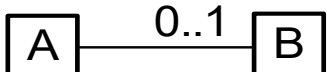
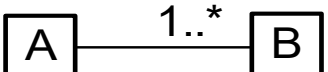
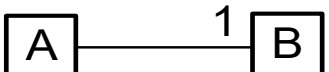


مفهوم رابطه

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“

↑ (پایین به بالا) **PART-OF**

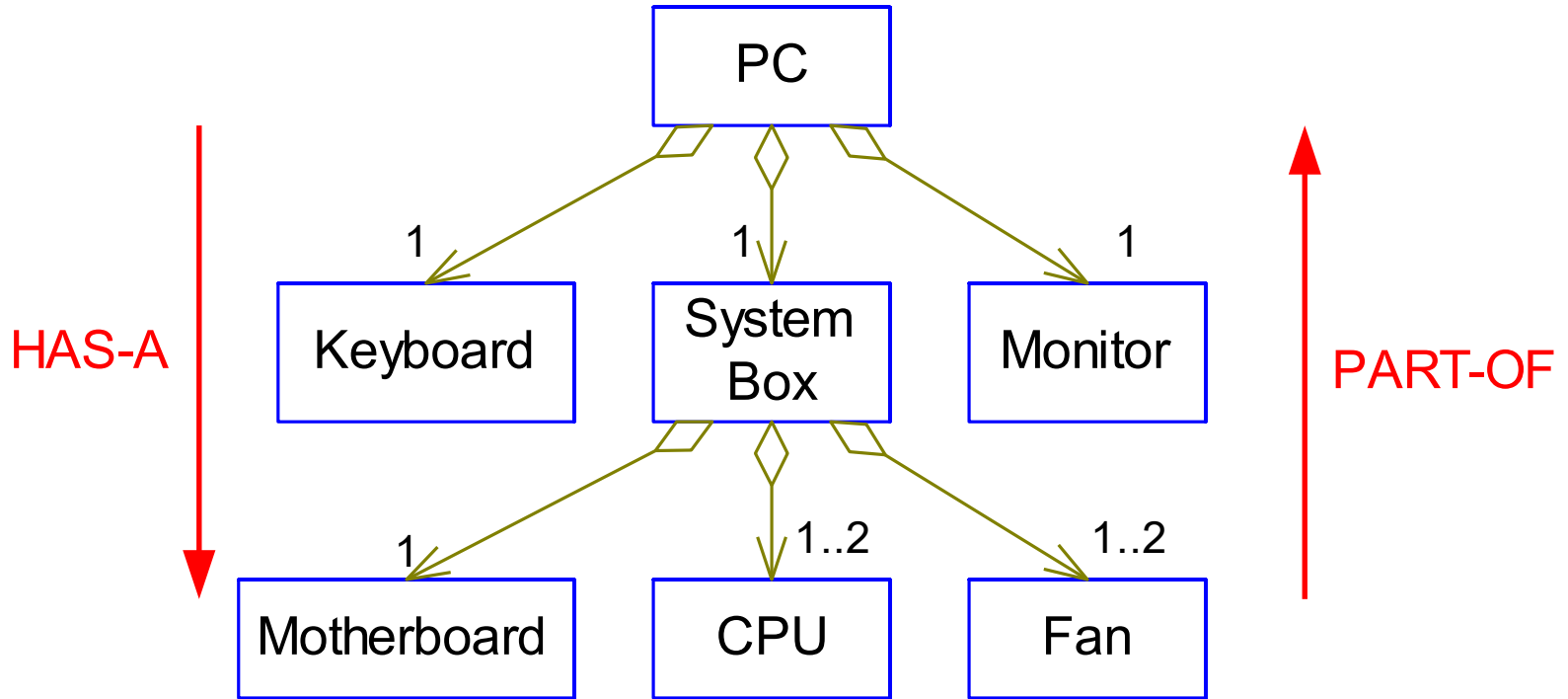
(بالا به پایین) **HAS-A**

برای مثال:

A PC **HAS-A** Keyboard

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

رابطه تجمعی (ادامه)



رابطه عام/خاص



”یک رابطه طبقه بندی (*Taxonomic Relationship*) بین یک عنصر عمومی تر و یک عنصر خاص تر به طوریکه عنصر خاص تر ساختار و/یا رفتار عنصر عمومی تر را به ارث برده و علاوه بر آن از ویژگیها و/یا رفتارهای اضافی برخوردار است“

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

برای مثال :

The Hourly Employee *IS-A* N 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.
- 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)

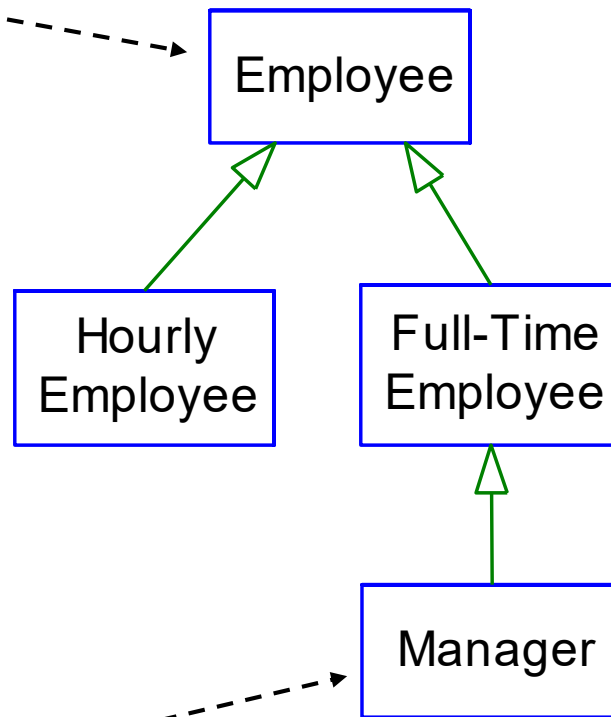


مثال ۱: انواع کارمند (وراثت یگانه)

Super Class

SPECIALIZED-
TO

Sub-Class

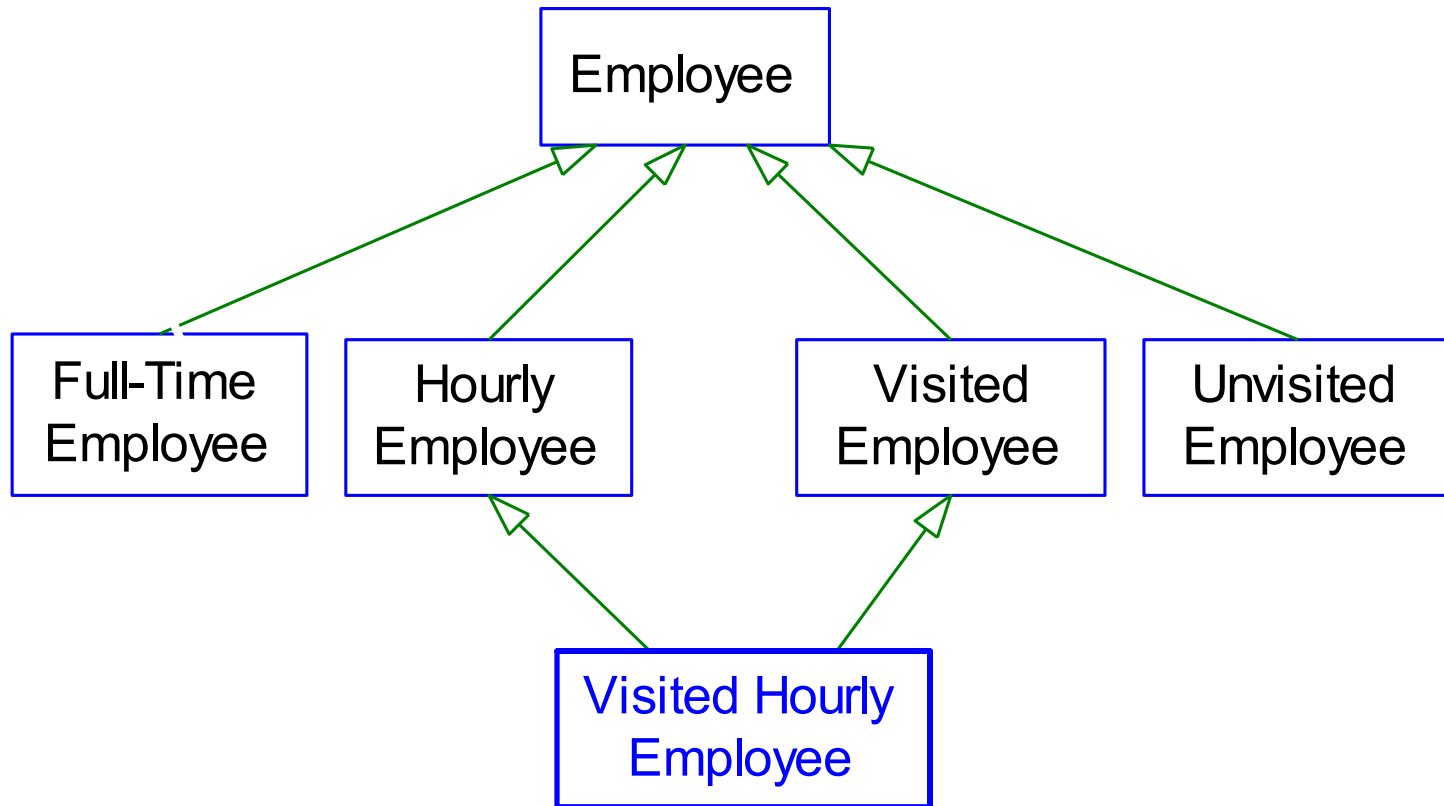


IS-A

رابطه عام/خاص (ادامه)



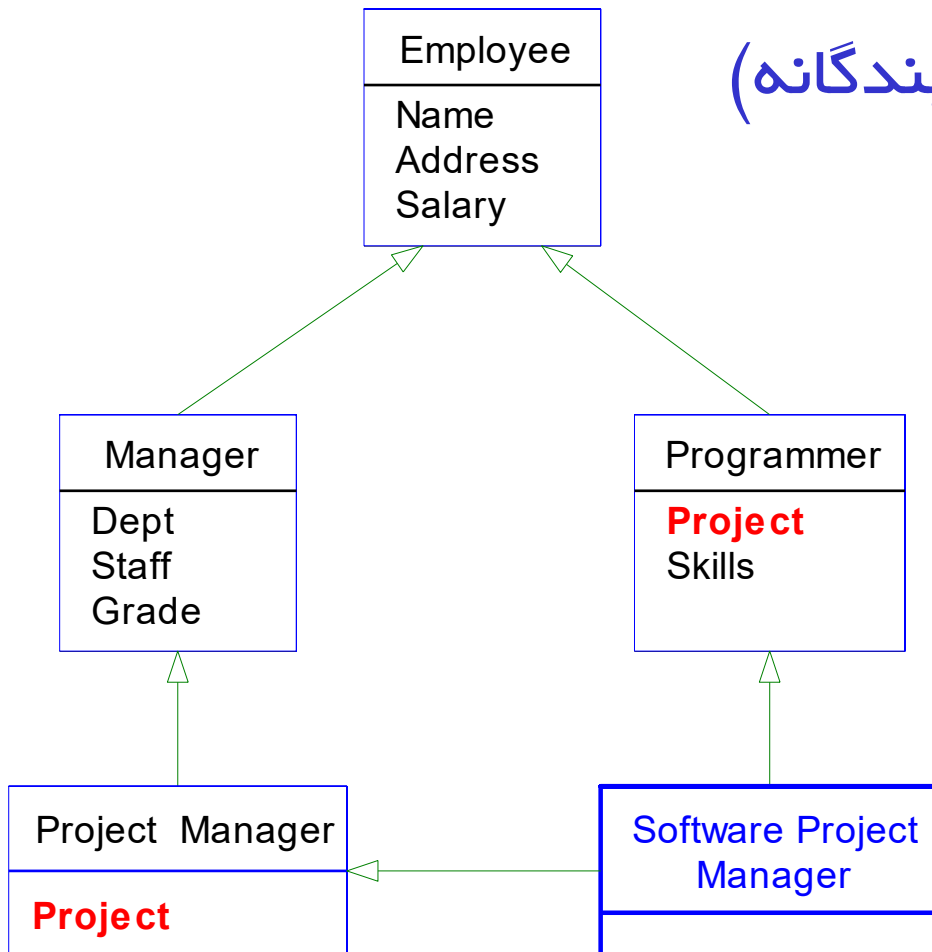
مثال ۲: انواع کارمند (وراثت چندگانه)



رابطه عام/خاص (ادامه)



مثال ۳: انواع کارمند (وراثت چندگانه)



رابطه عام/خاص (ادامه)



پیاده سازی بخشی از مثال ۳ در زبان C++

```
class Programmer
{
private:
    string Skills;
protected:
    string Project;
};
```

```
class ProjectManager
{
protected:
    string Project;
};
```

رابطه عام/خاص (ادامه)



```
class SoftwareProjectManager {  
public:  
    SoftwareProjectManager ()  
    {  
        Programmer::Project = "Project 1";  
        ProjectManager::Project = "Project 2";  
    }  
};
```