

بسمه تعالی

درس مهندسی نرم افزار پیشرفته

معرفی اصول شیء گرائی برای مقابله با پیچیدگی نرم افزار

نیمسال اول: ۸۷ - ۱۳۸۶

اهداف جلسه

- آشنائی با اصول شیء گرائی
- درک نقش اصول شیء گرائی در کنترل پیچیدگی سیستمهای نرم افزاری
- درک مزایای مدل شیء و کاربردهای آن

فهرست

- مقدمه
 - روشهای طراحی
 - تجرید
 - دربرگیری
 - واحدبندی
 - سلسله مراتب
 - مزایای مدل شیء و کاربردهای آن
- Abstraction
Encapsulation
Modularity
Hierarchy

مقدمه



• نقش نرم افزار در **روزهای اولیه** عصر کامپیوتر

▪ نقش ثانویه تلقی می شد (**Afterthought**)

▪ هزینه اساسی طراحی یک سیستم کامپیوتری از آن سخت افزار بود

▪ بیشتر نرم افزارها بوسیله یک نفر تولید و توسعه می شدند

مقدمه (ادامه)



- نقش نرگ افزار در **روزهای اولیه** عصر کامپیوتر (ادامه)
 - فرایند طراحی به صورت ضمنی در ذهن برنامه نویس انجام می شد
 - زبان رایج: زبان ماشین سپس اسمبلی ابداع شد
 - قابلیت سخت افزار بسیار محدود بود ← برنامه ها کوچک و ساده بودند
 - ظاهرا نیازی به مستند سازی نبود



• نقش نرم افزار در روزهای کنونی

- نقش بسیار اساسی

- هزینه اساسی طراحی یک سیستم کامپیوتری از آن نرم افزار است

- بیشتر نرم افزارها بوسیله تیمهای چند نفره تولید و توسعه می شوند

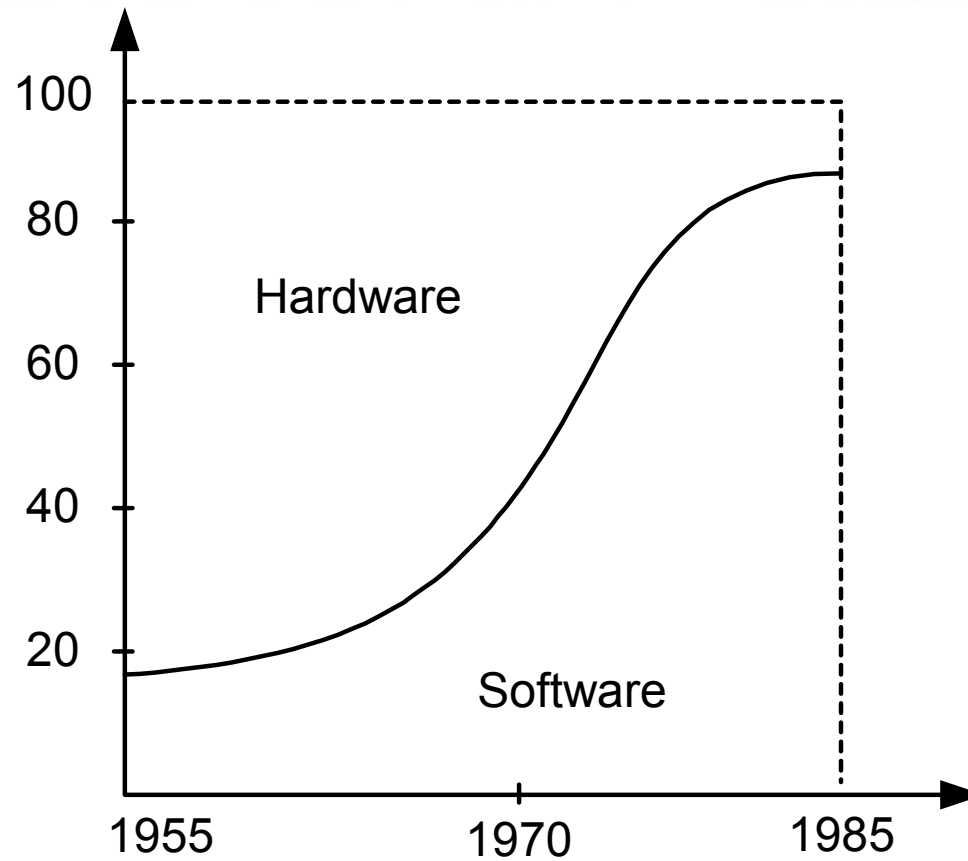
- فرایند طراحی به صورت صریح در خارج از ذهن برنامه نویس انجام می شود



• نقش نرم افزار در (روزهای کنونی) (ادامه)

- سخت افزار سریعتر، ارزانتر، و قابل اطمینان تر ←
- اقتصادی شدن فرایند خودکار سازی بسیاری از کاربردهای صنعتی و تجاری ←
- تقاضا بر نرم افزارهای پیچیده تر
- زبانهای رایج: زبانهای سطح بالا، ساخت یافته، و شیء گرا
- درک اهمیت مستندسازی سیستمها
- احساس نیاز به روشهای تحلیل و طراحی

مقدمه (ادامه)



درصد هزینه های نرم افزار در مقابل هزینه های سخت افزار از سالهای ۱۹۵۵ تا ۱۹۸۵

روشهای طراحی



- طراحی ساخت یافته **Structured Design**
- طراحی مبتنی بر داده ها **Data-Driven Design**
- طراحی شیء گرائی **Object-Oriented Design**

اصول شئ گرائی



شئ گرائی

تجربید

محصور سازی

واحد بندی

سلسله مراتب

تجرید (Abstraction)



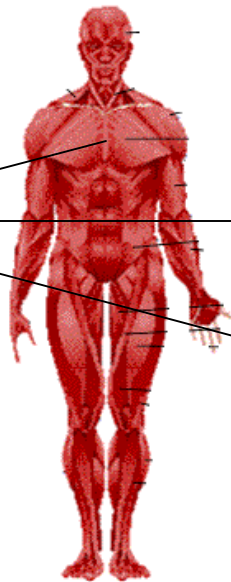
”تجرید عبارتست از فرایند متمرکز شدن روی ویژگیها و رفتارهای اصلی یک پدیده، و نادیده گرفتن ویژگیهای موقت و غیر مهم آن پدیده، از یک زاویه دید مشخص“

تجزید (ادامه)

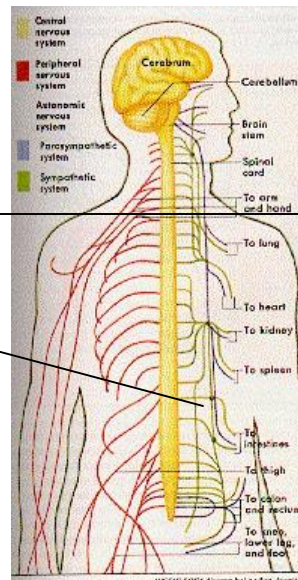
مثالهایی از تجزید



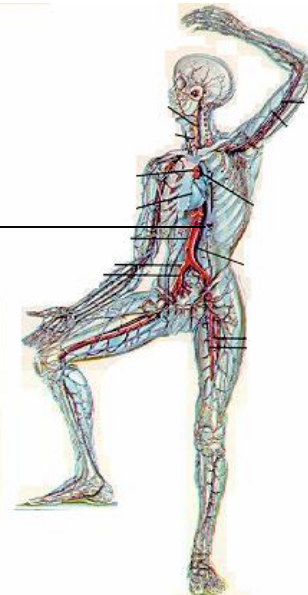
دیدهای
گوناگون از
بدن انسان



عضلات بدن




سیستم عصبی



سیستم گردش خون

تجربید (ادامه)



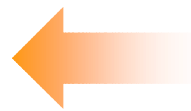
- ۲ بیان روابط میان اجزاء یک سیستم مکانیکی توسط یک معادله ریاضی
- ۳ استفاده از نماد  برای نمایش حضور موجودیت انسان در یک صحنه
- ۴ نمایش گرافیکی رفتار یکی سیستم

نقش تجربید در کنترل پیچیدگی:

■ یکی از ابزارهای اصلی کنترل و تسلط بر پیچیدگی

بوسیله تجربید

تنها ابعاد اساسی پدیده
مد نظر خواهد شد



جزئیات بی شماری که درباره یک پدیده مطرح است

تجريد (ادامه)

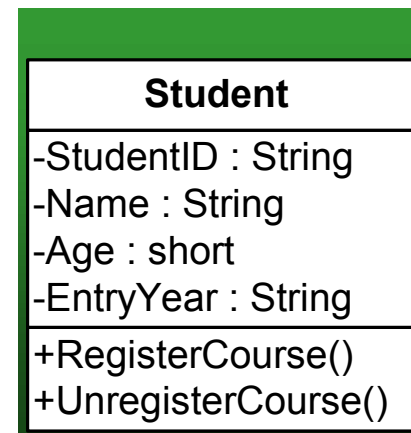
انواع تجريد:

■ تجريد موجوديت (Entity Abstraction)

Real Object: Student



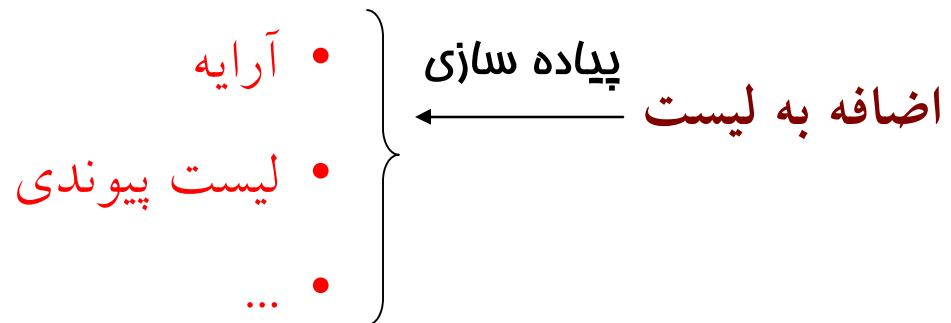
Abstraction: Student



تجريد (ادامه)

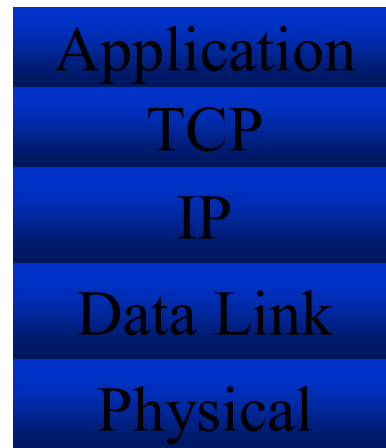


■ تجريد رفتار:



■ تجريد ماشين مجازي:

پروتکل TCP/IP



تجرید (ادامه)



ویژگیهای تجرید:

- برای یک شیء تجریدهای گوناگونی وجود دارد
- تجرید با نمود خارجی یک شیء سر و کار دارد
- تجرید سطوحی دارد (میزان پرداختن به جزئیات)
- همه تجریدها دارای ویژگیهای ساکن و پویا هستند
- در شیء گرائی مفهوم تجرید خود را در قالب نوع داده مجرد (Abstract Data Type) نشان می دهد

محصور سازی (Encapsulation)



”محصور سازی عبارت از عدم پذیرش تاثیرات ناخواسته
(*Side Effects*) و یا کنترل نشده و محدود کردن طرق
دسترسی به یا استفاده از یک شیء“

با توجه به این اصل هر شیء از دو مولفه زیر تشکیل می گردد:

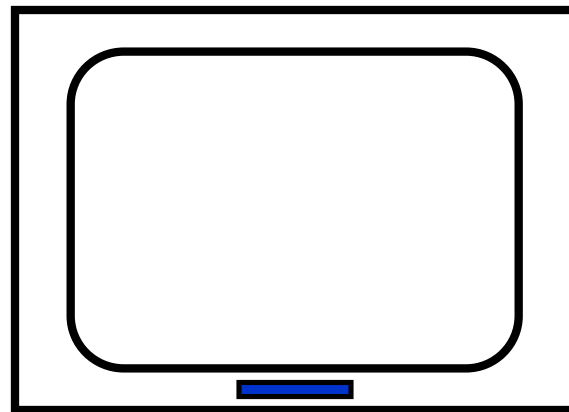
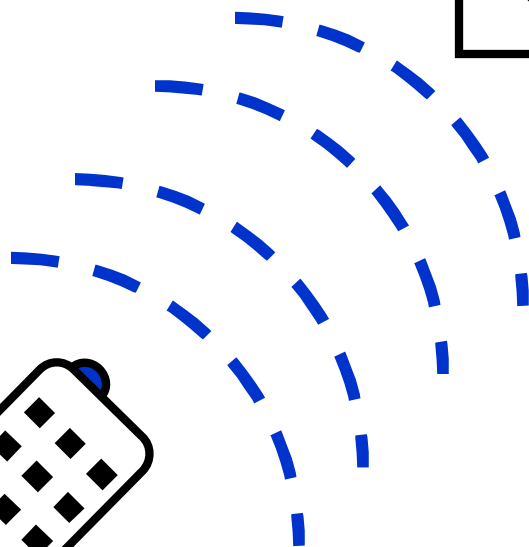
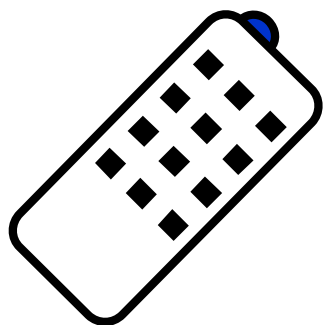
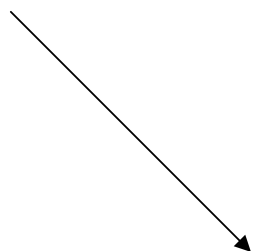
۱- واسط (Interface): توصیفی از سرویسهایی که این شیء در اختیار Clientها قرار می دهد.

۲- ساختار داخلی: داده ها + اعمال روی داده ها

محصول سازی (ادامه)



واسطه



شیء



محصول سازی (ادامه)

نقش محصول سازی در کنترل پیچیدگی:

- کنترل و تسلط بر پیچیدگی بوسیله کنترل راه های دسترسی به یک شیء که باعث جلوگیری از خرابکاریهای احتمالی و محلی کردن گستره خطاها در خود شیء
- با ثبات واسط یک شیء، می توان هر تغییری در پیاده سازی آن شیء انجام گیرد
- مفهوم یک واسط، چند پیاده سازی امکان استفاده مجدد را بالامی برد



محصول سازی (ادامه)

قاعده باز و بسته:

”هر ماژولی برای تعریف کننده آن باز و برای استفاده کننده بسته است“

ماژول باز: ماژولی که برای اعمال تغییرات آماده باشد

ماژول بسته: ماژولی که امکان تغییرات در آن وجود ندارد

”ما به ماژولهایی نیاز داریم که همزمان باز و بسته باشند“

محصور سازی (ادامه)

ویژگیهای دربرگیری:

- ارتباط بین اشیاء تنها از راه واسطها باشد.
- تجرید مکانیزم تعیین جزئیاتی است که باید پنهان شود، است. اما محصورسازی، فرایند پنهان سازی جزئیات و کنترل دسترسی به آن خواهد بود.
- محصور سازی یک مفهوم نسبی.

واحد بندی (Modularity)



”سیستمی را واحد بندی شده می گویند که به مجموعه ای از ماژولهای (واحدها) منسجم و معنی دار که وابستگی بین آنها حد اقل است تجزیه شده باشد“

• **ماژولها:** واحد تشکیل دهنده ساختار فیزیکی سیستم نرم افزاری (شبیه مدارات مجتمع در سخت افزار)

مثالی از واحدها:

- فایلها در C و C++
- واحدها (Units) در Object Pascal
- مولفه ها (Components) در استانداردهای COM ، Java Beans و .NET.

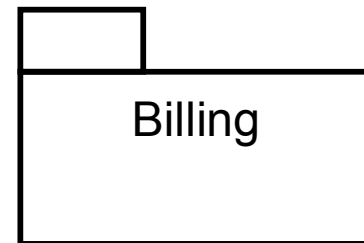
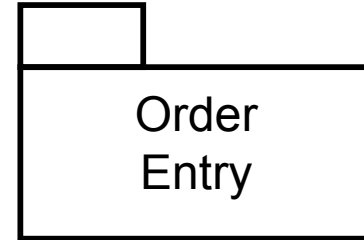
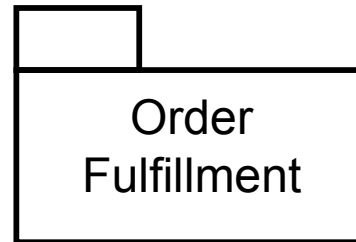
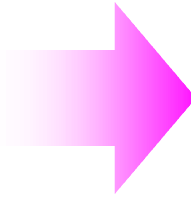
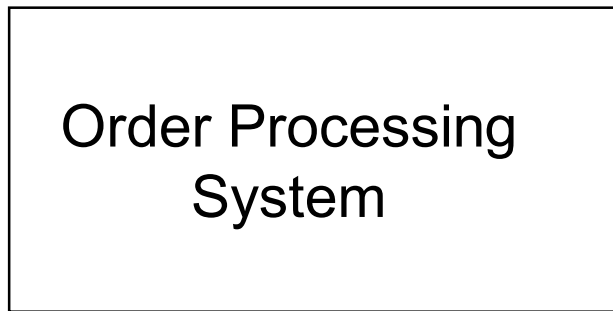
واحد بندی (ادامه)

- انسجام (Cohesion): انسجام عبارتست از درجه ارتباط عملکردهای عناصر داخلی یک ماژول

- وابستگی (Coupling): وابستگی عبارتست از درجه ارتباط ماژولهای گوناگون با یکدیگر

واحد بندی (ادامه)

مثال:



واحد بندی (ادامه)

نقش واحد بندی در کنترل پیچیدگی:

■ شکستن مساله به اجزائی کوچکتر یکی از راههای کارا برای مقابله با پیچیدگی

مثال: اگر مسئله P را به زیر مسئله های $P1$ ، $P2$ و $P3$ تقسیم نماییم آنگاه

$$C(P) > C(P1) + C(P2) + C(P3)$$

$$E(P) > E(P1) + E(P2) + E(P3)$$

C: Complexity

E: Solving Energy

توجه معادلات فوق: هنگام شکستن P وابستگی بین $P1$ ، $P2$ ، و $P3$ نادیده گرفته می شود

واحد بندی (ادامه)

بنابر استدلال قبل می توان نوشت:

اگر $P = P_1, P_2, \dots, P_n$

سلسله عددی

و اگر $n \rightarrow \infty$ آنگاه $E(P_i) \rightarrow 0$ $i \leq n$

$$E(P) = E(P_1) + E(P_2) + \dots + E(P_n) \rightarrow 0$$

یعنی

واحد بندی (ادامه)



در روابط قبل برای سادگی، تلاش لازم برای یکپارچگی (Integration) راه حلها با یکدیگر نادیده گرفته شده است. بنابراین باید نوشت:

$$E(P) > E(P_1) + E(P_2) + E(P_3) + I(E(P_1), E(P_2), E(P_3))$$

I : تلاش لازم برای یکپارچه سازی راه حلها

توجه:

- هنگام شکستن P ، روابط موجود بین زیر مسئله ها نادیده گرفته می شود
- دو مرحله شدن راه حل (شکستن سپس یکپارچگی)

واحد بندی (ادامه)



ویژگیهای واحد بندی:

- اگر شرایط بیان شده در تعریف واحد بندی رعایت گردد آنگاه ماژولهای بدست آمده قابلیت استفاده مجدد بالایی خواهد داشت
- تعداد زیر مساله ها نباید زیاد یا کم باشد
- تعیین معیار شکستن یک مساله مهمترین عامل برای موفقیت استفاده از این ویژگی

واحد بندی (ادامه)



واحدها باید ویژگیهای Building Blocks را داشته باشند:

■ استقلال (Independent)

■ واسطهای خوش تعریف (Well-defined Interfaces)

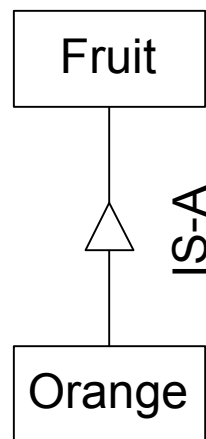
سلسله مراتب (Hierarchy)

”سلسله مراتب عبارت از مرتب ساختن تجربدها در سطوح مختلف“

انواع سلسله مراتب:

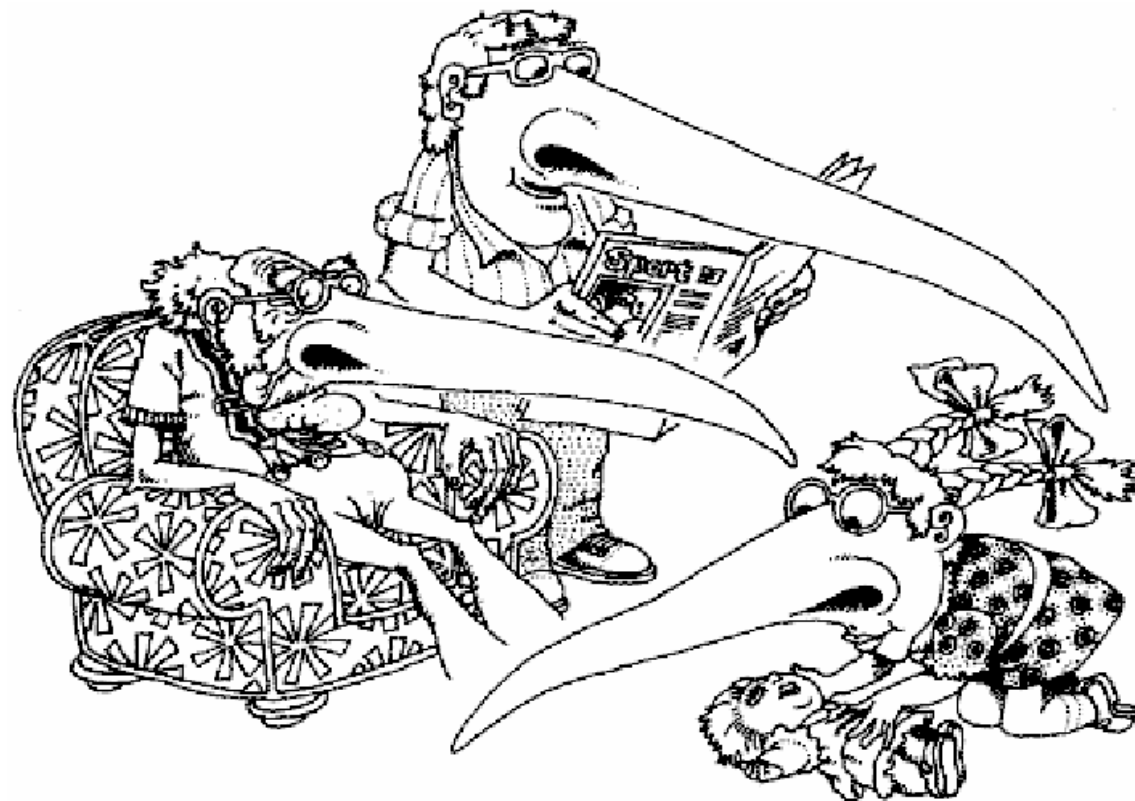
۱ ساختار کلاس (IS-A)

مثال: *Orange IS-A Fruit*



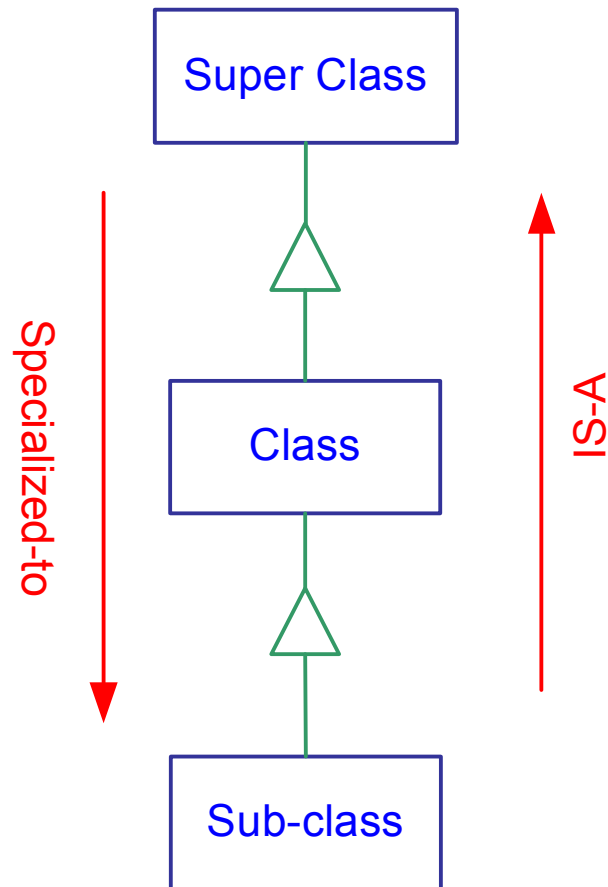
سلسله مراتب (ادامه)

“وراثت (Inheritance) یکی از معروفترین انواع رابطه IS-A”



[Booch94]

سلسله مراتب (ادامه)

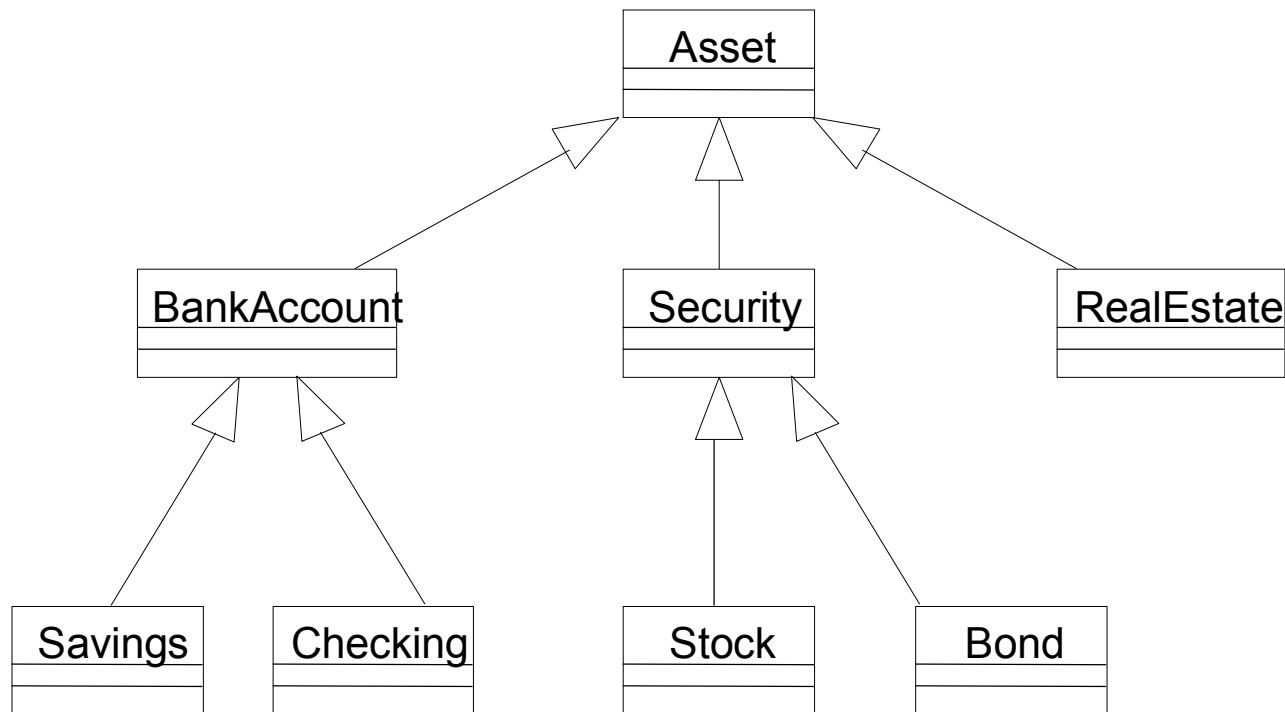


روابط وراثتی کلاس پدر، کلاس، و کلاس فرزند

سلسله مراتب (ادامه)

“سطوح تجرید متفاوت در سطوح مختلف سلسله مراتب نمایان می گردد”

افزایش تجرید

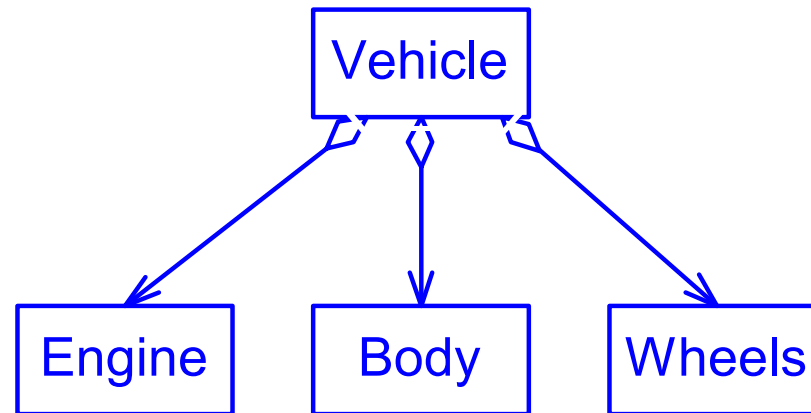


کاهش تجرید

سلسله مراتب (ادامه)



ساختار شیء (PART-OF) ۲



The Vehicle **HAS-An** Engine

The Engine is **PART-OF** Vehicle

سلسله مراتب (ادامه)



نقش سلسله مراتب در کنترل پیچیدگی:

- با سازماندهی تجربدها در سلسله مراتب **PART-OF** و **IS-A** درک ما نسبت به سیستم افزایش می یابد
- اهمیت سلسله مراتب **PART-OF**: روابط موجود بین اشیاء و فعل و انفعالاتی که رخ می دهد را نمایان می سازد
- اهمیت سلسله مراتب **IS-A**: افزونگی موجود در سیستم را مدیریت می نماید (Economy of Expression)

سلسله مراتب (ادامه)



ویژگیهای سلسله مراتب:

- استفاده از وراثت با محصورسازی تام تعارض دارد زیرا مستلزم دسترسی مستقیم کلاس فرزند به بعضی از اعمال و داده های اختصاصی کلاس پدر است

مزایای مدل شئی



۱ هدف نهائی تکنولوژی OO انجام فرایند تولید نرم افزار به صورت مشابه فرایند تولید سخت افزار (فرایند استاندارد و سیستماتیک)

۲ قابلیت پشتیبانی از سیستم های توزیع شده (اشیاء یا مولفه ها روی سایت های گوناگون توزیع می شوند)

۳ ارائه مدل قویتری که پتانسیل مدیریت پیچیدگی کاربردهای امروزی را دارا باشد

مزایای مدل شیء (ادامه)



- ④ کاهش هزینه تولید و نگهداشت نرم افزار بوسیله در نظر گرفتن اشیاء بعنوان واحد مجتمع پذیر تفکیک نشدنی
- ⑤ افزایش مقیاس پذیری و قابلیت توسعه سیستمها بوسیله محصورسازی
- ⑥ استفاده مجدد بوسیله تکنولوژی مولفه ها (COM, .NET, Java Beans) که بر مفاهیم مدل شیء مبتنی است