

مهندسی نرم افزار

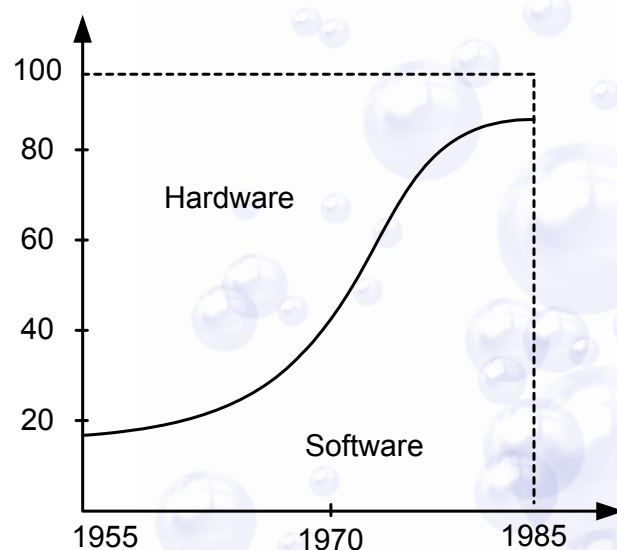
فصل سوم: مفاهیم و اصول متدولوژی شی گرا (Object Oriented)
مدرس: اسماعیل نورانی

اهداف جلسه

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

Abstraction
Encapsulation
Modularity
Hierarchy

- مقدمه
- روشهای طراحی
- تجرید
- دربرگیری
- واحدبندی
- سلسه مراتب
- مزایای مدل شیء و کاربردهای آن



درصد هزینه های نرم افزار در مقابل هزینه های سخت افزار از سالهای 1955 تا 1985

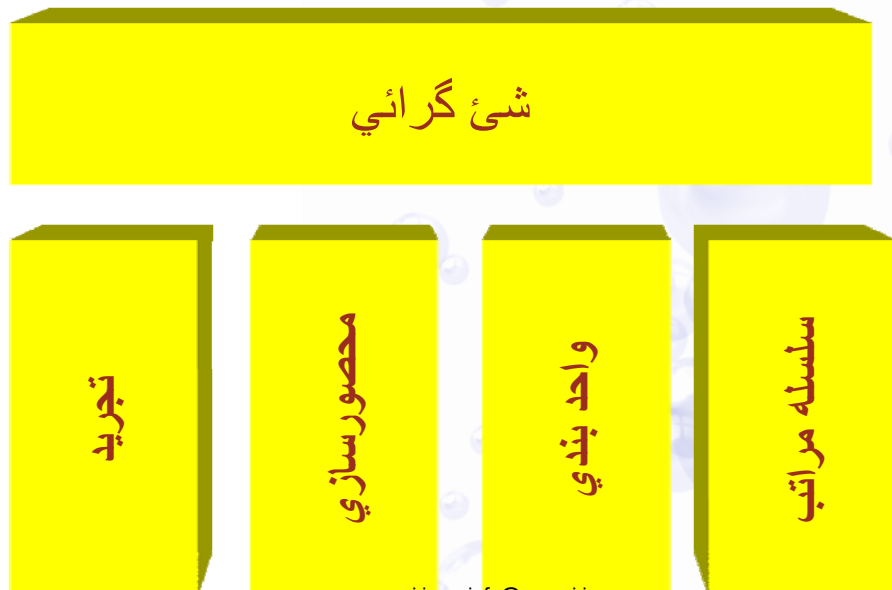


روشهاي طراحی

- طراحی ساخت یافته
 - طراحی شیء گرائی
- Structured Design
Object-Oriented Design



اصول شیء گرائی



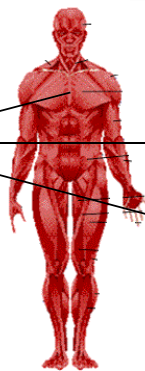
تجريد (Abstraction)

“تجريد عبارتست از فرآيند متمرکز شدن روي
ويژگيها و رفتارهاي اصلي يك پديده، و ناديدنه
گرفتن ويژگيهاي موقت و غير مهم آن پديده، از
يك زاويه ديد مشخص”

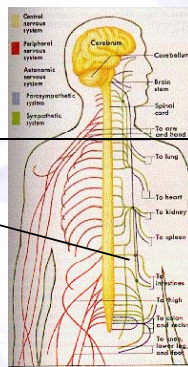
تجريد (ادامه)

مثالهاي از تجريد

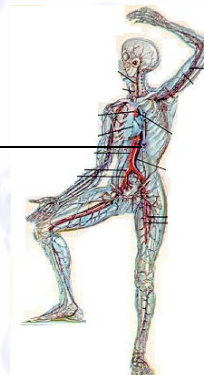
ديدهاي
گونگون از
بدن انسان




عضلات بدن



سيستم عصبي

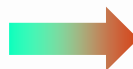


سيستم گردش خون

- بيان روابط میان اجزاء يك سيستم مكانيكي توسط يك معادله رياضي
- استفاده از نماد  براي نمايش حضور موجوديت انسان در يك صحنه
- نمايش گرافيكي رفتار يك سيستم
- يكي از ابزارهاي اصلي كنترل و تسلط بر پيچيدگي

■ تجريد موجوديت (Entity Abstraction)

Real Object: Student



Abstraction: Student

Student
-StudentID : String
-Name : String
-Age : short
-EntryYear : String
+RegisterCourse()
+UnregisterCourse()

ویژگیهای تجريد:

- برای يك شئ تجريدهای گوناگوني وجود دارد
- تجريد با نمود خارجي يك شئ سر و کار دارد
- تجريد سطوحی دارد (میزان پرداختن به جزئیات)
- در شئ گرائی مفهوم تجريد خود را در قالب نوع داده مجرد (Abstract Data Type) نشان می دهد .

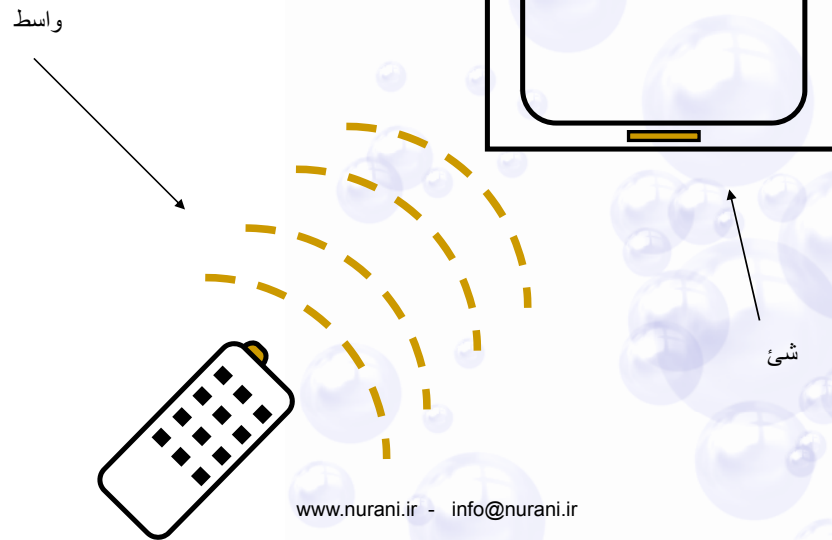
محصور سازی (Encapsulation)

“محصور سازی عبارت از عدم پذیرش تاثیرات ناخواسته (Side Effects) و یا کنترل نشده و محدود کردن طرق دسترسی به یا استفاده از يك شئ”
با توجه به این اصل هر شئ از دو مولفه زیر تشکیل می گردد:

- 1- واسطه (Interface): توصیفی از سرویسهایی که این شئ در اختیار Client ها قرار می دهد.

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

محصور سازي (ادامه)



www.nurani.ir - info@nurani.ir

13

محصور سازي (ادامه)

نقش محصور سازي در کنترل پيچيدگي:

- کنترل و تسلط بر پيچيدگي بوسيله کنترل راه هاي دسترسي به يك شيء كه باعث جلوگیری از خرابكاريهاي احتمالي و محلي كردن گستره خطاها در خود شيء
- با ثبات واسط يك شيء، مي توان هر تغييری در پياده سازي آن شيء انجام گیرد
- مفهوم يك واسط، چند پياده سازي امکان استفاده مجدد را بالامي برد



Slide 14 of x

محصول سازي (ادامه)

قاعده باز و بسته:

“هر ماژولي براي تعريف کننده آن باز و براي استفاده کننده بسته است”

ماژول باز: ماژولي که براي اعمال تغييرات آماده باشد
ماژول بسته: ماژولي که امکان تغييرات در آن وجود ندارد

“ما به ماژولهايي نياز داريم که همزمان باز و بسته باشند”

محصول سازي (ادامه)

- ارتباط بين اشياء تنها از راه واسطها باشد.
- تجريد مکانيزم تعيين جزئیاتی است که باید پنهان شود، است. اما محصول سازي، فرایند پنهان سازي جزئیات و کنترل دسترسی به آن خواهد بود.

واحد بندي (Modularity)

“سیستمي را واحد بندي شده مي گویند که به مجموعه اي از ماژولهاي (واحدها) منسجم و معني دار که وابستگي بين آنها حد اقل است تجزيه شده باشد”

- ماژولها: واحد تشکیل دهنده ساختار فیزیکی سیستم نرم افزاری (شبيه مدارات مجتمع در سخت افزار)

مثالي از واحدها:

- فایلها در C و C++
- مولفه ها (Components) در استانداردهاي COM ، Java Beans و .NET.

واحد بندي (ادامه)

- انسجام (Cohesion): انسجام عبارتست از درجه ارتباط عملکردهاي عناصر داخلي يك ماژول
- وابستگي (Coupling): وابستگي عبارتست از درجه ارتباط ماژولهاي گوناگون با یکدیگر



واحد بندي (ادامه)

Order Processing System



Order Fulfillment

Order Entry

Billing

مثال:



واحد بندي (ادامه)

نقش واحد بندي در کنترل پیچیدگی:

■ شکستن مساله به اجزائي کوچکتر يکي از راههاي کارا براي مقابله با پیچیدگی

مثال: اگر مسئله P را به زیر مسئله هاي P1، P2 و P3 تقسیم نمایيم آنگاه

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

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

C: Complexity

E: Solving Energy

واحد بندي (ادامه)

بنابر استدلال قبل مي توان نوشت:

اگر $P = P_1, P_2, \dots, P_n$

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

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

واحد بندي (ادامه)

در روابط قبل براي سادگي، تلاش لازم براي يکپارچگي (Integration) راه حلها با يکديگر نادیده گرفته شده است. بنابر اين بايد 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 براي يکپارچه سازي راه حلها

توجه:

- هنگام شکستن P ، روابط موجود بين زیر مسئله ها نادیده گرفته مي شود
- دو مرحله شدن راه حل (شکستن سپس يکپارچگي)

ویژگیهای واحدبندی:

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

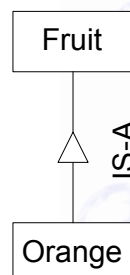
سلسله مراتب (Hierarchy)

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

انواع سلسله مراتب:

① ساختار کلاس (IS-A)

مثال: **Orange IS-A Fruit**





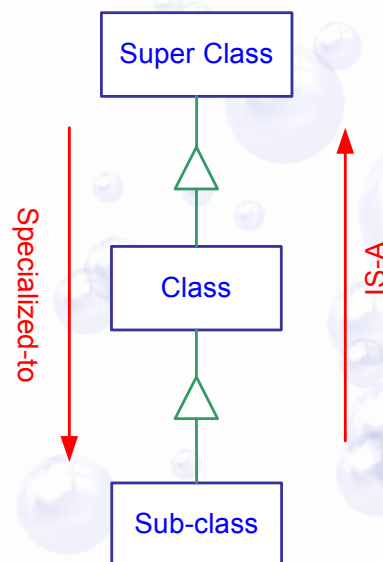
سلسله مراتب (ادامه)

“وراثت (Inheritance) يكي از معروفترين انواع رابطه IS-A”



سلسله مراتب (ادامه)

روابط وراثتي کلاس پدر، کلاس، و کلاس فرزند



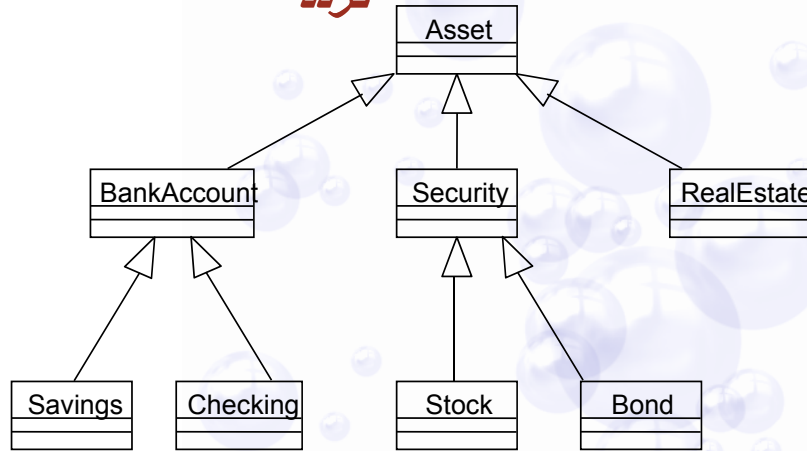


سلسله مراتب (ادامه)

“سطوح تجريد متفاوت در سطوح مختلف سلسله مراتب نمايان مي گردد”
افزايش تجريد

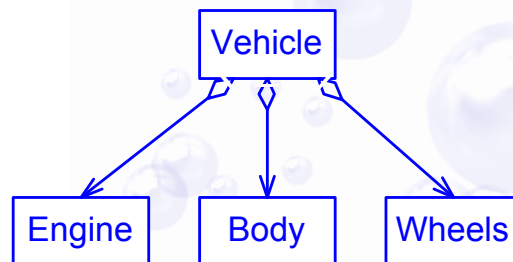


كاهش تجريد



سلسله مراتب (ادامه)

ساختمان شیء (PART-OF)



The Vehicle **HAS-AN** Engine
The Engine is **PART-OF** Vehicle

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)

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 شیء

- ① هدف نهایی تکنولوژی OO انجام فرایند تولید نرم افزار به صورت مشابه فرایند تولید سخت افزار (فرایند استاندارد و سیستماتیک)
- ② قابلیت پشتیبانی از سیستم های توزیع شده (اشیاء یا مولفه ها روی سایت های گوناگون توزیع می شوند)
- ③ ارائه مدل قویتری که پتانسیل مدیریت پیچیدگی کاربردهای امروزی را دارا باشد

مزایای مدل شیء (ادامه)

- ④ کاهش هزینه تولید و نگهداشت
- 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