

بسمه تعالی

فصل چهارم

روشهای شناسائی و یافتن کلاسها

مدرس: فریدون شمس

اهداف جلسه



- درک اهمیت طبقه بندی در OOAD
- آشنائی با روشهای یافتن کلاسها
- درک تفاوت بین روشهای مبتنی بر داده و مبتنی بر وظیفه
- آشنائی با مدلسازی بوسیله کارتهای CRC
- آشنائی با لایه بندی و نقش آن
- آشنائی با مقوله بندی و نقش آن
- کارتهای CRC و مقوله بندی



Classification

- طبقه بندی

- روشهای مبتنی بر داده-Data-Driven

- روشهای مبتنی بر وظیفه-Responsibility-Driven

- روشهای شناسائی کلاسهای اولیه

- روش CRC

Layering

- لایه بندی

Stereotyping

- مقوله بندی

طبقه بندی

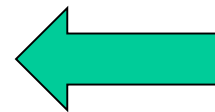
(Classification)

” طبقه بندی ابزاری است که بوسیله آن، ما انسانها،
دانش خود را مرتب می نماییم “

- مسئله طبقه بندی در همه رشته های علوم محض، کاربردی و مهندسی مطرح است

۱) کلاسهای مناسب یا
کلیدی

۲) غیر مناسب یا کم
اهمیت



بوسیله طبقه
بندی

کلاسهای مطرح برای
مدلسازی یک سیستم

طبقه بندی

(ادامه)

ویژگیها ها:

۱) هر طبقه بندی با توجه به معیاری (یا معیارهایی) انجام می گیرد.

طبقه بندی های



معیارهای متفاوت
مختلف

۲) طبقه بندی ایده آل (یعنی بهترین طبقه بندی بدون توجه به شرایط موجود) وجود ندارد!

۳) فرآیند طبقه بندی یک فرآیند افزایشی و تکراری

طبقه بندی

(ادامه)



ویژگیها های طبقه بندی در OOAD:

۱) کلاسهایی باید انتخاب شوند که با توجه به معیارها و محدودیت های موجود (اقتصادی، تکنولوژی، و ...) سازگار باشند.

۲) نباید دنبال راه حل پلائی بلکه باید بدنبال راه حل مناسبتر باشیم!

طبقه بندی

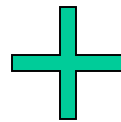
(ادامه)



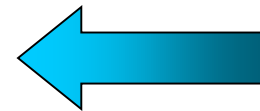
۳) طبیعت افزایشی و تکراری فرآیند طبقه بندی به صورت زیر خود را نشان می دهد:



بهبود کلاسهای اولیه



شناسائی کلاسهای دیگر

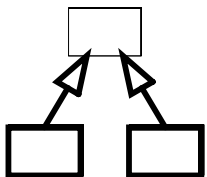


افزایش درک مسئله + تلاش برای تشریح

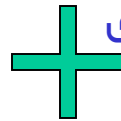
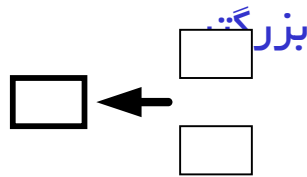
تعدادی کلاس کاندیدا (اولیه)



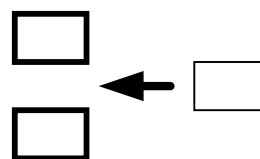
ایجاد زیرکلاس های جدید



ترکیب تعدادی کلاس کوچکتر برای تشکیل دادن یک کلاس بزرگتر



ساختار و رفتار تقسیم یک کلاس بزرگتر به تعدادی کلاس کوچکتر



طبقه بندی

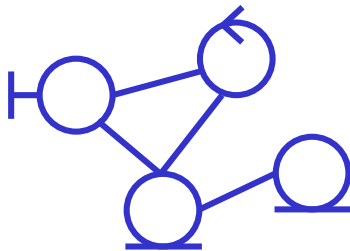
(ادامه)



دو منبع اصلی برای تشخیص کلاسها:

Problem Space (۱) فضای مسئله

Solution Space (۲) فضای راه حل



← مدل تحلیل

توصیف فضای مسئله

← مدل طراحی

طبقه بندی

(ادامه)



فرآیند شناسایی کلاسها شامل دو فعالیت اکتشاف
(Discovery) و ابداع (Invention) است



ابداع کلاسهایی برای پیاده
سازی کلاسهای موجود



اکتشاف کلاسهای
موجود

مثال: ساختمان داده
ها (لیست، آرایه،
درخت، ...)، ارتباط با پایگاه
داده ها، کلاسهای کنترلی و
هماهنگ کننده، ...

یافتن کلاسها



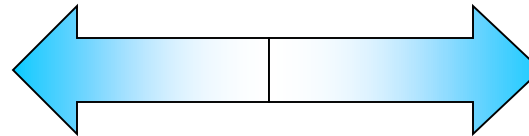
رهیافت ها:

(۱) روشهای مبتنی بر داده **Data Driven Approach**

(۲) روشهای مبتنی بر رفتار **Responsibility Driven Approach**

Responsibilities

اولویت با وظایف



Data

اولویت با داده ها

روشهای مبتنی بر

داده

” در این روشها **مبنای شناسائی کلاسهای مناسب**
سیستم، شناسائی ساختار داده های مورد نیاز هر
کلاس است“

فرآیند تعیین کلاسها با پرسیدن دو سوال زیر صورت می
گیرد:

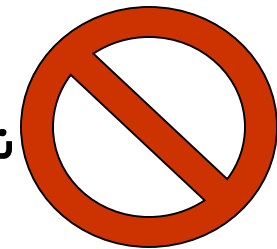
- ۱ - ساختار هر کلاس چیست؟
- ۲ - چه عملیاتی بوسیله هر کلاس انجام می گیرد؟

روشهای مبتنی بر داده

(ادامه)



نقض اصل محصورسازی و هدف از واسط



پیشنهادی را معین کنید

تعیین کنید

سرویس گیرندگان به ساختار داخلی
کلاس وابسته خواهند شد

سرویس های کلاس به ساختار
داخلی آن وابسته خواهند بود

روشهای مبتنی بر داده

(ادامه)



مزیت:

- سهولت یادگیری مخصوصا برای کسانی که پیش زمینه ای در روشهای ساخت یافته و طراحی مبتنی بر داده ها را دارند

روشهای مبتنی بر وظیفه



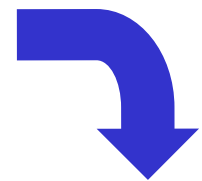
” در این روشها **مبنای شناسائی کلاسهای مناسب سیستم، شناسائی مسؤلیتهای (Responsibilities)** مورد نیاز هر کلاس است“

فرآیند تعیین کلاسها با پرسیدن دو سوال زیر صورت می گیرد:

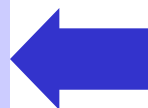
۱- هر کلاس چه **مسئولیتی** دارد؟ (چه عملیاتی بوسیله این کلاس انجام می گیرد؟)

۲- این شیء، چه اطلاعاتی با بقیه اشیاء **به اشتراک** می گذارد؟

روشهای مبتنی بر وظیفه (ادامه)



سرویس گیرندگان، **مستقل** از ساختار داخلی کلاس خواهند شد



سرویس های کلاس به ساختار داخلی آن وابسته **نخواهند بود**

روشهای مبتنی بر وظیفه (ادامه)



”نگاه اصلی این روش به هر کلاس عبارت از موجودیتی است که در هر آن می تواند نقش سرویس دهنده یا نقش سرویس گیرنده را ایفا نماید“

هر کلاس در نقش سرویس دهنده می تواند فراهم کننده خدمات برای ۳ نوع سرویس گیرنده باشد:

۱- سرویس گیرندگان خارجی **External Clients**

۲- سرویس گیرندگان مشتق شده **Derived Clients**

۳- خود کلاس **Self Client**

روشهای مبتنی بر وظیفه (ادامه)



مزیت:

- بزرگترین مزیت این روش به حد اکثر رساندن محصورسازی در سطح طراحی است که باعث افزایش قابلیت نگهداری و انعطاف پذیری سیستم نسبت به تغییرات آتی خواهد گردید.

روشهای شناسائی کلاسهای اولیه



۱) طبقه بندی های پیشنهاد شده بوسیله متدولوژیهای OO

۲) تحلیل دامنه Domain Analysis

۳) تحلیل موارد کاربری Use Case Analysis

۴) تحلیل لغوی صورت مساله Problem Statement Analysis

۵) استفاده از الگوها Patterns

۶) کارتهای CRC

طبقه بندی های پیشنهاد



منابع **بالقوه** زیر برای **شناسائی کلاسها** پیشنهاد می

شوند: **۱) دستگاه ها (Devices):** دستگاه هایی که برنامه با آنها تعامل دارد.

۲) نقشها: نقشهای گوناگون که کاربر در تعامل با سیستم ایفا می نماید.

۳) محلهای فیزیکی (مانند دفاتر، سایتها، ...) که برای سیستم مهم هستند.

۴) سازمانها: مجموعه های سازماندهی شده (از مردم، منابع، ابزار، ...) که دارای ماموریتهای مشخصند.

طبقه بندی های پیشنهاد شده...

(ادامه)



۵) مفاهیم منطقی: اصول و ایده های منطقی که در منطق کاری سازمان بکار گرفته می شوند.

۶) ساختار: همان روابط **IS-A** و **PART-OF**.

۷) دیگر سیستمها: سیستمهای خارجی که برنامه با آنها تعامل دارد

تحلیل دامنه (Domain Analysis)



” عبارتست از شناسایی کلاسها و اشیاء مشترک در همه برنامه های کاربردی (*Applications*) متعلق به یک دامنه مشخص (مانند کامپایلرها، سیستمهای اطلاعاتی جغرافیایی، ...)“

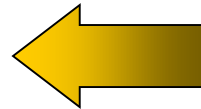
در این روش، با دیدن سیستمهای مرتبط و اسناد آنها و با صحبت با کارشناسان خبره در زمینه سیستم مورد نظر می توان کلاسهای کلیدی یک سیستم را حدس زد.

تحلیل موارد کاربری



”دنباله ای از عملیات است که یک سیستم انجام می دهد تا یک نتیجه قابل مشاهده و ارزشمند برای کاربر فراهم نماید“

کلاسهای کلیدی



موارد کاربری

تجزیه و
تحلیل

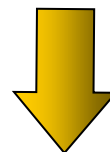
تحلیل لغوی صورت مساله



”با تحلیل صورت مکتوب مساله می توان کلاسهای اولیه را بدست آورد“

گام ۲: حذف
نامها و فعلهای
غیر ضروری

گام ۱: تعیین
نامها و فعلهای
موجود



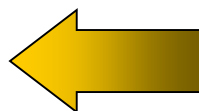
فعلها: عملیات
کاندیدا

نامها: کلاسهای
کاندیدا

استفاده از الگوها

” یک الگو، یک مساله طراحی که در یک زمینه مشخص مرتباً تکرار می گردد را توصیف کرده و سپس یک راه حل کلی و تکرار پذیر برای آن ارائه می کند“

کلاسهای کلیدی



گزینش

کاتالوگی از
الگوهای
استاندارد

کارت‌های CRC



” روشی غیر رسمی (*Informal*) برای شناسایی و توصیف کلاسها، رفتار و مسئولیتهای آنها و همکارانی (کلاسهای دیگر) که به کمک آنها وظایف خود را انجام می‌دهد، می‌باشد“

Class

Responsibilities

Collaborators

کلاس (رده)

مسئولیتها

همکاران

CRC

کلمه

کارت‌های CRC (ادامه)



Class Name	Collaborators
Responsibilities	

کارت‌های "۲×۳"

اسم کلاس :

مسئولیتها :

همکاران :

دارا بودن مجموعه ای از اطلاعات لازم و کافی (به صورت *Abstract*) از هر عنصر در یک سیستم:

- در چه رده ای ؟ (Class)
- با چه وظایفی؟ (Responsibilities)
- با چه همکارانی؟ (Collaborators)

کارت‌های CRC (ادامه)



هدف:

- فراهم نمودن روشی برای آموزش مفاهیم شیء گرا (کمک به پیاده نمودن شیوه ای که در آن اشیاء محور می باشند)
- حل مشکل شناسایی اشیاء برای مسائلی که آشنایی زیادی در مورد آن نداریم

کارت‌های CRC (ادامه)



ویژگیها:

- سادگی روش : بر اساس یک بازی ساده با کارت‌ها



- طبیعی بودن روند کار و نمایش سناریوهای واژه *What if...*



فرایند‌گرایی بر اساس کار گروه

کارت‌های CRC (ادامه)



فرآیند مدلسازی بوسیله کارت‌های CRC:

۱- موارد کاربری کلیدی سیستم را مرور کنید.

۲- در صورت نیاز یکی (یا ترکیبی) از تکنیک‌های یافتن کلاس‌های اولیه را برای شناسائی مجموعه ای از کلاس‌های کاندیدا بکار ببرید.

۳- به ازای هر مورد کاربری گام‌های زیر را انجام دهید:

الف) از کلاس‌های موجود، کلاس‌هایی که مناسب این مورد کاربری است را مشخص نمایید.

ب) اگر کلاس مناسبی وجود نداشته باشد پس کلاس جدیدی

کارت‌های CRC (ادامه)



ج) مسؤلیتهای کلاس را تشخیص دهید:

- این کلاس باید چه وظیفه ای را انجام دهد؟
- اگر وظیفه ای را در اختیار دارید، این وظیفه متعلق به کدام کلاس است؟
- بعضی از مسؤلیتها بوسیله همکاری کلاس با دیگر کلاسها انجام پذیرند. بنابراین عجله نکنید.

کارت‌های CRC (ادامه)



د) همکاران کلاس را تشخیص دهید:

- سناریوی “**What if...?**” را اجرا کنید.
- همکاری هنگامی رخ می‌دهد که یک کلاس نیازمند اطلاعاتی باشد که در اختیار ندارد.
- همکاری هنگامی رخ می‌دهد که یک کلاس نیازمند به روز رسانی اطلاعاتی باشد که در اختیار ندارد.
- در هر همکاری، حد اقل یک کلاس آغاز کننده باید وجود داشته باشد

کارت‌های CRC (ادامه)



ه) کارت‌های CRC را دور میز چرخش دهید:

- کارت‌های کلاس‌هایی که با یکدیگر همکاری دارند نزدیک هم قرار دهید.

- هرچه همکاری قویتر باشد نزدیکی دو کلاس به یک دیگر می‌بایست بیشتر باشد.

- کارت‌های پر(شلوغ) را در وسط میز قرار دهید.

- کارت‌ها را دور میز بچرخانید.

۳۲ از کسانی که در جلسه شرکت دارند بخواهید که به کارت‌های در

حال چرخش توجه نمایند.

کارت‌های CRC (ادامه)



- حاضرین در جلسه ارتباطات جدیدی بین کلاسها تشخیص خواهند داد .
- سناریوی "چه می‌شود اگر...؟" را اجرا نمایید.

کارت‌های CRC (ادامه)

مثال: ماشین ATM

A cash machine is made of several devices: a display screen, a card reader, a numeric key pad and a cash dispenser. The user inserts a card which is read by the card reader. If it can't be read, the card is ejected and an error message is displayed on the screen. If it can be read, the user is prompted to enter their PIN. If their PIN not the same as one encoded on the card then the user is given two more chances before the card is 'swallowed'. Once the correct PIN is entered, the user is prompted to input how much money they require. If there are sufficient funds in the account (and sufficient resources in the machine), the money is returned by the cash dispenser (and the account and resources suitably updated). Otherwise a suitable message is displayed.

کارت‌های CRC (ادامه)



CRC cards showing collaborators.

cashMachine	
handleTransaction updateResources knowResources	cardReader cashDispenser

displayScreen	
displayText	

cardReader	
readCard ejectCard getValidCard displayUnreadable promptPIN swallowCard	displayScreen displayScreen numericKeyPad

account	
knowAccount£ updateAccount displayInsufficient£	displayScreen

numericKeyPad	
getInput	

device	

cashDispenser	
prompt£ dispense£	displayScreen account cashMachine

کارت‌های CRC (ادامه)

مثال: مسئله Producer/Consumer

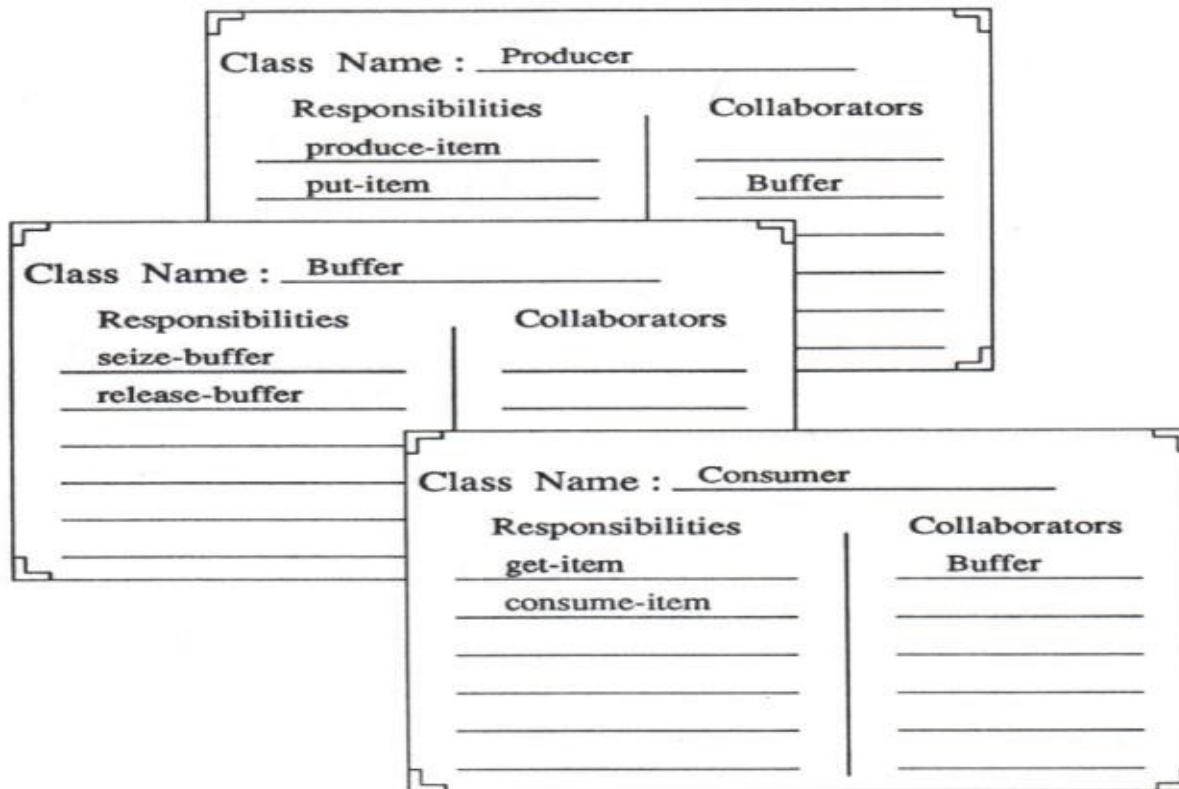


Figure 5.1: The CRC Cards for Producer-Consumer Process Example

کارت‌های CRC (ادامه)

مثال:

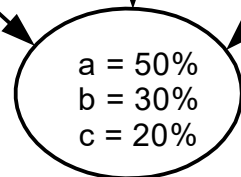
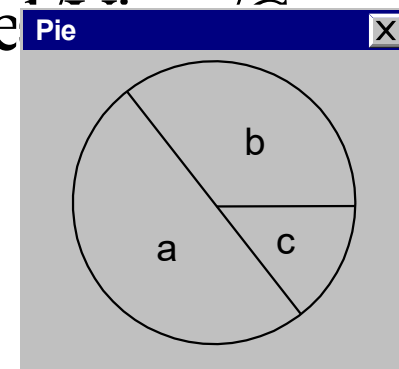
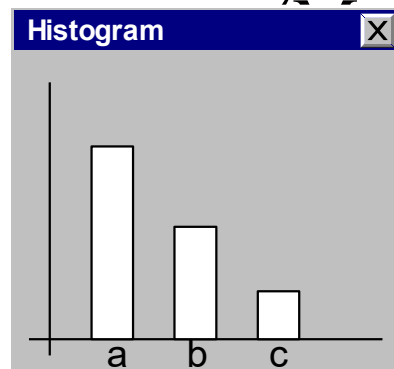
The screenshot shows the Microgold CRC Pro interface with three overlapping class cards. The top card is for 'Vehicle', the middle for 'Car', and the bottom for 'Passenger'. The 'Car' card shows it inherits from 'Vehicle'. The 'Passenger' card shows it inherits from 'Person' and lists responsibilities 'Drive' and 'BuckleUp', and a collaborator 'Car'.

Class	Superclasses	Subclasses	Responsibilities	Collaborators
Vehicle				
Car	Vehicle			
Passenger	Person		Drive BuckleUp	Car

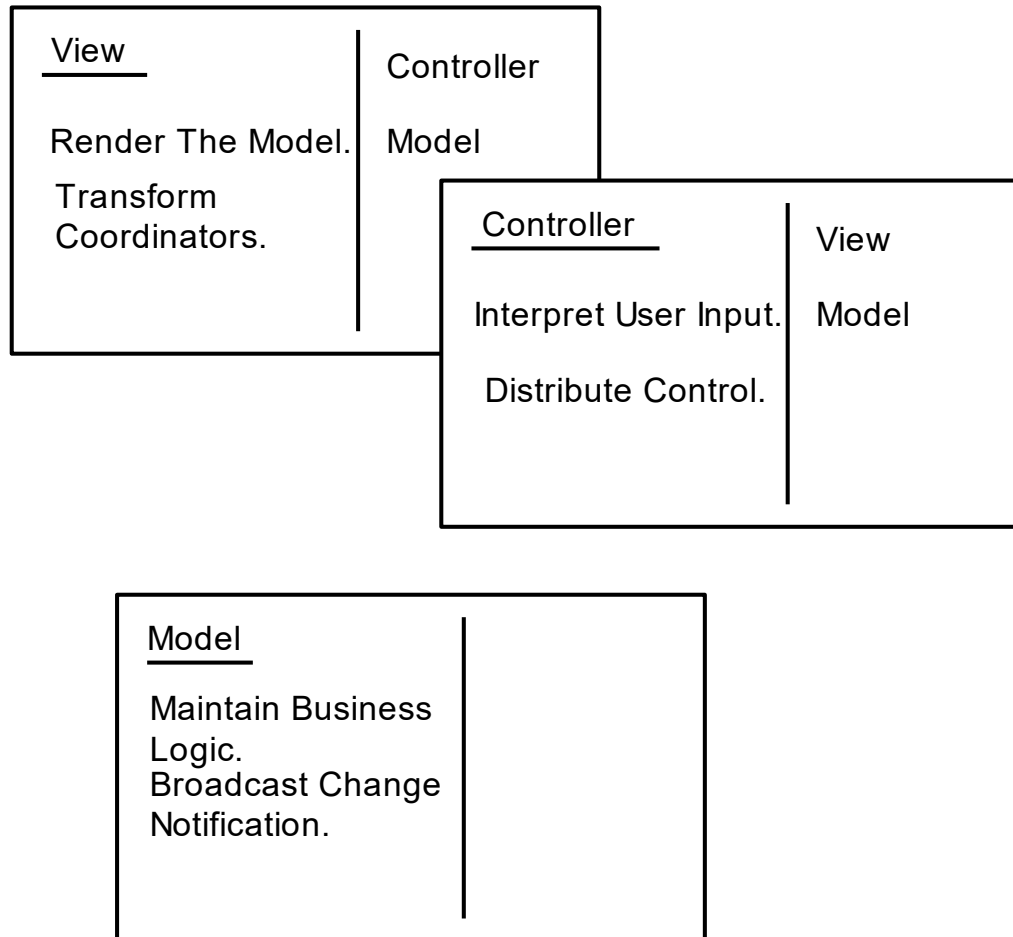
کارت‌های CRC (ادامه)

مثال: الگوی مدل/دید/کنترل کننده

	a	b	c
X	30	70	10
Y	50	30	20
Z	10	80	10



کارت‌های CRC (ادامه)



کارت‌های CRC (ادامه)



مزایا:

- نقطه مناسبی برای شروع تحلیل (جرقه ذهنی) **Brainstorming**
- پوشش جنبه‌های اصلی یک سیستم

(Encapsulation, Instantiation, Communication)

- قابلیت شبیه‌سازی رفتار سیستم (مرور سناریو)
- قابلیت انتقال حرکت در اطراف و تجمع
- با نمودار کلاسها سازگار است

کارت‌های CRC (ادامه)



• بیان معماری یک سیستم

▪ در اختیار گرفتن جوهر نرم افزار **Essence**

▪ استفاده از ایده **Client/Server**

▪ نداشتن مشکل سیم بندی **Wire Syndrome**

▪ مقوله بندی **Stereotyping**

• امکان کار گروهی

• ۴۱ تحلیل بوسیله خبرگان

کارت‌های CRC (ادامه)



معایب:

- مشکل برقرای ارتباط با کاربران
- کارت‌های CRC تنها **بخشی** از نیازمندیهای یک سیستم شئ گرا را تشکیل می‌دهند

لایه بندی و مقوله

بندی

” یک سیستم پیچیده ابعاد گوناگونی دارد لذا باید از زوایای متفاوتی به آن نگریست“

مثال: ساختمان

یک ساختمان معمولاً از دید مشتری دارای یک بعد می باشد. اما معمار، ساختمان را از یک یا چند زاویه مختلف برای

مشتری تصویر می



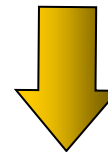
لایه بندی و مقوله بندی

(ادامه)



”یک سیستم نرم افزاری نیز همانند یک ساختمان یک
موجودیت واحد است، اما برای تولید و توسعه آن،
توصیف کاملی از سیستم نیاز است که آن را از زوایای
گوناگون مورد بررسی قرار می دهد“

هرکدام از سهامداران به سیستم از یک زاویه دید معین می نگرند



کاربر نهائی، مشتری، مدیر پروژه،
تحلیلگر، طراح، معمار، برنامه نویس، ...

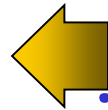
لایه بندی و مقوله بندی

(ادامه)



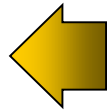
در **توصیف** یک سیستم نرم افزاری **دو اصل**
وجود دارد:

لایه بندی



(۱) از دید چه کسی این سیستم توصیف می
گردد؟

مقوله بندی



(۲) میزان پرداختن به جزئیات چه قدر است؟

لایه بندی و مقوله بندی

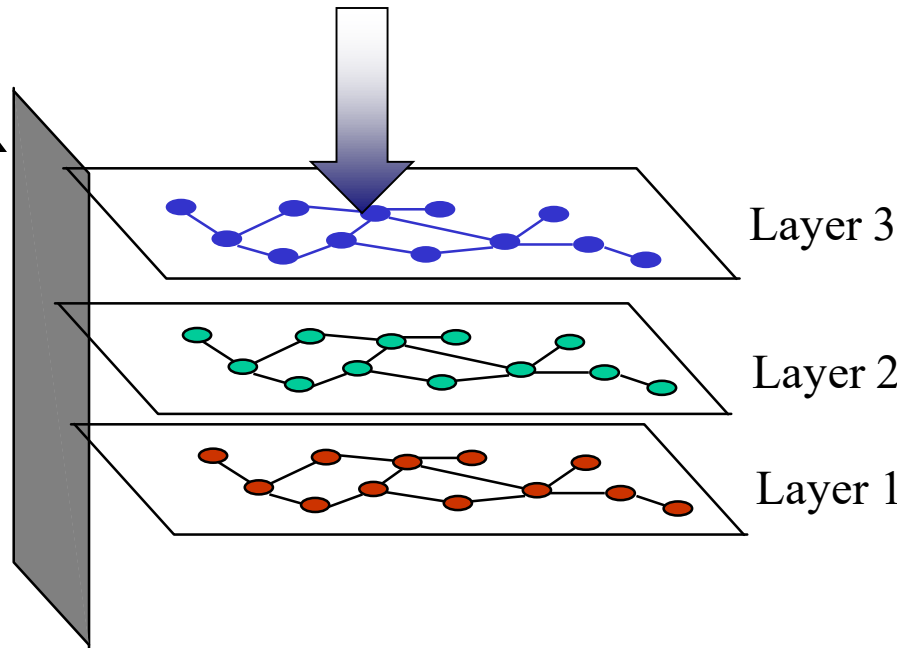
(ادامه)



در این لایه چه گروه هایی از کلاسها را می توان یافت؟

مقوله بندی

سیس
تم



لایه بندی

لایه بندی (Layering)



” در لایه بندی یک سیستم نرم افزاری به صورت
تعدادی از لایه ها تقسیم بندی می گردد “

لایه بندی وابستگی ها را کاهش می دهد بطوریکه
لایه های پایین تر از جزئیات و واسطهای لایه های
بالتر اطلاعی ندارند.

لایه بندی (ادامه)



یک برنامه کاربردی از نظر منطقی به سه لایه کلی تقسیم می شود:

User Interface
Business Logic
Data Services

(۱) واسط کاربر
(۲) منطق کاری
(۳) سرویسهای داده ای

لایه بندی

(ادامه)



با توجه به تقسیم بندی قبل ۳ معماری وجود دارد:

۱) معماری متمرکز

۲) معماری Client/Server

۳) معماری 3-Tier

معماری متمرکز



” در این معماری بخشهای سه گانه برنامه **شدیدا** با یکدیگر **آمیخته اند**. در واقع، **تنها یک لایه وجود دارد**“
مزایا:

(۱) سادگی در طراحی و پیاده سازی

معایب:

(۱) عدم قابلیت استفاده مجدد

(۲) عدم انعطاف پذیری، سختی استفاده و قابلیت
ضعیف تغییر

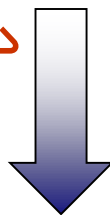
معماری متمرکز (ادامه)



معایب: (ادامه)

۳) قابلیت توسعه پذیری و مقیاس پذیری ضعیف

در صورت نیاز به قدرت
محاسباتی بیشتر



پدیده جزایر برنامه های
کاربردی

(Applications Islands)

معماری Client/Server



” در این معماری بخشهای سه گانه برنامه در **دو لایه** اصلی جا می گیرند.“

لایه ها:

(۱) سرویس گیرنده: واسط کاربر + بخشی از منطق کاری

(۲) سرویس دهنده: سرویسهای داده ای + قسمت اعظم منطق کاری

معماری Client/Server (ادامه)



مزایا:

۱) واسط کاربر قابل استفاده مجدد است.

۲) استفاده از مدل برنامه نویسی مبتنی بر رویداد

۳) وجود محیطهای برنامه سازی یکپارچه و قوی مانند **Delphi**، **Visual Basic**، و **PowerBuilder** که امکان دسترسی به داده های قدیمی و جدید سازمان به صورت یکپارچه را فراهم می کنند

۴) امکان استفاده مجدد جزئی از منطق کاری در قسمت

معماری Client/Server (ادامه)



معایب:

۱) محل منطق کاری مشخص نیست

۲) تنها بخشی از منطق کاری قابل استفاده مجدد است

۳) منطق کاری قابل استفاده مجدد، معمولاً، به صورت روال های ذخیره شده که متعلق به یک پایگاه داده معینی است، وجود دارد. به علت این وابستگی منطق کاری قابلیت استفاده مجدد را در سطح تمام سازمان نخواهد داشت.

معماری 3-Tier



” در این معماری بخشهای سه گانه برنامه در سه لایه اصلی جا می گیرند.“

لایه ها:

۱) واسط کاربر

۲) منطق کاری

۳) سرویسهای داده ای

منطق کاری بر روی سرویس دهنده مستقلی که به سرویس دهنده کاربردی

(Application Server) معروف است، قرار می گیرد.

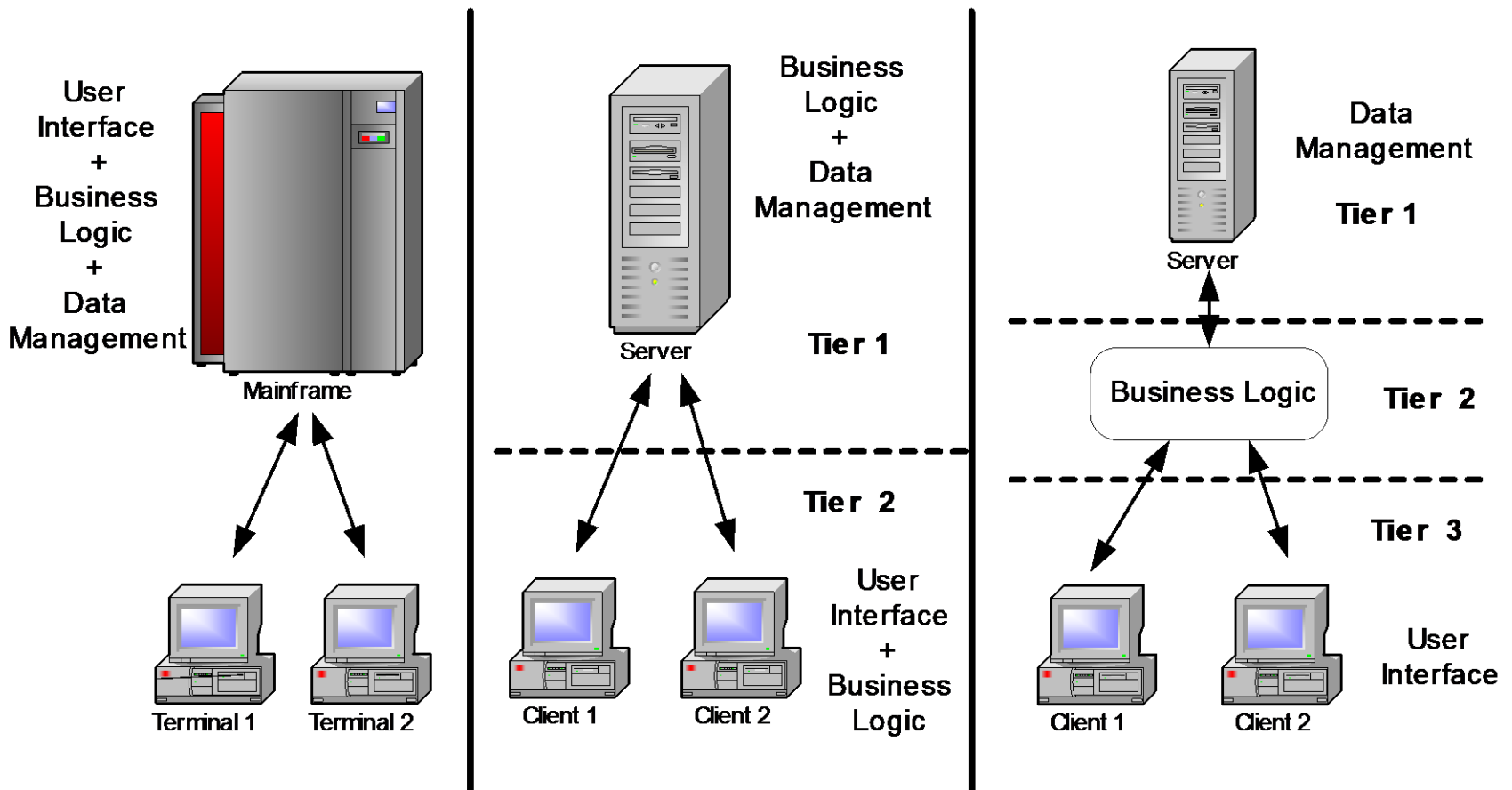
معماری 3-Tier (ادامه)



مزایا:

- ۱) معماری ۳-لایه ای از تقسیم بندی منطقی برنامه تبعیت می کند
- ۲) جداسازی منطق کاری از بقیه قسمت‌های برنامه باعث به **معیب: اکثر رساندن قابلیت استفاده مجدد آن است.**
- ۱) پیچیدگی در طراحی و پیاده سازی.

معماری 3-Tier (ادامه)



معماریهای متمرکز، سرویس گیرنده/سرویس دهنده، و ۳-لایه

معماری 3-Tier (ادامه)



مثال: معماری متدولوژی شیء گرای *Select Perspective*

۱) فرآیندهای کاری

۲) سرویسهای کاربر

۳) سرویسهای کاری

۴) سرویسهای داده ای



مقوله بندی (Stereotyping)

”مقوله بندی اشیاء به ما کمک می کند که سطوح تجربیدی در یک سیستم نرم افزاری را بشناسیم و بدین صورت تمام افراد تیم می توانند در یک سطح واحدی از تجرید فعالیت نمایند“

مقوله بندی روشی برای میزان کردن سطح تجرید است

(Abstraction Level Tuner)

مقوله بندی

(ادامه)



مثال: مقوله بندی در متدولوژیهای

Unified Software Development Process (USDP) (۱)

Select Perspective (۲)

Responsibility Driven طراحی مبتنی بر مسئولیتها (۳)
Design

مقوله بندی در USDP



در USDP کلاسها به سه گروه زیر تقسیم می شوند:

۱) کلاسهای مرزی **Boundary Classes**

مثال: واسط کاربر، پروتکل‌های ارتباطی (مانند TCP/IP)، واسط چاپگر، احساسگرها (Sensors)، و ترمینالهاست.

۲) کلاسهای کنترلی **Control Classes**

۳) کلاسهای موجودیتی **Entity Classes**

کلاسهای موجودیتی مفاهیم کلیدی در سیستم در حال توسعه را نمایش می

مقوله بندی در Perspective



در Perspective کلاسها به سه گروه زیر تقسیم می شوند:

Interface Classes

۱) کلاسهای واسط

مثال: در سیستم هتل داری : اپراتوری که با سیستم کار می کند با فرم هایی(همان اشیاء واسط) ارتباط دارد مانند: فرم رزور اتاق ، فرم حساب شخص ، گزارشی از تعداد افرادی که رزور کرده اند ، و ...

Check Out
Form

Check In
Form

Reservation
Form

Booking
Report

مقوله بندی در Perspective (ادامه)



Business Classes

(۲) کلاسهای کاری

مثال: در سیستم هتل داری، اشیائی مثل رزرو کردن (Reservation) و درخواست مشتری (Customer Order)، فاکتور (Invoice)، حساب (Account)، و... مشاهده می شوند که سرویسهای مورد انتظار از سیستم را فراهم می نمایند

Customer
Order

Reservation

Invoice

Account

مقوله بندی در Perspective (ادامه)



Data Classes

۳) کلاسهای داده ای

مثال: در سیستم هتل داری ، اشیائی مانند: مدیر داده های حساب (Account Data Manager)، تبدیل کننده داده های حساب (Account Data Conversion)، دستیابی به داده های حساب (Account Data Access)، و ... مشاهده می شوند که سرویسهای داده ای مورد انتظار اشیاء دیگر را فراهم می نمایند

Account Data
Conversion

Account Data
Manager

Account Data
Access

مقوله بندی در RDD

در RDD کلاسها به ۶ گروه زیر تقسیم می شوند:

- ۱) کلاسهای کنترل کننده
Controller Classes
- ۲) کلاسهای هماهنگ کننده
Coordinator Classes
- ۳) کلاسهای واسط
Interface Classes
- ۴) کلاسهای فراهم کننده سرویس
Service Provider Classes
- ۵) کلاسهای نگهدارنده اطلاعات
Information Holder Classes
- ۶) کلاسهای ساختاری
Structure Classes

کارت‌های CRC و مقوله بندی



” می‌توان از کارت‌های CRC برای تشخیص طبقه بندی (مقوله بندی) اشیاء استفاده نمود“

مثال: در یک سیستم بانکی مشتری می‌خواهد مبلغی از حساب خود برداشت نماید. این کار بوسیله ماشین پرداخت پول (Automatic Teller Machine) انجام می‌گیرد. کارت‌های CRC این مثال بصورت ذیل است:

کارتهای CRC و مقوله بندی

(ادامه)



Withdrawal Transaction	
Supercalss:Financial Transaction	
Responsibilities	Collaborators
Knows account Knows amount Perfoms Withdraw Logs Transaction Initiates Dispensing cash	Cash Dispenser

کارت‌های CRC و مقوله بندی

(ادامه)



Withdrawal Transaction	
Purpose	Withdraws cash from an account and dispense it
Stereotypes	Service Provider, Coordinator, Business Object

بیشتر کارت CRC، Withdrawal Transaction که در آن طبقه بندی این کلاس دیده می شود