

# Object Oriented Analysis &Design

## 面向对象分析与设计

CRC card:  
Identifying Classes and Objects  
Chapter 14.6

Presenter: **Xiaohong Chen**

陈小红

2020/11/17

# OOAD Review

## What is OOA?

finding and describing the **objects or concepts** in the problem domain

## What is OOD?

defining **software objects** and how they collaborate to fulfill the requirements.

## For example, Airplane example of object and class discovery

### OOA:

in the case of the flight information system, some of the concepts include  
Plane, Flight, and Pilot.

### OOD:

a Plane software object may have a tailNumber attribute and a getFlightHistory method

At last, implement...

domain concept



representation in an  
object-oriented  
programming language

Plane

tailNumber

visualization of  
domain concept

```
public class Plane  
{  
    private String tailNumber;  
  
    public List getFlightHistory() {...}  
}
```

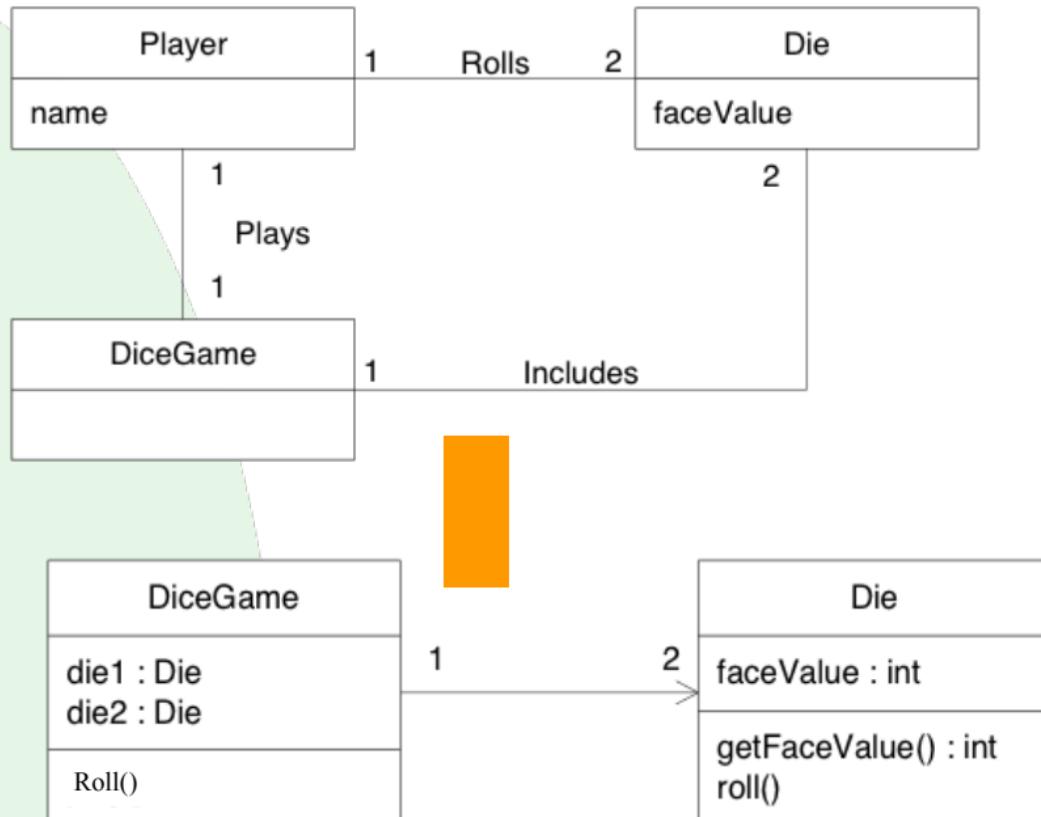
# Review: OOAD Simple Example

## Dice game example

software simulates a player rolling two dice. If the total is seven, they win; otherwise, they lose



# Class diagram



- What's next for design?



methods

# How to find methods in class?

- Part of identifying the classes
  - is the process of *assigning responsibilities* to each class
  - need not described in Domain Model, but in design model
- Every activity that a system must accomplish must be represented by one or more methods in one or more classes
- We generally use **verbs** for the names of methods
- In early stages it is not necessary to determine every method of every class
  - – begin with primary responsibilities and evolve the design

- One way to find class and assign its responsibilities
  - (“Responsibility-Driven Design”)
- From Domain Classes and Use Cases to Design Classes

# 掌握的内容：

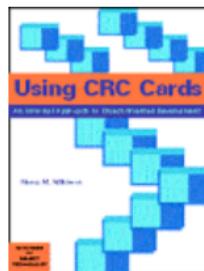
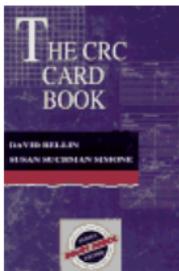
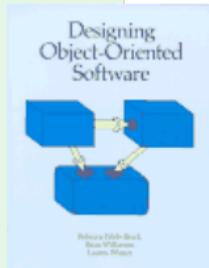
- 会熟练使用CRC进行
  - 类的识别
  - 方法的识别
- 理解CRC在OOAD中的阶段

# CRC

- Stands for:
  - Classes (of objects)
  - Responsibilities (of the objects in each class)
  - Collaborations (with objects in other classes)
  - In UML, these will be examples of “associations”

# Origin of CRC

- Contributors:
  - Kent Beck and Ward Cunningham
    - “A Laboratory for Teaching Object-Oriented Thinking”
    - OOPSLA’89
    - also founders of the ideas of XP and design patterns
  - Rebecca Wirfs-Brock popularized with “Responsibility-Driven Design” (RDD)
- References
  - Rebecca Wirfs-Brock, Brian Wilkerson, Lauren Wiener, *Designing Object-Oriented Software*, Prentice-Hall, 1990.
  - David Bellin and Susan Simone, *The CRC Card Book*, Addison Wesley Longman, 1997. (google 图书)



# The Basic Idea

- CRC cards are paper index cards on which one writes the **responsibilities** and **collaborators** of classes.
- Each card represents **one class**.
- A CRC modeling session involves
  - a group sitting around a table, discussing and writing on the cards as they play "what if" scenarios with the objects, considering what they must do and what other objects they must collaborate with
- A card contains:
  - The **name** of the class.
  - The **responsibilities** of the class.
  - **Collaborations:** other classes with which this class inter-operates, in conjunction with the attendant responsibility.

# A typical CRC card

Class: WHO AM I?	
Responsibilities:	Collaborations:
WHAT SHOULD I DO?	WHO CAN HELP ME?

# Image of CRC cards

		Super/sub-class are optional
Class Name	super class sub-classes	
Responsibilities	Collaborations	
_____	_____	
_____	_____	
_____	_____	

Use index cards, or single PowerPoint slides.

Limiting the size of a card is an attempt at preventing the class from becoming too complex.

# CRC Cards Technique

- CRC properties

- Informal, non-detailed
- Used for group **brain-storming**
- Ending result is a first cut at **classes** for an object-oriented model
- Not intended to provide a **complete** design

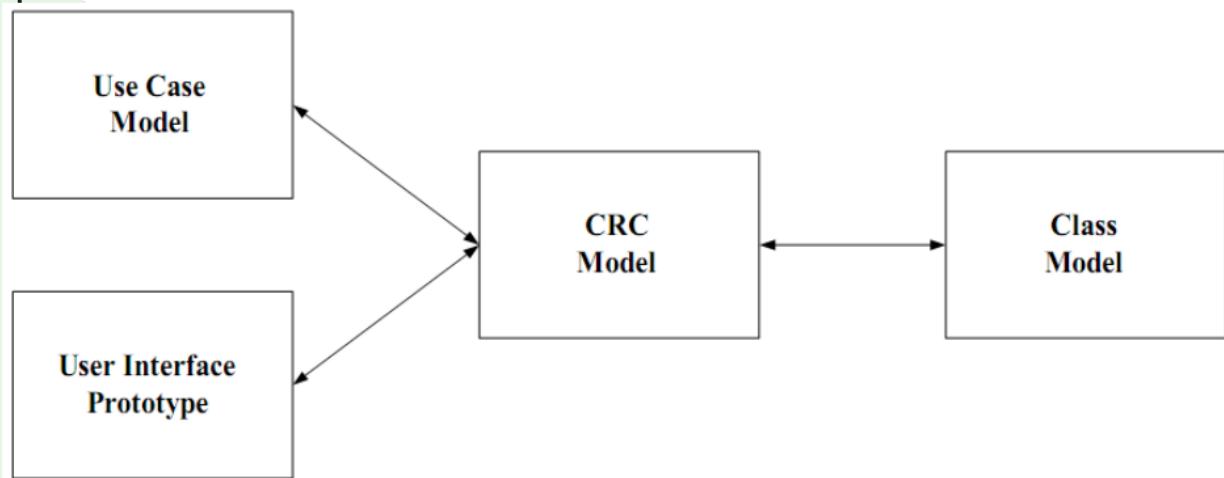
# Process

1. Brainstorming
2. Identify core classes (go to analysis)
  - Critical classes
  - Irrelevant classes
  - Undecided classes
3. Assign responsibilities
4. Assign collaborations

# CRC Input : Use-Case

- A good **starting point** for CRC analysis is a clear statement of all of the use-cases.
- Use-cases drive the introduction of CRC cards.
- Use-cases, or their accompanying scenarios, can be used as a kind of script for the **role-playing method** of checking the CRC cards.
  - The role-playing could be replaced with sequence diagrams.

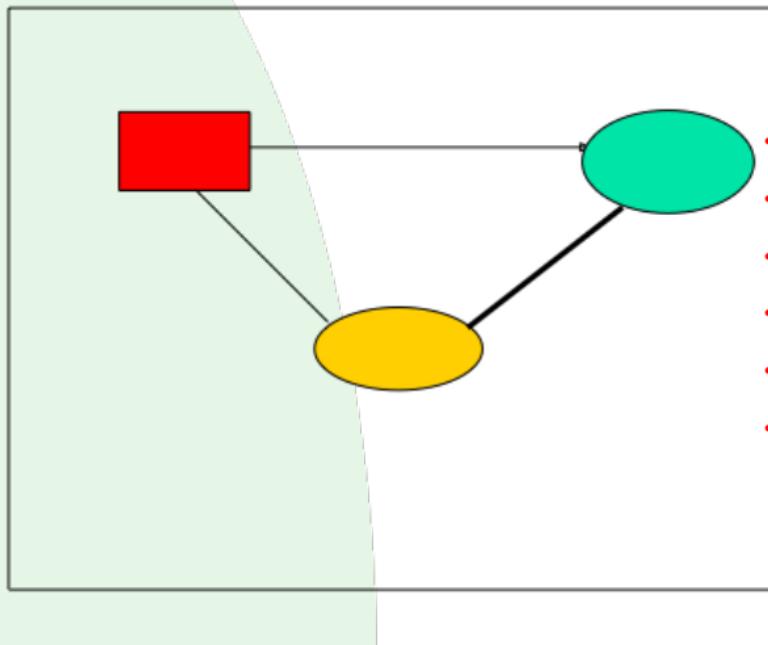
# How CRC modeling fits in



# Sample Application:

## A graph-drawing program

Process screen image



### Typical Application Use-Cases:

- Draw shape**
- Move shape**
- Resize shape**
- Connect shapes**
- Erase shape**
- Erase connector**

# Example of CRC card for a graph-drawing program (1)

Class



Shape

## Example of CRC card for a graph-drawing program (2)

Responsibilities

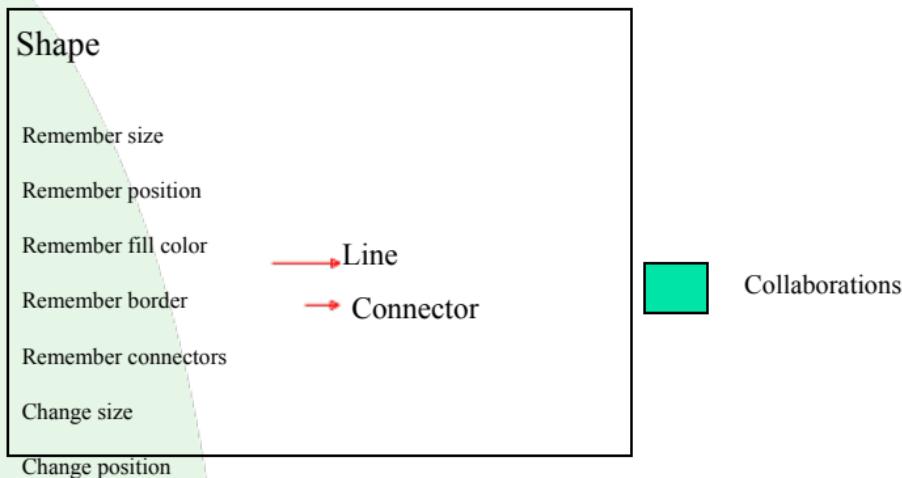
### Shape

- Remember size
- Remember position
- Remember fill color
- Remember border
- Remember connectors
- Change size

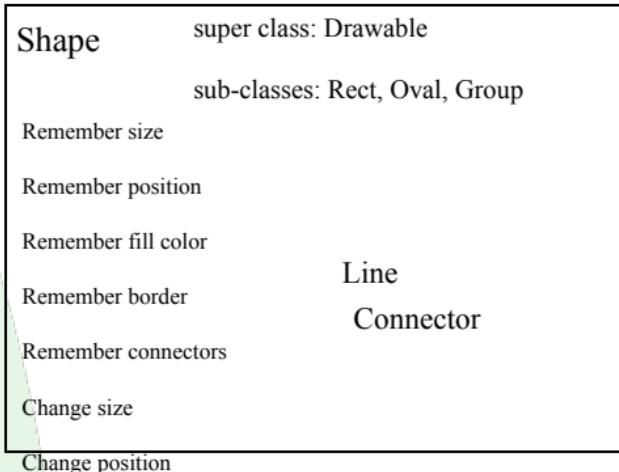
Change position



## Example of CRC card for a graph-drawing program (3)



## Example of CRC card for a graph-drawing program (4)



Super- and  
Sub-classes

## Example of CRC card for a graph-drawing program (5)

Responsibilities

**Drawable**  
super class:  
Draw self on canvas

sub-classes: Shape  
Canvas

Collaborations

Note: The Drawable doesn't necessarily need to *remember* a Canvas,  
since the Canvas could be passed as an argument to the *draw* method.

## Example of CRC card for a graph-drawing program (6)

Responsibilities

**Canvas**  
Remember  
Drawables contained  
in self.

super class:

sub-classes:

Drawable

Collaborations

# Summary

- Guidelines:
  - Who?
  - How?
  - Where?



|下面的内容自学



# CRC guide-who

- Who Writes CRC Cards?
  - No one individual has all needed knowledge
  - Best to write CRC cards as a team, rather than individually
  - Analysis team should consist of five or six experienced individuals :
    - one or two user domain experts
    - one or two systems analysts
    - one OO software designer
    - one team facilitator and leader
    - The team can include clients as well as developers (even though we are partly in the design phase)
  - Smaller teams lack diversity, larger teams make consensus difficult to achieve
  - Facilitator should be skilled at OO and group techniques, not be a "boss" of team
  - Writing of CRC cards focuses team activities

# CRC guide-how

## • Brainstorming

Groups work by

- collecting a diverse set of ideas,
- comparing the ideas, and
- synthesizing unanticipated solutions.

A "brainstorming" session is a good way to collect the ideas quickly and creatively.

• **Brainstorming principles** [Bellin and Simone]:

• **All ideas are potentially good ideas.**

• Don't censure yourself or others -- all ideas are equal

• **Think fast and furiously; ponder later.**

• A fast-paced discussion encourages individual creativity

• **Give every voice a turn.**

• Include everyone in the group. Don't let the loudmouths dominate.

• **A little humor can be a powerful force.**

• Humor can help break down barriers, relieve tension, and build trust needed for a cohesive and effective group

# CRC guide--where to find the candidate classes

## Sources for the Candidate Classes

Before the brainstorming session, assign each team member an investigative task [Bellin and Simone]:

### **Read all requirements documents**

- Examine the formal requirements document (if any)
- Don't overlook the indirect sources -- memos, meeting minutes, etc.
- Circle nouns and noun phrases -- these are potential classes

### **Look carefully at reports**

- Examine the reports generated in the system being replaced (manual or automated)
- Examine the profiles for the reports desired in the new system
- Again circle nouns and noun phrases -- these are potential classes

### **Conduct interviews**

- Talk to experienced users of the current system
- Record interview or take careful, precise notes
- Again identify nouns and noun phrases as candidate classes

### **Examine documentation and files**

- Review the documentation on the current system
- Try to review any unofficial or personal notes users or maintainers of the system have
- Again identify nouns and noun phrases as candidate classes

**A good analyst is a good detective!**

- Once the CRC cards are constructed ...
- Examine?

## Once the CRC cards are constructed ...

- Team can engage in **role-playing** to verify that use-case **scenarios** make sense for chosen CRC.
- Each person can role-play one or more class cards.
- If something doesn't work, change the class accordingly.
- Revision of use-cases might also be indicated.

## Summary: Steps for creating Model using CRC

### 1) Team building

Including client, designer, analyst, and a leader. If there are not so many people, the team can consist of only you and your client.

### 2) objects finding

Find out the nouns and phrases in the requirement, and pay attention that the singular number of plural numbers (usually a set) is the correct one.

write down the concepts you first conceive on the paper, no matter how ridiculous they are.

### 3) Selection

Classify objects into three groups, the core object (the first to be realized), optional (uncertainty right now), and the unnecessary object. Your project scope should be confirmed before hand. Some objects which are not belonging to this projects can be put aside

### 4) Card fetching

Fetch the CRC card, and write core class down on each card, then write optional class and excluded class down on different papers.

### 5) Role playing

A team is better than a person alone to carry out the task. Everyone is in charge of several classes. For every scenario in use case.

The leader assign a class to one person, and he will evaluate that if he can complete it independently, if not, everyone will look at their classes, and stand up if he could do it so as to continue this process.

Anyone can sit down after fulfilling his responsibilities.

In this process, responsibilities of class would be continuously modified and actor's name would be written down.

# 总结

- CRC卡方法
  - 类的识别
  - 方法的识别
- 思考
  - CRC card 得到的类？
  - 是分析类还是设计类？？

# 作业：

- Exercise yourself\*
  - Review the ATM case and CRC principle and method carefully, and understand the essence of using such method to analyze classes and responsibilities.
- Deadline:
- 11.24