北京航空航天大學
BEIHANG UNIVERSITY

信息系统分析与设计

面向对象的系统分析与设计 Object-Oriented Analysis

信息系统系　刘冠男

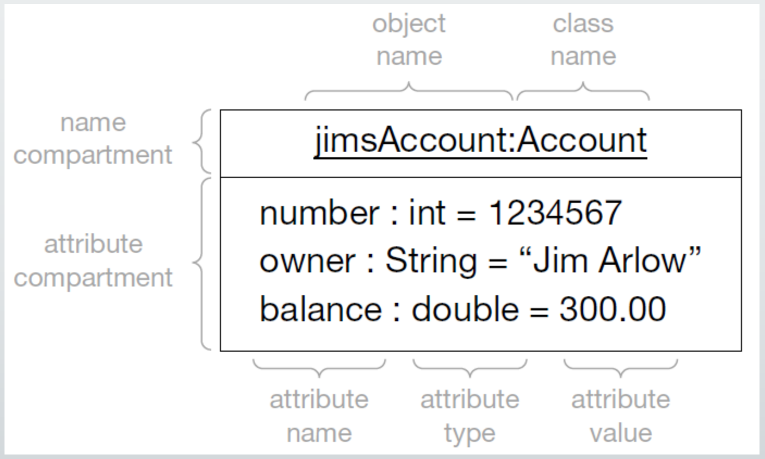
数据智能研究中心
RESEARCH CENTER FOR DATA INTELLIGENCE

面向对象的分析与设计

- Works well in situations where complicated systems are undergoing continuous maintenance, adaptation, and design
- Objects, classes are reusable
- The **Unified Modeling Language (UML)** is an industry standard for modeling object-oriented systems.
- Reusability
  - Recycling of program parts should reduce the costs of development in computer-based systems
- Maintaining systems
  - Making a change in one object has a minimal impact on other objects

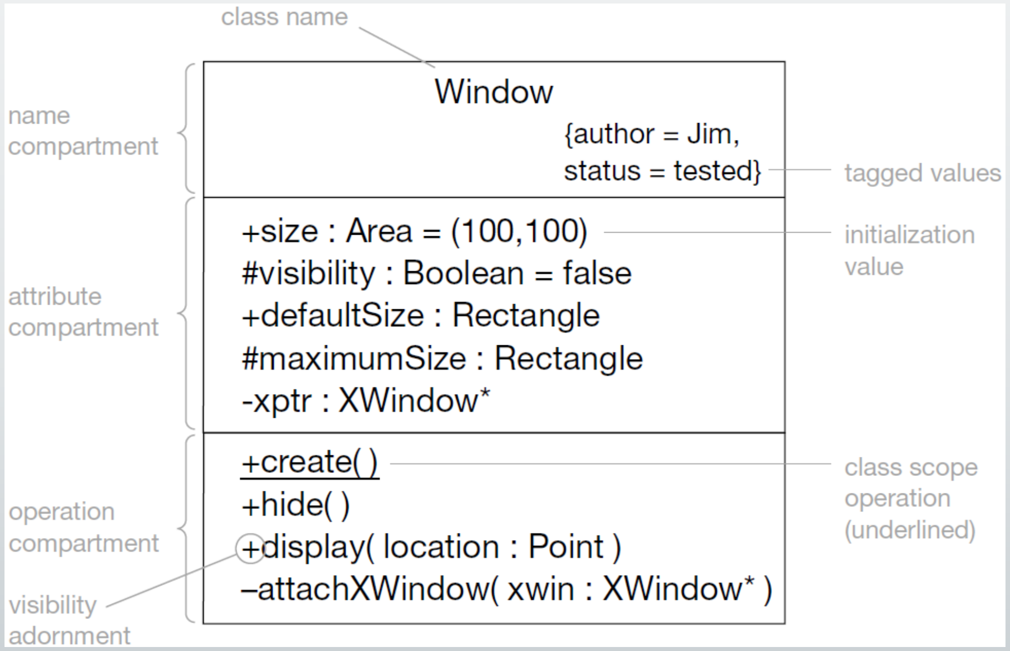
对象 Object

- **Definition**
  - "A discreet entity with a well-defined boundary that encapsulates state and behavior; an instance of a class." --- *UML Reference Manual*
- **An example**

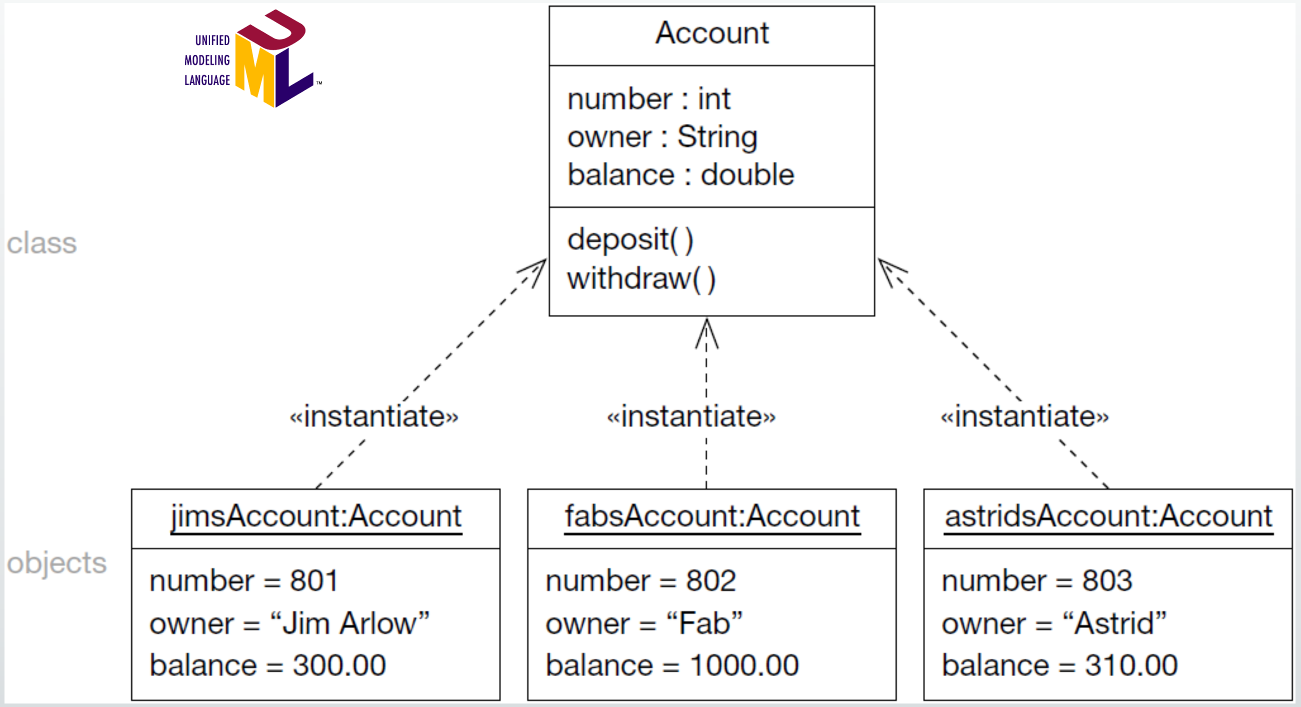


类 Class

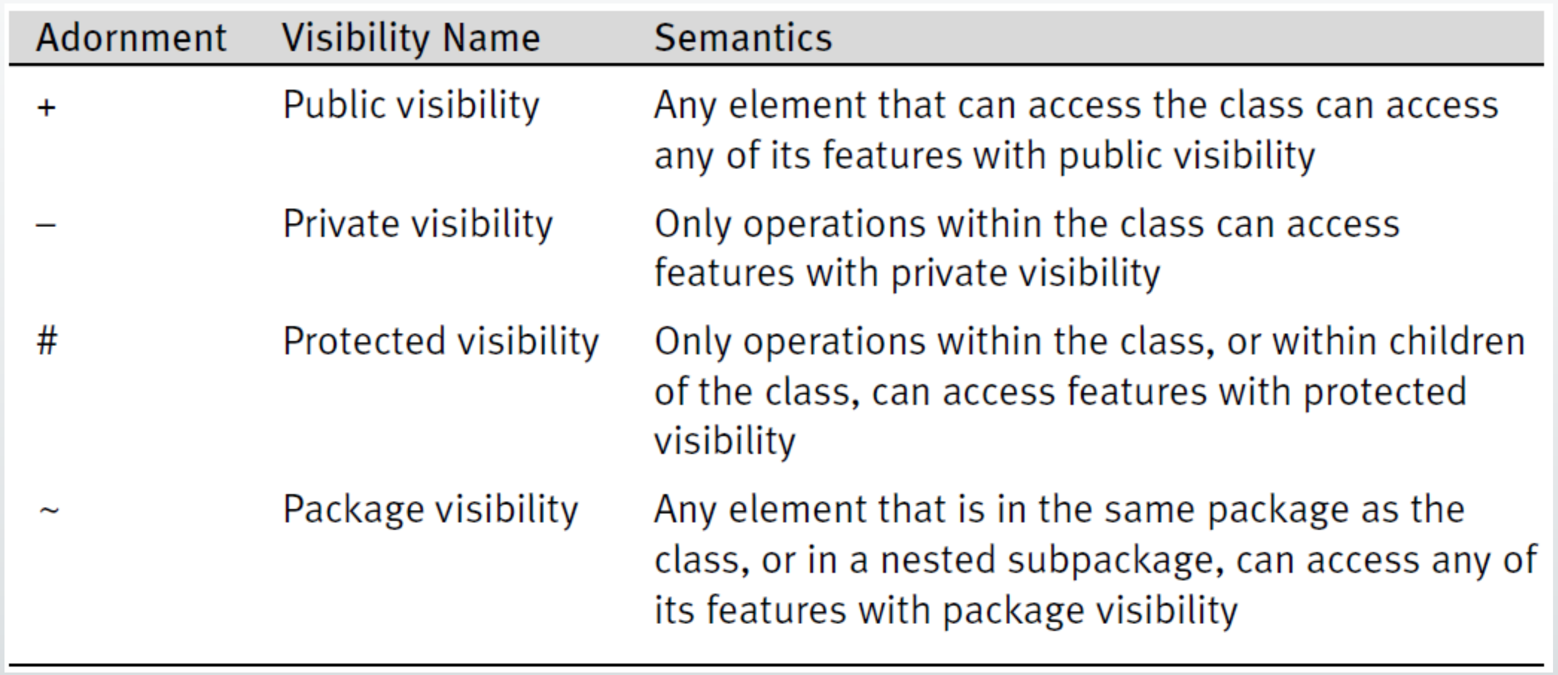
- **Definition**
  - "The descriptor for a set of objects that share the same attributes, operations, methods, relationships, and behavior." --- *UML Reference Manual*
- **An example**



Class and Object



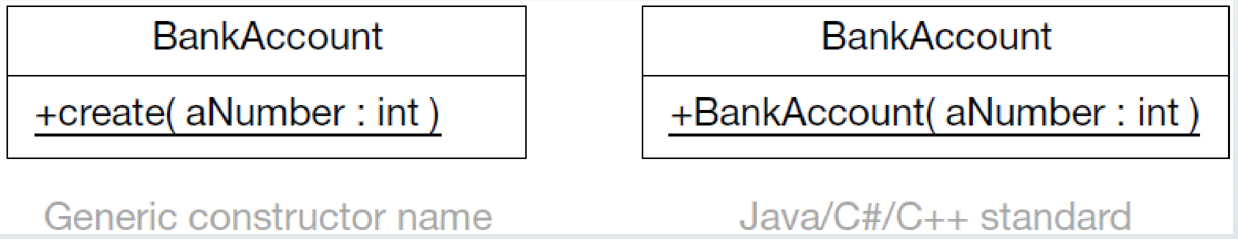
Class: The Visibility



Adornment	Visibility Name	Semantics
+	Public visibility	Any element that can access the class can access any of its features with public visibility
–	Private visibility	Only operations within the class can access features with private visibility
#	Protected visibility	Only operations within the class, or within children of the class, can access features with protected visibility
~	Package visibility	Any element that is in the same package as the class, or in a nested subpackage, can access any of its features with package visibility

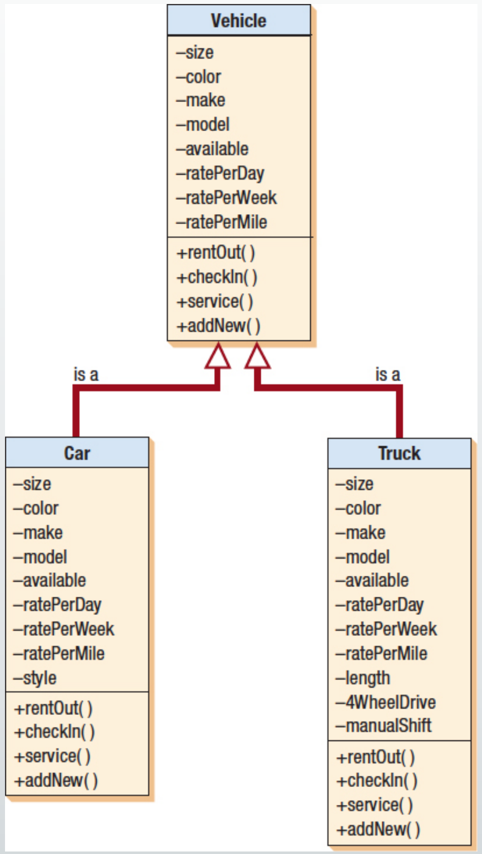
Class: The Constructor

- Constructors are special operations that create new instances of classes – these operations **must** be class scope.



继承

- When a derived class inherits all the attributes and behaviors of the base class
- Reduces programming labor by using common objects easily
- A feature only found in object-oriented systems



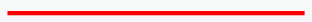
Relationship

- **What is a relationship?**
  - Relationships are semantic (meaningful) connections between modeling elements – they are the UML way of connecting things together.
- **Some already learned relationships:**
  - between actors and use cases (association);
  - between use cases and use cases (generalization, «include», «extend»);
  - between actors and actors (generalization).

You Never Walk Alone --- Anthem of Liverpool

Types of Relationships

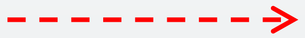
- **Association**



  - connections between two elements

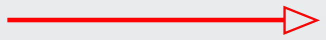
Why claim "elements"?

- **Dependency**



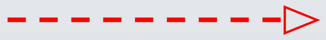
  - relationship between two elements where a change to one element (the supplier) may affect or supply information needed by the other element (the client)

- **Generalization**



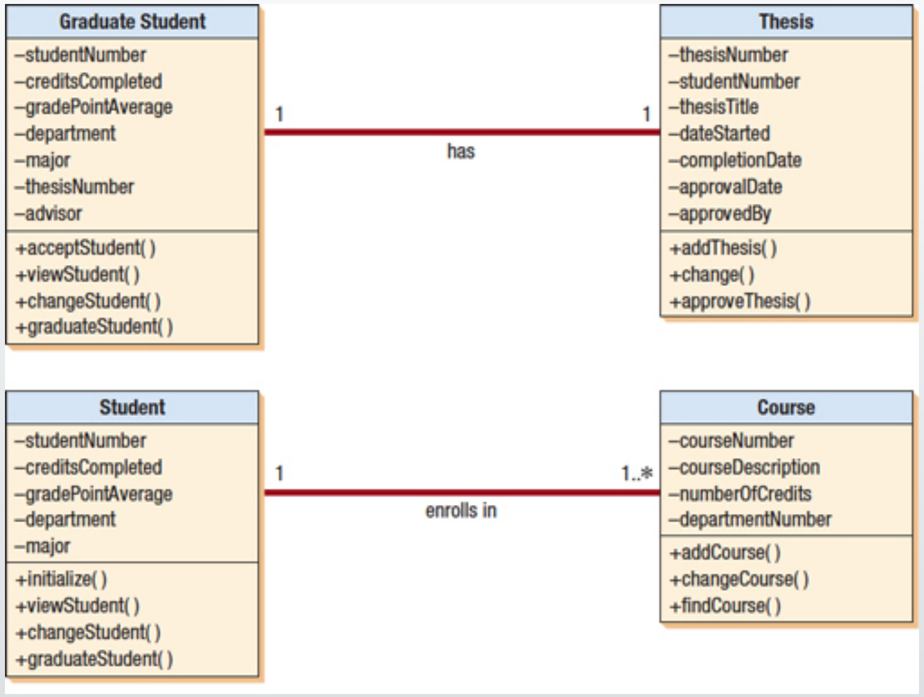
  - relationship between a more general thing and a more specific thing.

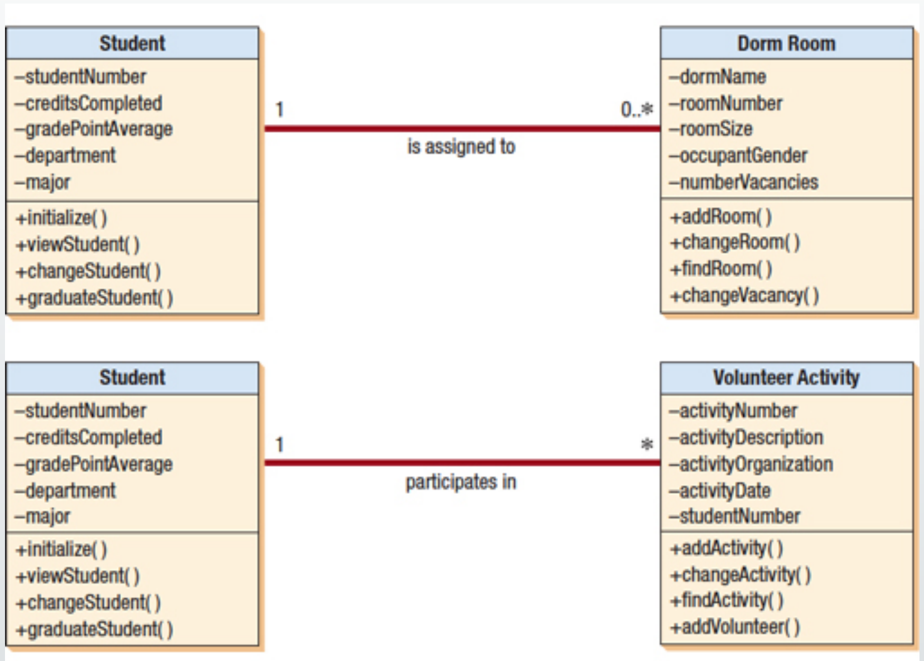
- **Realization**



  - relationship between a specification (e.g. the interface) and its realization

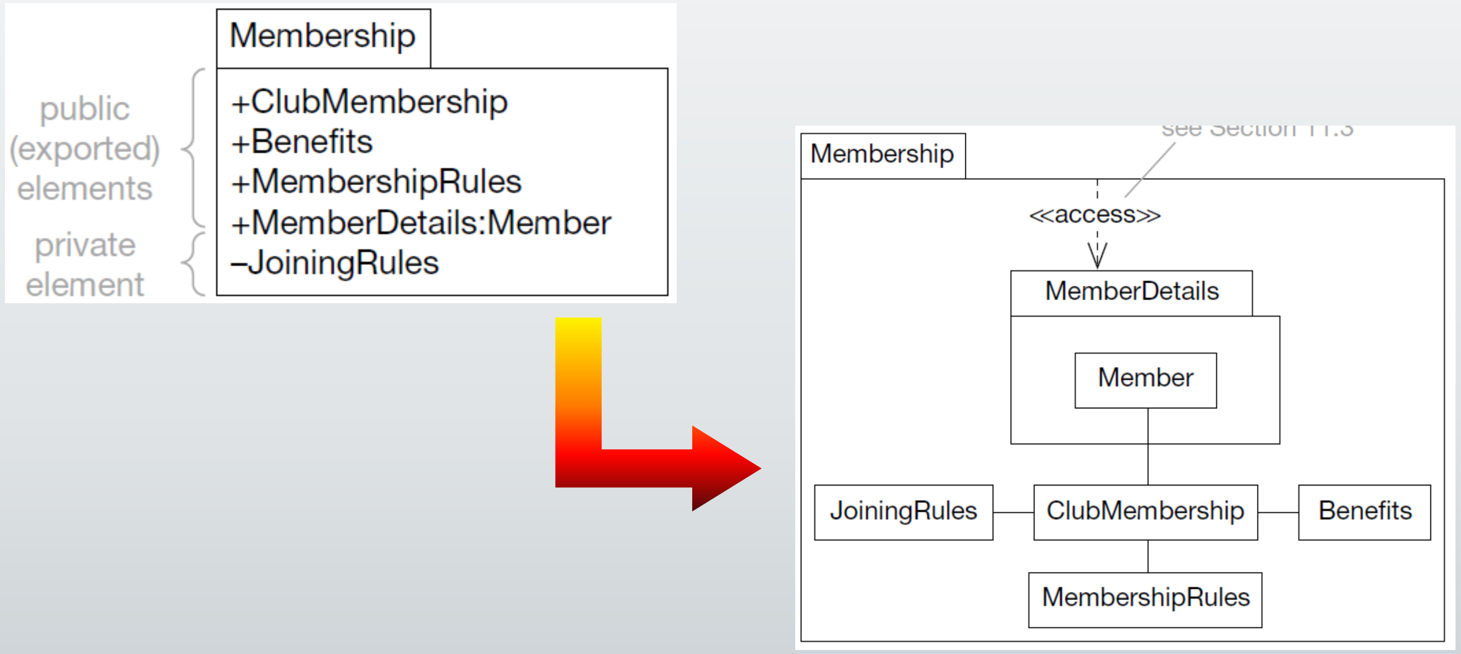
关联关系 Association





包 Package

- 包是一种容器，将信息分类，形成逻辑单元
- The package is the UML mechanism for grouping things.



包图 Package

- **The package is the UML mechanism for grouping things.**
- **一种容器，将信息分类，形成逻辑单元**
  - 在物理上组织和管理文件的包装器，将类文件按一定的规则有序地放置在一起
  - 整合复杂的信息，语义上相关或某方面具有共同点
- **可以容纳任何UML元素**
  - 可以无限分包
- **分包的一些指导性原则**
  - 同一个包内的元素相互联系紧密，不可分割，又具有某些相同的性质
  - 最理想情况：修改任意一个包的元素，其他任何一个包中的内容不受影响
  - 保证包之间的依赖关系不会被传递，B–>A, C–>B
  - 避免双向依赖和循环依赖

分析类

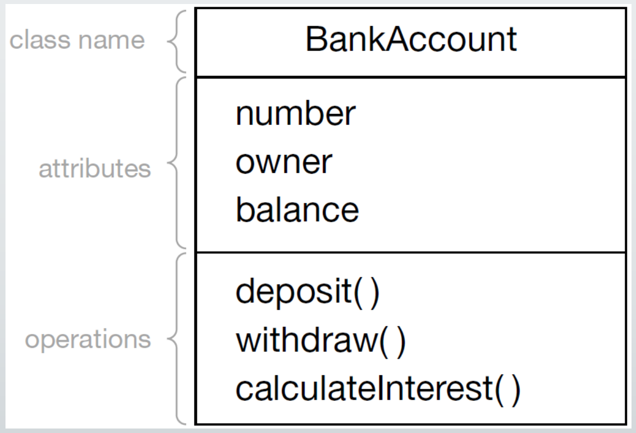
- **分析类用于获取系统中主要的“职责簇”，产生系统设计的主要抽象**
  - Analysis classes model **important aspects** of the problem domain such as “customer” or “product”
- **概念层次，与具体实现技术无关**
- **找到正确地分析类是面向对象分析设计的关键**
- **The idea of an analysis class is that you try to capture the essence of the abstraction, and leave the implementation details until you come to design.**
- **业务需求向系统设计转化过程中的最重要元素**
  - 在高层次抽象出系统实现业务需求的原型
  - 分析类将其逻辑化

分析类的几种类型

- **边界类**
  - 用于对系统外部环境与其内部运作之间的交互进行建模的类
    - 边界对象将系统与其外部环境的变更分隔开，变更不会对系统其它部分造成影响
  - 边界类对系统中依赖于环境的那些部分进行建模
  - 常见的边界类有窗口、通信协议、打印机接口、传感器和终端
- **控制类**
  - 对一个或几个用例所特有的控制行为进行建模
  - 表示系统的动态行为，处理主要的任务和控制流，可以帮助理解系统
- **实体类**
  - 对必须存储的信息和相关行为建模的类
  - 实体对象用于保存和更新一些现象的有关信息，例如：事件、人员

Anatomy of an Analysis Class

- **Only have key attributes and very high-level responsibilities.**
  - Name – this is mandatory.
  - Attributes – only names are mandatory; important ones
  - Operations – only names are mandatory; high level ones
  - Visibility, Stereotype, Tagged Values – not necessary



Finding Classes

- **No simple algorithm for finding the right analysis classes.**
- **You can try:**
  - Noun/Verb analysis;
  - CRC analysis;
  - Looking for other sources of classes.

Noun/Verb Analysis

- **Step 1:** Collect information from
  - Requirement specifications;
  - Use cases;
  - Project glossary;
  - Other resources (architecture, vision documents, etc.)
- **Step 2:** Highlight the following
  - nouns – e.g. flight;
  - noun phrases – e.g. flight number;
  - verbs – e.g. allocate;
  - verb phrases – e.g. verify credit card.

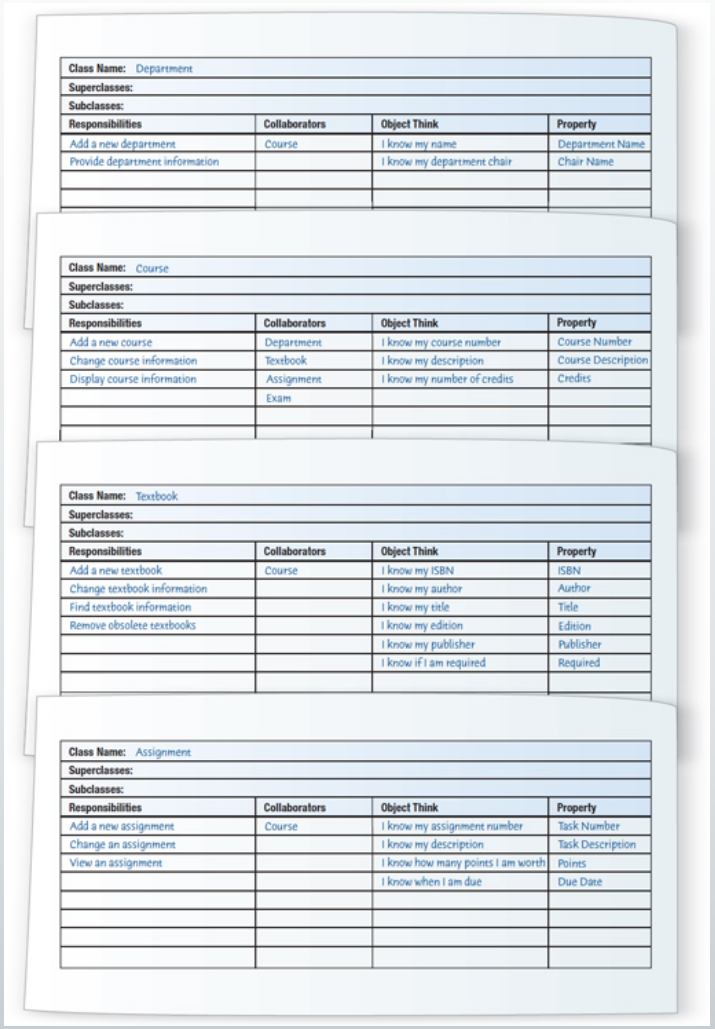
In noun/verb analysis you analyze text. Nouns and noun phrases indicate classes or attributes. Verbs and verb phrases indicate responsibilities or operations.

Noun/Verb Analysis, cont’d

- Step 3: Allocate the attributes and responsibilities to the classes.
- Step 4: Add relationships to the classes.
- Tip: If any terms that you don’t understand
  - Seek immediate clarification from a domain expert and add the term to the Project Glossary.
  - Avoid any synonyms and homonyms.

CRC Analysis

- CRC is a brainstorming technique.
- CRC
  - Class
  - Responsibilities
  - Collaborators



Class Name: Department
Superclasses:
Subclasses:

Responsibilities	Collaborators	Object Think	Property
Add a new department	Course	I know my name	Department Name
Provide department information		I know my department chair	Chair Name

Class Name: Course
Superclasses:
Subclasses:

Responsibilities	Collaborators	Object Think	Property
Add a new course	Department	I know my course number	Course Number
Change course information	Textbook	I know my description	Course Description
Display course information	Assignment	I know my number of credits	Credits
	Exam		

Class Name: Textbook
Superclasses:
Subclasses:

Responsibilities	Collaborators	Object Think	Property
Add a new textbook	Course	I know my ISBN	ISBN
Change textbook information		I know my author	Author
Find textbook information		I know my title	Title
Remove obsolete textbooks		I know my edition	Edition
		I know my publisher	Publisher
		I know if I am required	Required

Class Name: Assignment
Superclasses:
Subclasses:

Responsibilities	Collaborators	Object Think	Property
Add a new assignment	Course	I know my assignment number	Task Number
Change an assignment		I know my description	Task Description
View an assignment		I know how many points I am worth	Points
		I know when I am due	Due Date

CRC Analysis, cont’d

- **Step 1: Brainstorm – gather the information**
  - Explain that this is a true brainstorm.
  - Name the “things”.
  - State responsibilities.
  - Find collaborators and relationships.
- **Step 2: Analyze information**
  - Decide which sticky notes should become classes and which should become attributes.

Other Sources of Classes

- **Still Other Sources:**
  - Physical objects such as aircraft, people, and hotels
  - Paperwork
  - Interfaces such as screens, keyboards

What are Good Analysis Classes

- **Some key points:**
  - its name reflects its intent;
  - it is a crisp abstraction that models one specific element of the problem domain;
  - it maps on to a clearly identifiable feature of the problem domain;
  - it has a small, well-defined set of responsibilities;
  - it has high cohesion;
  - it has low coupling.

刘冠男
liugn@buaa.edu.cn

数据智能研究中心
RESEARCH CENTER FOR DATA INTELLIGENCE