Fundamental Concepts of Object Oriented Methods

Yang YI, Ph.D
Computer Science Department, SYSU
issyy@mail.sysu.edu.cn

Chapter 2 Best Practices of Software Engineering

agenda

Methodology of objected oriented based system analysis & design

Discussion

- **What is your perception of object technology?**
- **What do you perceive as object technology's strengths? Its weaknesses?**
- **Why are you making the shift to object technology?**

What Is Object Technology?

- A set of principles (abstraction, encapsulation, polymorphism) guiding software construction, together with languages, databases, and other tools that support those principles.
  - (*Object Technology – A Manager's Guide*, Taylor, 1997.)

The Strengths of Object Technology

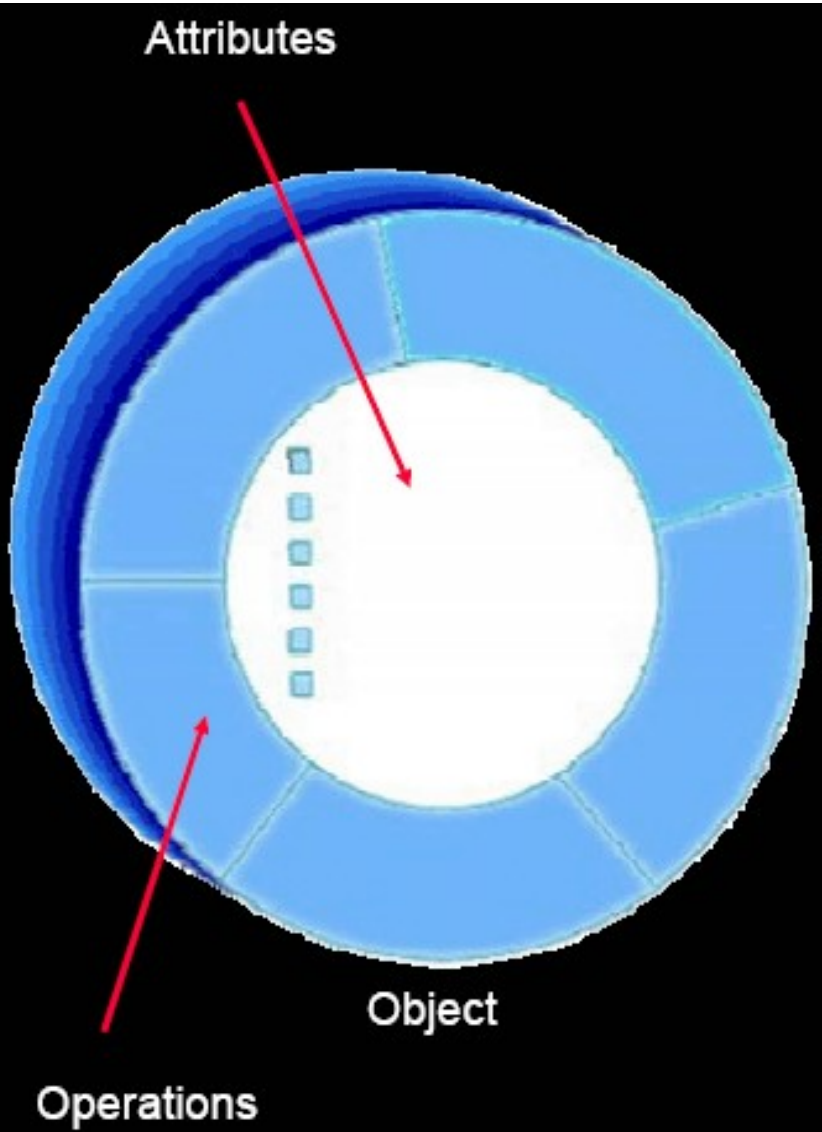
- Reflects a single paradigm
- Facilitates architectural and code reuse
- Reflects real world models more closely
- Encourages stability
- Is adaptive to change

Differences Between OO and Structured Design Object-orientation (OO)

- Melds the data and data flow process together early in the lifecycle
- Has a high level of encapsulation
- Promotes reuse of code differently
- Permits more software extensibility

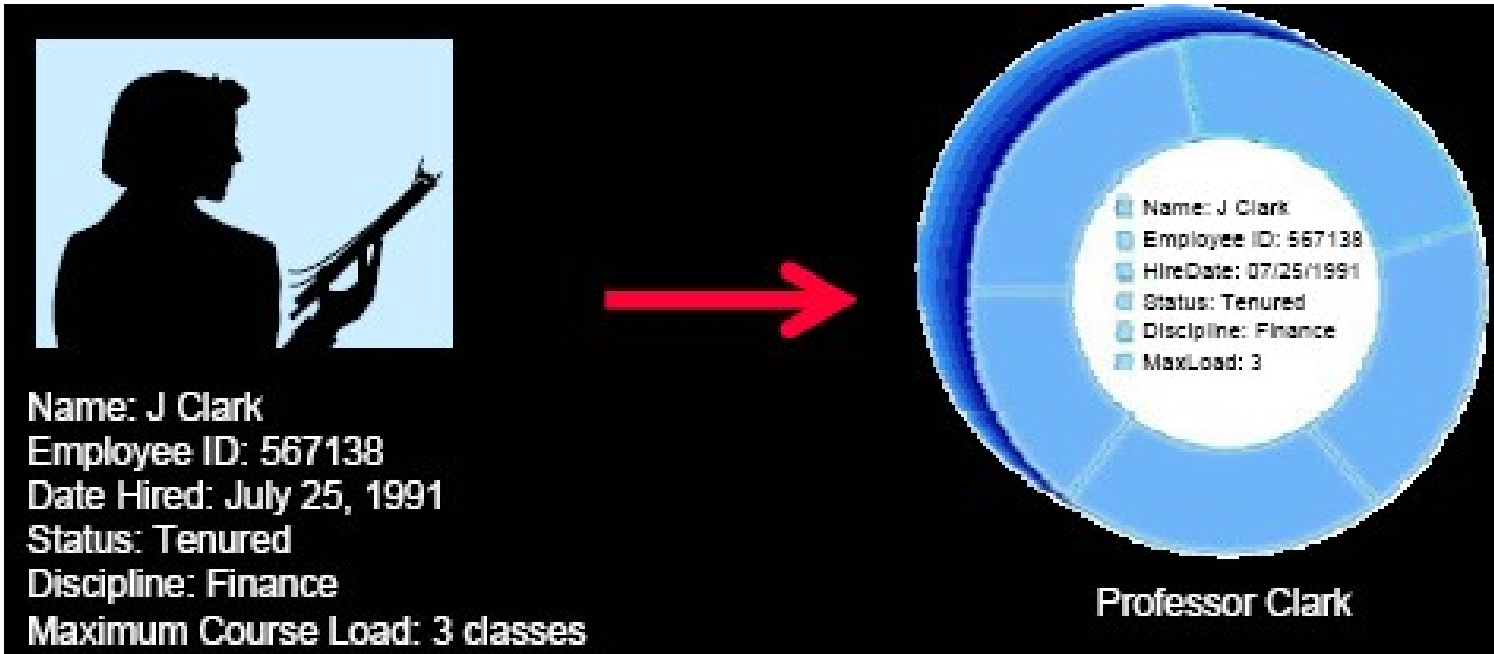
A Formal Definition of Object

- An object is an entity with a well-defined boundary and *identity* that encapsulates *state* and *behavior.*
  - State is represented by attributes and relationships.
  - Behavior is represented by operations, methods, and state machines.



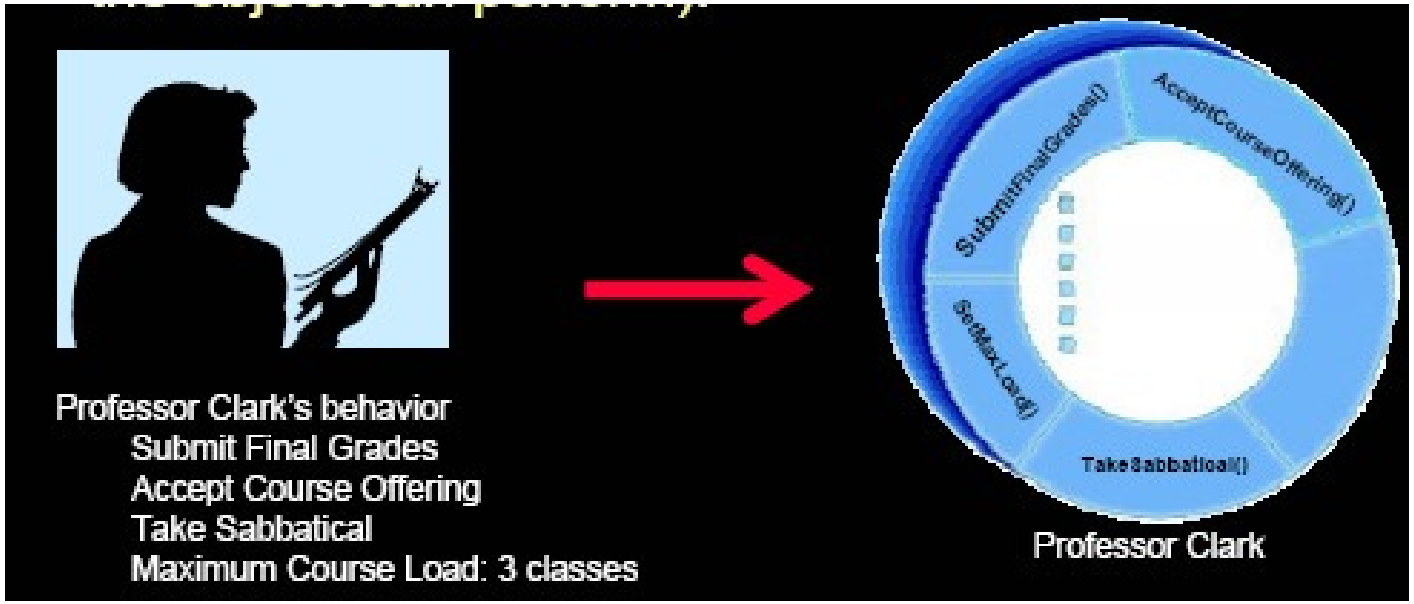
An object has state

- State is a condition or situation during the life of an object, which satisfies some condition, performs some activity, or waits for some event.
- The state of an object normally changes over time.



An object has behavior

- Behavior determines how an object acts and reacts.
- The visible behavior of an object is modeled by a set of messages it can respond to (operations that the object can perform).



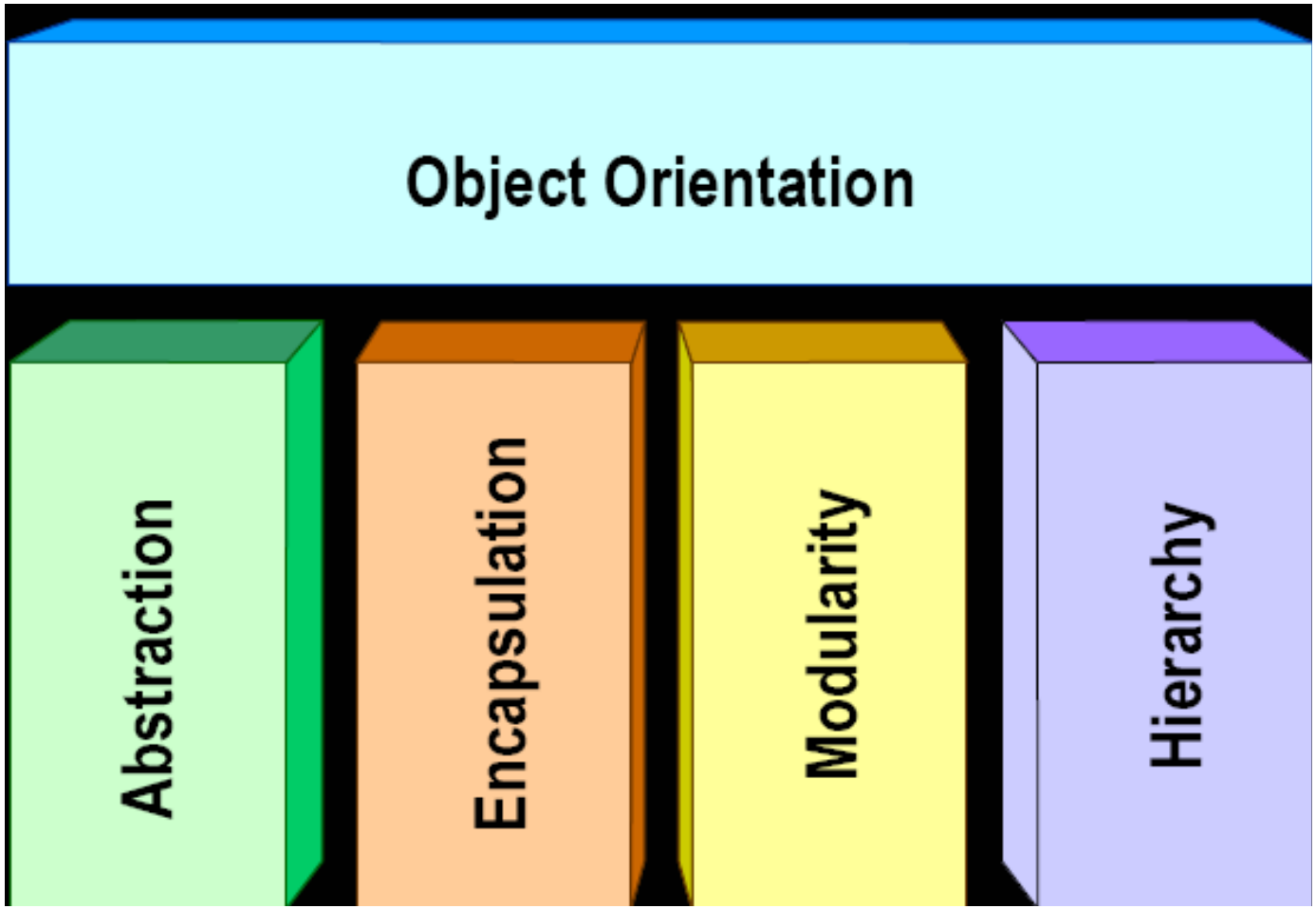
对象可唯一识别

- 即使对象有相同的特性，还是能识别每个不同的对象

Professor “J Clark” teaches Biology

Professor “J Clark” teaches Biology

Basic Principles of Object Orientation

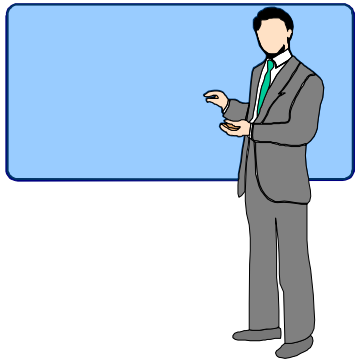


What Is Abstraction?

- The essential characteristics of an entity that distinguishes it from all other kinds of entities.
- Defines a boundary relative to the perspective of the viewer.
- Is not a concrete manifestation, denotes the ideal essence of something.

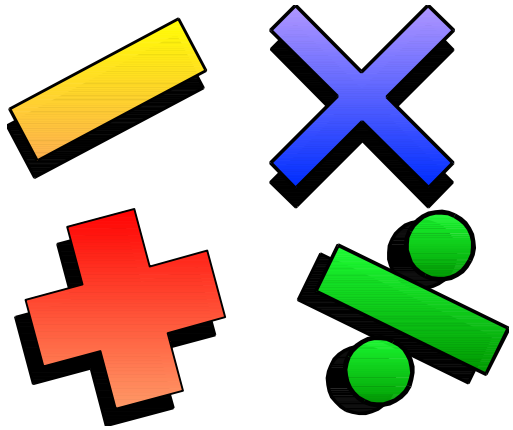
抽象示例

学生



教授

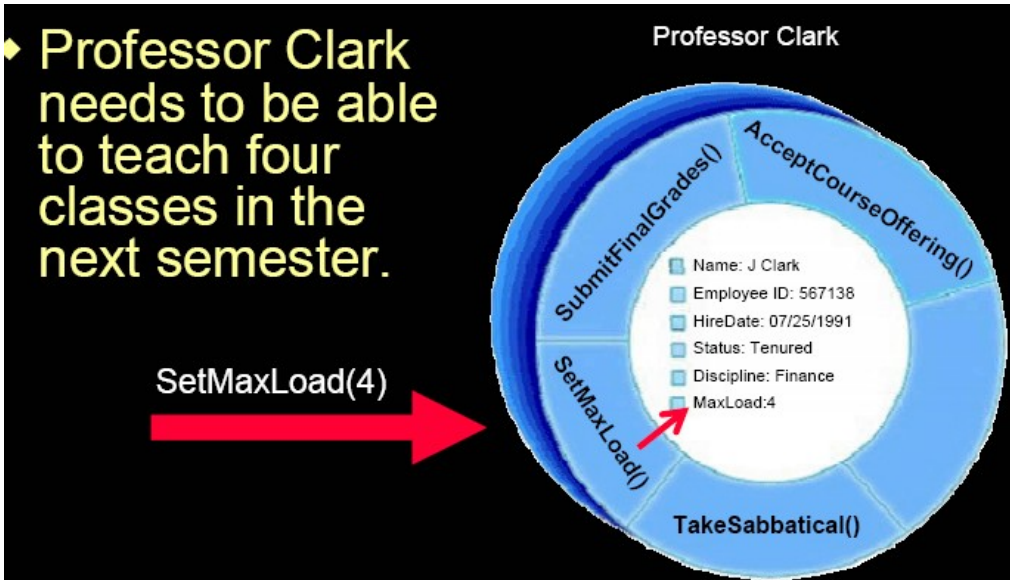
课程提供 (9:00 AM，周一，周三，周五)



课程（例如，代数学）

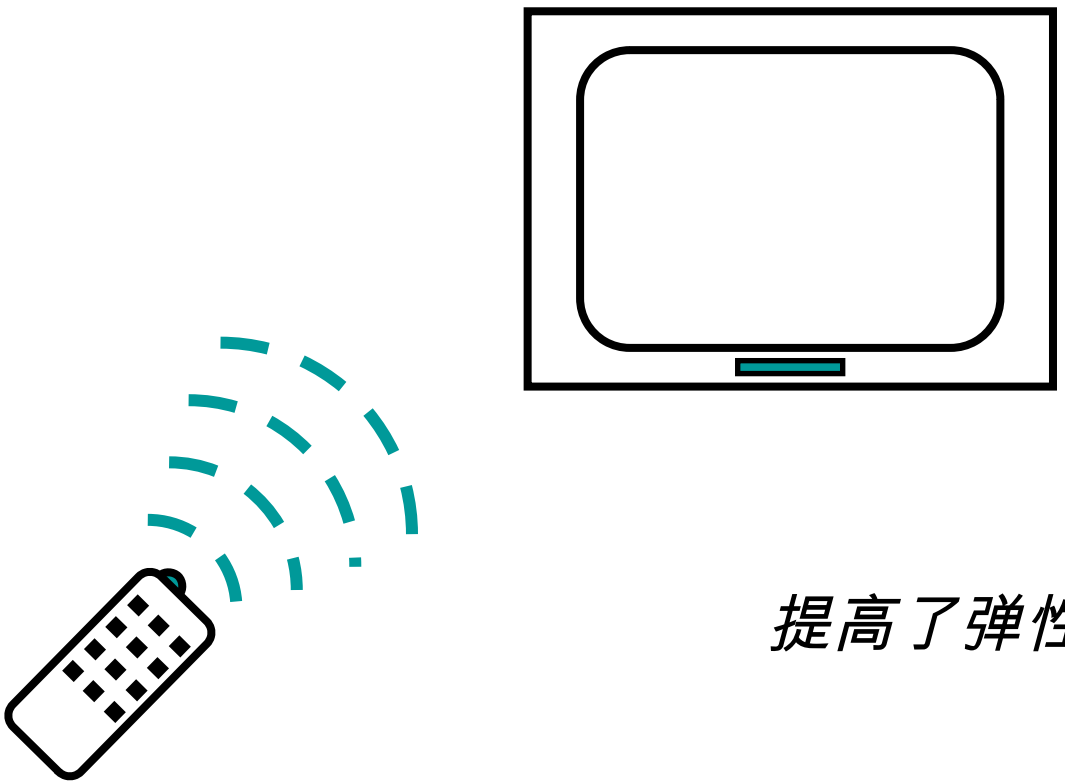
What Is Encapsulation ?

- Hides implementation from clients.
- Clients depend on interface.
- Improves Resiliency



封装（ Encapsulation ）

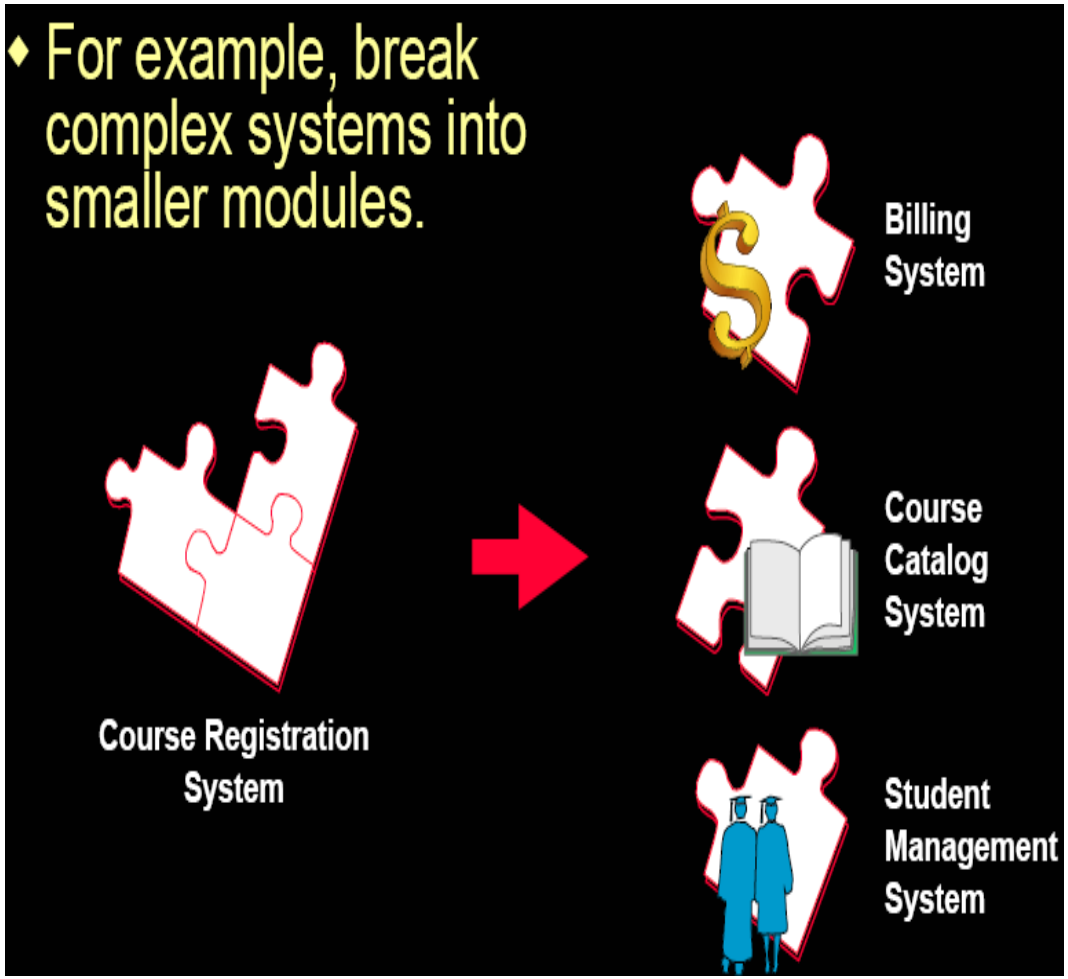
- 对客户隐藏实现
  - 客户依赖于接口



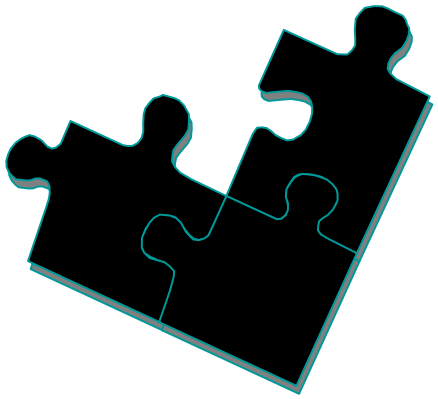
提高了弹性

What Is Modularity?

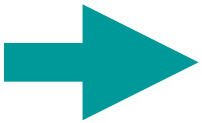
- Breaks up something complex into manageable pieces.
- Helps people understand complex systems.



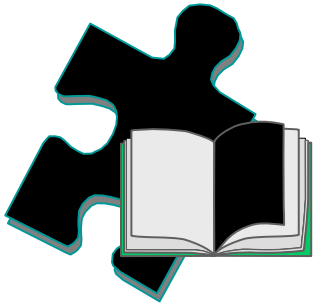
模块化示例



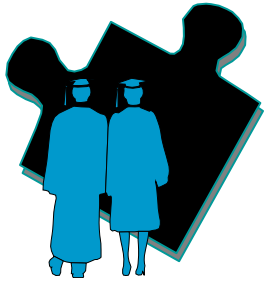
课程注册系统



付费系统

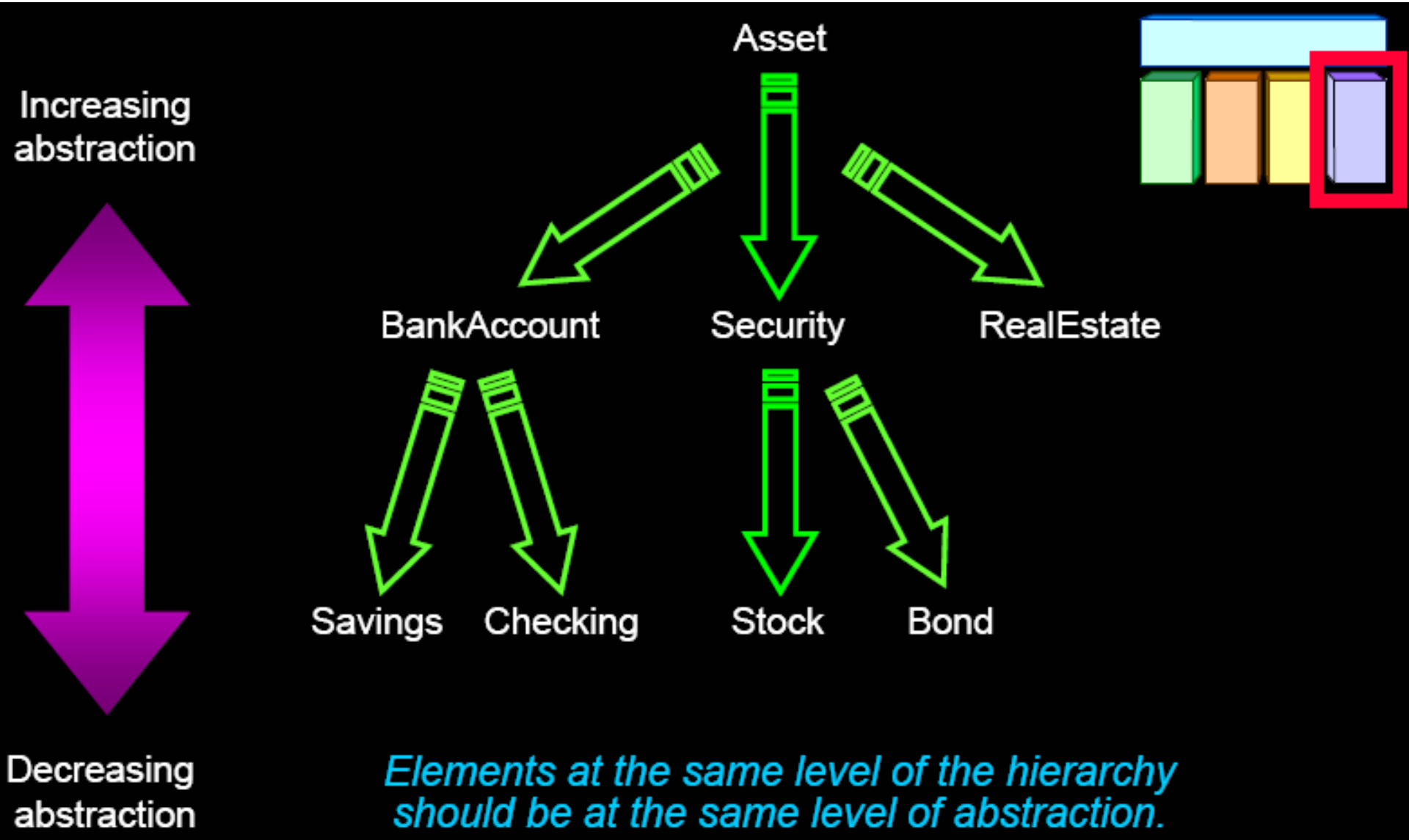


课程目录系统



学生管理系统

What Is Hierarchy ?



Review(1)

- What is an object?
- What is an attribute? An operation?
- What are the four principles of object orientation? Describe each.
  - What Is Abstraction?
  - What Is Encapsulation ?
  - What Is Modularity?
  - What Is Hierarchy ?

Representing Objects in the UML

- An object is represented as a rectangle with an underlined name.

Professor J Clark

J Clark :
Professor

Named Object

: Professor

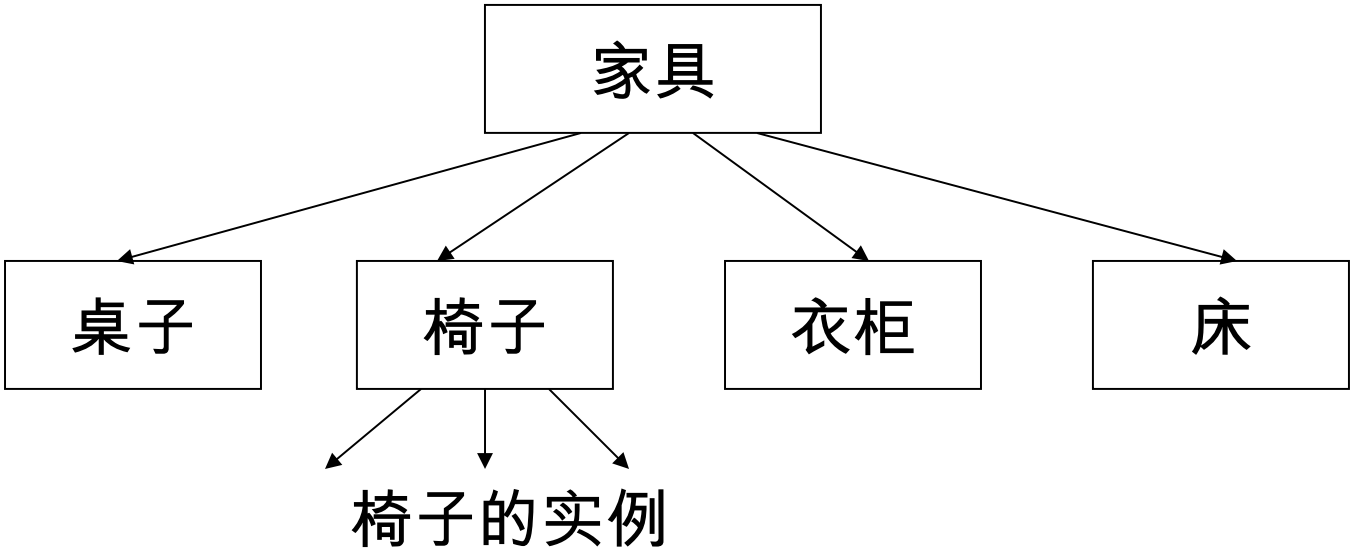
Anonymous Object

What Is Generalization?

- A relationship among classes where one class shares the structure and/or behavior of one or more classes.
- Defines a hierarchy of abstractions in which a subclass inherits from one or more superclasses.
  - Single inheritance.
  - Multiple inheritance.
- Is an "is a kind of" relationship.

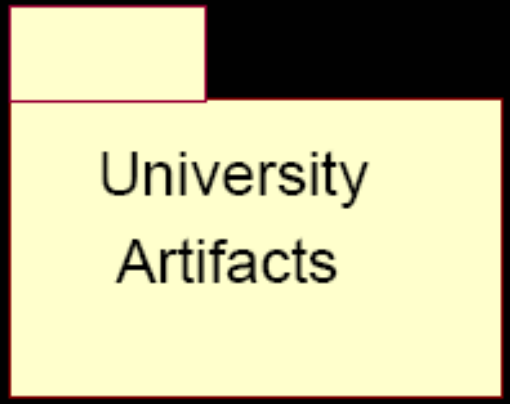
继承（ Inheritance ）

- 使用已存在的定义做为基础建立新定义的技术



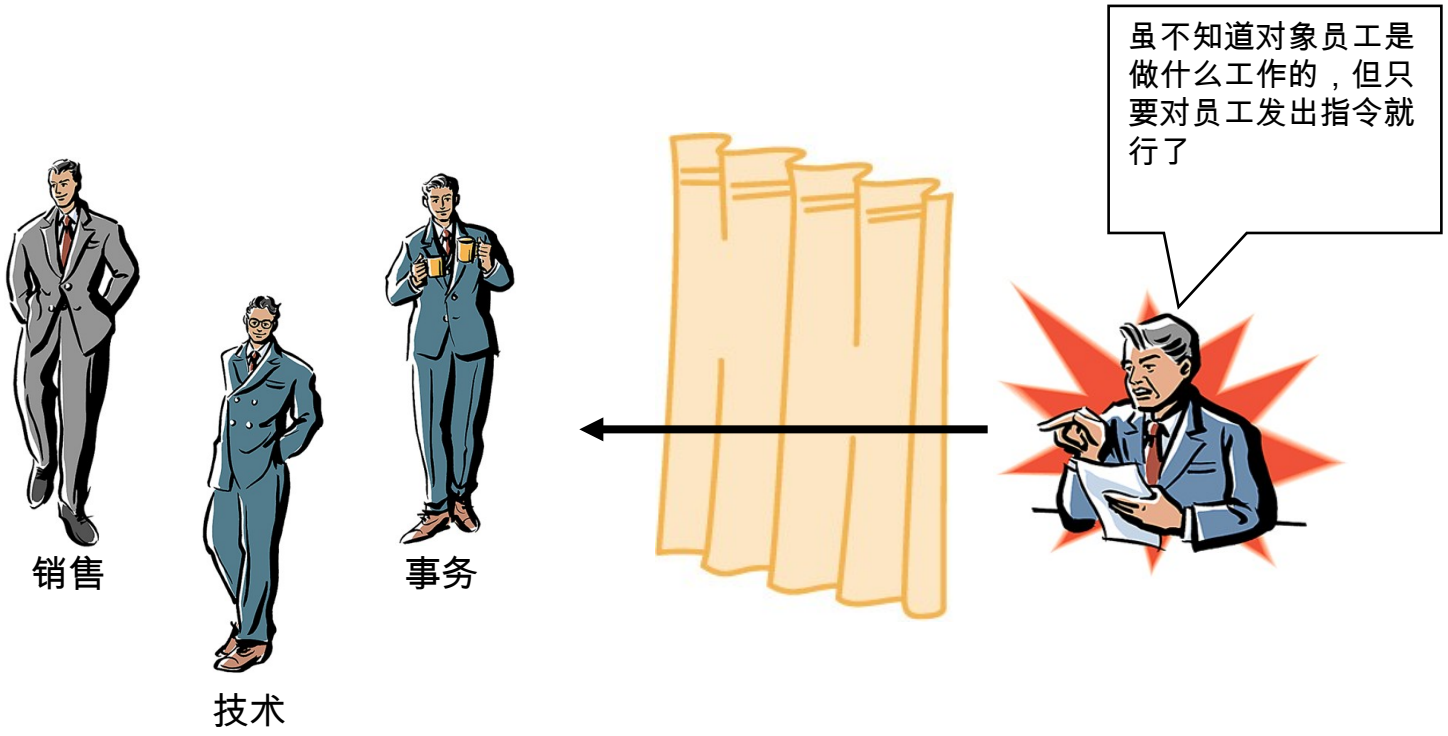
What Is a Package?

- A general purpose mechanism for organizing elements into groups.
- A model element that can contain other model elements.
- A package can be used:
  - To organize the model under development.
  - As a unit of configuration management.



多态（ Polymorphism ）

- 对于相同的消息，让各个对象产生不同的行为



多态性示例

计算各员工工资的程序

不用多态性时

```
if ( 员工对象 =“ 销售” ) {
    payment = 计算销售人员工资 ();
} else if ( 员工对象 = “ 技术” ) {
    payment = 计算技术人员工资 ();
} else if ( 员工对象 = “ 事务” ) {
    payment = 计算事务人员工资 ();
}
```

工作分工一增加，程序就必须修改

使用多态性时

```
payment = 员工对象 . 工资计算 ();
```

即使工作分工增加，程序也不需要修改

Discussion

- What is an activity diagram and why would you use one?
- What is the difference between an activity and an action?
- What is a partition?
- What are the different types of interaction diagrams?
- What is a combined fragment?
- What are some examples of interaction operators?

Discussion

- What is generalization? What is inheritance?
- What is package? Why use packages?
- Define polymorphism. Provide an example of polymorphism.

Assignments

- Group, 4-5 persons each, one team leader (PM)
- Prepare a software system,
  - whose scale is suitable for your future work in our course;
  - from next week, you are gonna to be asked to present or demonstrate your jobs in following course time
- Problem statement