## **₩**

# Object oriented Analysis & Design

面向对象分析与设计

Lecture\_01 面向对象概述

主讲: 姜宁康 博士

日期: 2018/11/7



■ 1、课程开始之前...



# 国家对软件人才的呼唤



"网络空间的竞争,归根结底是人才竞争。建设网络强国,没有一支 优秀的人才队伍,没有人才创造力迸发、活力涌流,是难以成功的。"

——习近平总书记

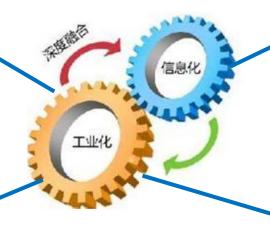


工业互联网



大数据

## 卓越人才



信息化和工业化深度融合



云计算



网络安全





# 软件, 无处不在!







# 软件危机

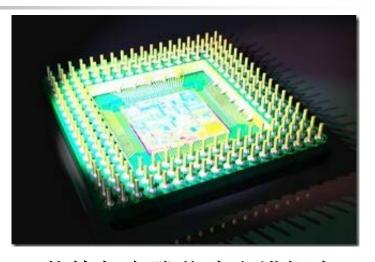




1996年阿丽亚娜火箭爆炸损 失超过3.7亿美元



1985-1987年间,Therac-25 软件漏洞造成至少5人死亡



英特尔奔腾芯片出错损失 4.75亿美元及公司信誉



2007年北京机场信息系统瘫痪,至少造成84个航班延误



# 卓越软件人才严重缺乏

- ◆课程的培养目标
  - ◆ 软件工程师
  - ◆ 高端软件人才





软件工程师

软件蓝领



# 卓越软件人才严重缺乏

- ◆课程的培养目标
  - ◆ 软件工程师
  - ◆卓越软件人才
    - 1) 系统分析和设计理论基础,掌握大系统需求分析、建模与仿真技术
    - 2)应用系统设计能力,能够从事新兴产业软件的研发
    - 3) 国际前沿技术学习能力和交流能力
    - 4)项目管理能力及自主业务提升能力

注: 这门课只是其中的一个环节!



# 卓越软件人才严重缺乏

## ◆课程的培养目标

我们作为IT产业的从业者,尤其是年青一代的IT人,要把个人理想与国家需要紧密联系联系起来,心中树立"振兴中国IT产业"的抱负,夯实自己的专业基础。

- ◆本课程,帮助同学们掌握面向对象的思维方式,并在 系统分析、软件设计中加以应用
- ◆希望课程结束之时,同学们能够有所收获,为今后的职业生涯打下良好的基础!

## ■ 2、课程定义

- 一种思维方式
- 一种思考如何用面向对象方法去思考的课程

# 关于思维



- 很难的问题:人是如何思考的?
- 更难的问题: 思考"人是如何思考的?"

《<u>庄子·秋水·子非鱼</u>》这个著名的故事,可以说明这一点。

庄子和惠子一道在濠水的桥上游玩。庄子说: "白儵 (shu)鱼游得多么悠闲自在,这就是鱼儿的快乐。"惠子说: "你不是鱼,怎么知道鱼的快乐?"庄子说: "你不是我,怎么知道我不知道鱼儿的快乐?"惠子说: "我不是你,固然不知道你;你也不是鱼,你不知道鱼的快乐,也是完全可以肯定的。"庄子说: "还是让我们顺着先前的话来说。你刚才所说的'你怎么知道鱼的快乐'的话,就是已经知道了我知道鱼儿的快乐而问我,而我则是在濠水的桥上知道鱼儿快乐的。"

## 关于思维



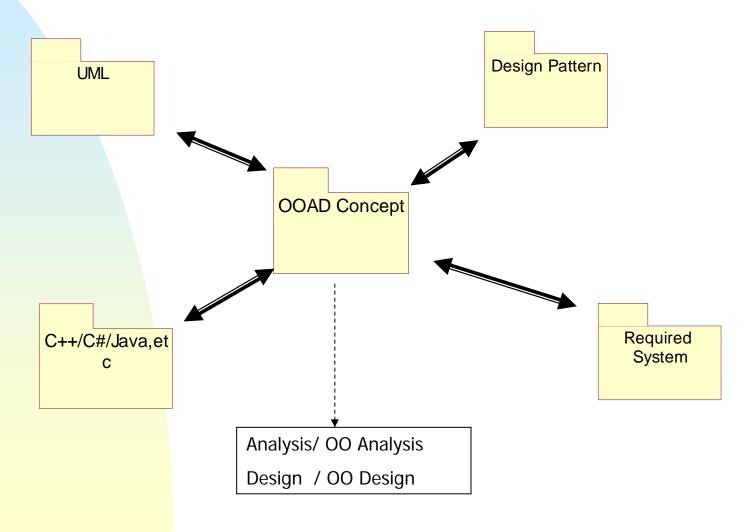
- 思维方式有多少种呢?
  - 无数!
  - 抽象思维、线性思维
  - 发散思维、横向思维
  - 辩证思维、究根思维
  - 明哲保身、一切向钱看....
- 面向对象 Object-oriented
  - 是一种思维方式
- 本课程
  - 学习面向对象的思维方式
  - 利用面向对象的思维方式去思考实际问题
    - 分析、设计、解决方案





# 与课程相关的其他课程

一些前导课程有益于本课程的学习





# 与课程相关的其他课程

- 这些前导课程有益于本课程的学习
  - Knowledge and experience in an object-oriented programming language 使用面向对象语言的编程经验
    - such as C++, Java, C#, or Python
    - 在本课程的学习过程中,可以用代码实现自己的设计思想,体会设计的妙用
  - Knowledge of fundamental OO concepts 了解面向对象的一些基本概念、特性
    - such as class, instance, interface, polymorphism, encapsulation, and inheritance 例如,类、实例、多态、封装、 继承等
    - UML 统一建模语言,是一种建模工具,表达设计思想的工具





# 与课程相关的其他课程

- "拥有一把锤子未必能成为建筑师"
  - 对创建对象系统来说,了解面向对象语言、UML工具 是必要的
  - 但是,了解"对象思想"才是最重要的!
- 窘境:
  - 如果一个程序员,不懂面向对象,那么真的可能被别人认为是"程序猿"了
  - 因为这样的程序员既像"猿"一样原始,又像"猿" 一样稀少

# 在课程的教/学过程中...

- 尝试回答如下一些问题:
  - 为什么要设计、为什么恰当的软件设计是一种过程、 它很重要吗?
  - 什么是面向对象设计、有没有超越面向对象设计的设计方法?
  - 什么是好的设计、如何区分好的设计与坏的设计、好的设计有哪些重要的特点?
  - 如何去设计、针对通常的设计问题,有没有现成的处方?



# 关于学习这一件事...

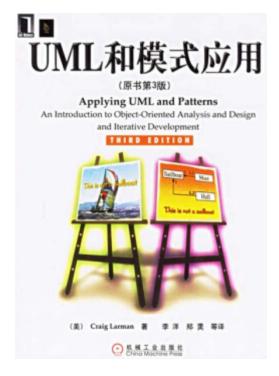
- 我们每个人都有自己认可的准则:
  - 为什么学?怎么学?...
- LOU HOItZ (卢·霍兹, American football coach ) 说到
  - Ability is what you're capable of doing. Motivation determines what you do. Attitude determines how well you do it. 你的天赋决定你能做什么事;你的动 机决定你化多少努力去做;你的态度决定了你能把事 情做得多好。
- 《黑客帝国》中的一句话
  - I'm trying to free your mind, Nero. But I can only show you the door. You're the one that has to walk through it. 师傅引进门,修行在个人。多多练习!



## Text& Reference Book



- Text Book 教材
  - UML和模式应用 Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development,
    - Third Edition, By Craig Larman





# Text& Reference Book

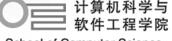


- Readings 其它阅读
  - 关于OOAD的书
    - 面向对象葵花宝典--- 思想、技巧与实践, 李云华编著, 2015 年,电子工业出版
    - 《深入浅出面向对象的分析与设计》(Head First Object-Oriented Design and Analysis)
    - The Object-Oriented Thought Process, Second Edition ,By Matt Weisfeld
    - UML2 and the Unified Process: practical Object-oriented Analysis and design, Second edition, Jim Arlow, Ila Neustadt
  - 阅读关于思维的书
    - 你的灯亮着吗? 唐纳德.高斯、杰拉尔德.温伯格,人民邮电出版
    - ■程序员思维修炼, Andy Hunt, 人民邮电出版
    - 黑客与画家, Paul Graham,



■ 3、"面向对象"思想的起源

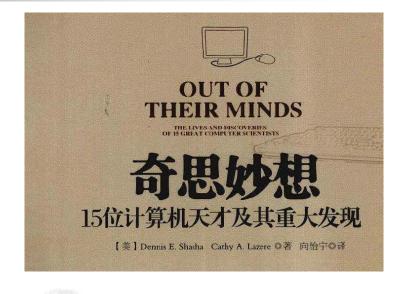




School of Computer Science and Software Engineering

# 计算机界的一位天才级人物 Alan.Kay





3

艾伦・C.凯 清晰的浪漫主义梦想

所有对事物的认识都始自于我们不愿盲目地接受这个世界

——艾伦. C.凯

# 艾伦.凯的成就



- Smalltalk语言之父
- 创造性地提出面向对象编程思想
- 提出Dynabook的概念
- 2003年获得图灵奖



# 生平: 家庭的熏陶



## 1940年,艾伦·凯出生在美国的马萨诸塞州

我的父亲是科学家,母亲是艺术家,所以在我童年的家庭氛围中充满了各式各样的想法,以及各种各样表达它们的方式。我至今也从未把"艺术"和"科学"分开过。

——芝伦•凯

凯3岁的时侯学会了阅读,小小的他经常沉醉在书的海洋里。同时他从母亲那儿接受了音乐的熏陶,不仅是学校合唱团的男中音独唱,还会演奏吉他。那时的他觉得自己未来肯定会成为一位音乐家。



# 生平: 善于观察与思考



后来,他进入美国空军服役,遇到了一个改变命运的契机:通过了计算机编程能力测试,成为了一名程序员,为空军解决各个航空训练设备之间**数据和过程**(procedure)的传输问题。

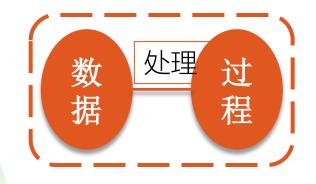
这时他发现有个程序员想出了一个聪明的办法,就是把 **数据和处理数据的过程**捆绑到一起发送。





# 生平: 善于观察与思考

这时他发现有个程序员想出了一个聪明的办法,就是把<u>数据</u>和**处理数据的过程**捆绑到一起发送。



**亮点**:通过这种方式,新设备里的程序就可以直接使用过程,而不用去了解数据文件的格式。

这种抛开数据、直接操作过程的想法给凯带来了极大的触动,为他日后形成"**对象**"的概念埋下了伏笔。



# 生平: 善于联想、类比



服役结束后,凯先从科罗拉多大学取得了数学和分子生物学的双学位。后来因为对计算机很感兴趣,他最终决定去犹他州立大学读计算机科学系。读书期间,他一直在寻找某种基础构件,支持一种简单、有效的编程风格。这时他的生物学知识给了他至关重要的灵感。

亮点: 勤于思考、知识点融合、类比,产生新的思想



# 生平: 善于联想、类比



亮点: 勤于思考、知识点融合、类比,产生新的思想

我的灵感就是把这些(指对象)看做生物学上的细胞。

——艾伦·凯

- 1)每个"例"细胞都遵从"主"细胞的某些基本行为
- 2)每个细胞都能独立运作,它们之间由能透过细胞膜的化学信号进行通信
- 3)细胞会分化---根据环境不同,同一个细胞可以变成鼻子的细胞,也可以变成眼睛或者脚趾甲的细胞。



这已经是提出面向对象思想 的黎明时分了!



# 生平: 面向对象思想的推出

## 亮点:

## 预测未来的最好办法,就是把它创造出来。 ——艾伦. C.凯

1971 年夏天,艾伦.凯负责设计一种新的语言,名字叫做Smalltalk

Smalltalk的设计确实与生物学上的类比相吻合: 相互独立的个体(细胞)通过发送讯息彼此交流。 每一条讯息都包含了数据、发送者地址、接收者地址, 以及有关接收者如何对数据实施操作的指令。

凯希望把这种简单的讯息机制贯彻到整个语言中去,这些想法组成了"面向对象"的核心内容

面向对象在20世纪90年代变成了软件设计的基本方法与技术! 它更贴近人类思维的特点

# 艾伦·凯(Alan Kay)的思考



- "预测未来的最佳方式就是去创造它"
- "愿景重于目标"
- "我把对象想象成生物细胞或者和网络上的单个计算机,它们之间只能通过消息进行通讯"--作为微生物学家背景
- 每个对象可以有多个代数与之关联,这些代数关系可以构成 家族,而这些概念会非常非常有用 --作为数学家背景
- 他提倡应该关注消息机制和模块间的松耦合和交互,而不是模块内部的对象组成:
  - 做出非凡和可成长的系统的重中之重是设计好模块之间如何通讯, 而不是模块应具有什么样的内部属性和行为方法



- 4、"面向对象"思想的基本概念
  - 类 Class
  - 对象 Object
    - 注:台湾学者把Object翻译成"物件"

## 类与对象



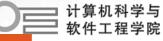
- 类和对象,这两者之间的关系
  - 有点像"先有鸡"还是"先有蛋"的关系一样,有一点纠缠
  - 从他们的定义就可以看出来:
    - 用对象来定义类、用类来产生对象!
  - 我们在学习本课程的时候,可以忽略这个问题
    - 有的时候 在表达概念的时候,对象 与 类可以通用!
  - 在设计软件系统的时候,对象是不存在的,
  - 在软系统运行时,在内存中创建对象。类不存在于物理世界





# 1. What is a Class. Object?

- A <u>class</u> is a description of a set of objects that share the same attributes, operations, relationships, and semantics "共享相同属性、操作、方法、关系或者行为的一组对象的描述符"
  - -Rumbaugh
- An object is an *Instance* created from a class. 一个对象是根据一个类创建的一个实例
  - An instance's *behaviour* and *information structure* is defined in the class. 类,定义了实例的行为和信息结构
  - Its current *state* (values of instance variables) is determined by operations performed on it. 对象的当前状态(实例变量的取值)取决于作用于该对象的操作。

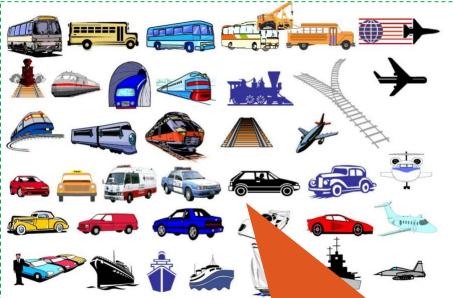


I of Computer Science oftware Engineering

## 类: 军队、部队、解放军







多个类: 汽车、火车、飞机、交通工具

学生、小学生、少先队员



omputer Science e Engineering



这是三班的陈刚

这是二排的王健康

这是一连长吴刚



对象:

这是张建伟同学、这是王海三同学 这是吴丽霞同学

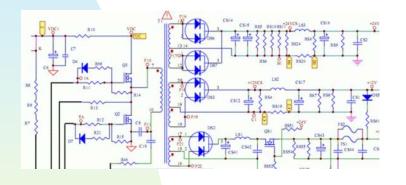


对象:

这是张三的红色小汽车 这是南航的**7892**号飞机

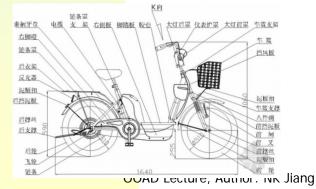
## 2、练习:类和对象

- 比喻
  - 做蛋糕的模板 --- (); 一块块蛋糕 --- ()
  - 电视机原理图 --- (); 一台台电视机 --- ()





■ 工厂生产自行车的蓝图 --- () 我的那辆黑色自行车 --- ()





School of Computer Science and Software Engineering

# 3、类的构成、对象的构成

类名

名称

对象2

属

性

取

值

# 教师

## Information:

Name, Age, Address, Friends,

## **Behaviour:**

jump
bend legs
stretch legs
lift arms,
...
walk

## John:教师

## Information:

John, 65, 华师大理科楼, 张三,

## **Behaviour:**

jump
bend legs
stretch legs
lift arms,
...
walk

talk

## Sue:教师

### Information:

Sue Smith, 25, 华师大理科楼, 李四,

### Behaviour:

jump
bend legs
stretch legs
lift arms,
...
walk

操作

属

性

talk

talk

# 3、类的构成、对象的构成

- 注意: 概念之间的互用
  - 属性Attribute == 数据Data == 状态state == 信息 information
  - 操作operation == 方法Method ==行为behaviour = = 职 责responsibility

## 4、软件功能是如何完成的?



#### ■ 类

- 定义了对象群体的逻辑结构,包括属性和操作
- 系统运行时,类作为产生对象的模板,在物理层面是不存在的

#### ■对象

- 系统运行时必须为每一个需要的对象分配内存、保存数据
- 对象存在于物理层面,每个对象都有自己的数据空间(内存)
- 所有的对象共享同一块代码空间
- ■消息
  - 对象之间的一种交流手段
    - 就像我们日常工作中的各种交流手段
- 所有相关对象之间相互协作完成软件功能



#### 计算机科学与 软件工程学院 School of Computer Science and Software Engineering

## 5、类和对象小结

- Everything is an object.
- A program is a bunch of objects telling each other what to do by sending messages.
- Each object has its own memory made up of other objects.
- Every object has a type.
  - each object is an instance of a class, in which "class" is synonymous with "type."
- All objects of a particular type can receive the same messages.
  - an object of type "circle" is also an object of type "shape," a circle is guaranteed to accept shape messages. This means you can write code that talks to shapes and automatically handle anything that fits the description of a shape.



■ 5、什么是面向对象的思考方式? (Object-oriented Thinking)

## 思维方式很重要



- 软件工程是与人打交道的、是为人提供服务的
  - 问题是什么?
  - 谁碰到了问题?
  - 问题的本质是什么?
  - 问题该如何解决?
- 对于想成为问题解决者的人来说,入门的关键是 "把单一思维模式切换成多重思维模式"
  - 一本<u>《红楼梦》</u>……经学家看见<u>《易》</u>,道学家看见 淫,才子看见缠绵,革命家看见排满,流言家看见宫 闱秘事。——鲁迅
  - 横看成岭侧成峰,远近高低各不同。——苏轼



#### 计算机科学与 软件工程学院 School of Computer Science and Software Engineering

## 什么是面向对象的思考方式?

- "观察到的一切都是对象"
  - -- 这就是面向对象思考方式



■ 软件系统是由多个对象组成,对象间通过消息相互交流、共同协作,以完成整个系统的功能





## 什么是面向对象的思考方式?

#### 定义

- 在对世界/系统 进行 观察/建模的时候,把它们看成是由一系列相互交流、互为影响的对象集(a set of objects)
- 有两点含义:
- 一,世界是由相互作用的对象组成的
- 二,描述与构建由对象组成的系统





## 面向对象的思考方式

■ 软件开发常规的两种思维方式:面向对象、面向过程

OO strategy

适合解决不确定的事件、创新性的事件

Structured Strategy

处理已知的事实、组织规划几乎没有重 要的条件仍未知的场景

- Example1, how one might design and plan a project
  - STRU: skeleton framework -> structural element
  - OO: no fixed plan of the final structure, but, small self contained subsections
- Example2, two writing strategies of a historian and a creative novelist
- Example3, 流水生产线与一场篮球赛

## 面向对象的思考方式

- C语言是一种面向过程的思维方式
  - 程序的运行"一切尽在掌握中":从main()函数的 逐条语句开始执行、调用了子程序就必须一层层返回 ,最终又返回main函数
  - 系统需要完成的功能,分配到各个子函数,由main 函数统一调度

#### ■比较

- 面向过程侧重于考虑方法的编写(哪个方法做什么事 ,不考虑所涉及的数据在哪里)
- 面向对象则致力于将数据和方法先做一个封装(分配一个对象做事,先考所需要的数据是否和它在一起)



#### 计算机科学与 软件工程学院 School of Computer Science and Software Engineering

## 面向对象的思考方式

- 比较:通信
  - <u>过程化解决方法</u>通过信道传递数据,服务器端需要有 专门的工具对接受的数据进行处理



- <u>面向对象解决方法</u>通过信道传递对象(数据+对数据的 处理方法)
  - 如,Web浏览器,接收到的Java Applet,就是一个对象。浏览器只要运行Applet里面的代码,就可以显示网页的内容
- 【误解】
  - 对象包装 (object wrap): 把任意一段过程化的代码堆 砌在一起

## 主要概念



#### ■包括

- 面向对象分析 OO analysis,
- 面向对象设计 OO design,
- 设计模式 design patterns,
- 统一建模语言 the Unified Modeling Language (UML),
- 面向对象编程 OO programming,
  - various OO programming languages,
    - C++/java/C#/.NET etc
    - and many other topics related to OO programming

## ■ 最重要的是:

 以面向对象的思维方式去思考要解决的问题 how you think in OO ways

## 面向对象的核心特征

- Following concepts are basics
  - Object / Class
  - Method / Message
  - Encapsulation
  - Inheritance
  - Interface / Implement
  - Polymorphism
  - Composition / aggregation
  - Abstraction
- Advance OO concepts
  - OO principle
  - Design Pattern



- 6、"面向对象"思考方式的核心特征(一)
  - 封装 Encapsulation
  - 继承 Inheritance
  - 多态 Polymorphism

## 1) 封装 Encapsulation

#### Encapsulation

- is the process of hiding the implementation details of an object隐藏了一个对象的实现细节
- The internal state is usually not accessible by other objects 内部的状态也不为其他对象所访问
- The only access to manipulate the object data is through its interface 对象的数据只能通过接口去访问
- Encapsulation allows objects to be viewed as 'black boxes' 封装使得对象可以被看成一个"黑盒子"
- It protects an object's internal state from being corrupted by other objects. 保护数据
- Also, other objects are protected from changes in the object implementation. 一个对象实现方法的改变,不影响其他相关对象
- Communication is achieved through an 'interface' 对象间通过"接口"进行通信



#### 计算机科学与 软件工程学院 School of Computer Science and Software Engineering

## 1) 封装 Encapsulation

- 要封装什么?
  - 内部的、不想让其他人随意了解的信息
  - 可以封装类的属性,如,"人"这个类,封装个人的工资、 资产、年龄等信息
  - 可以封装类的方法,如, "人"如何赚钱()?如何消磨时间()?
- 我们**为什**么要封装
  - 保护隐私
  - 保护数据安全
  - 隔离复杂度 (内部实现细节不对外公开)。如"空调",封装了制冷的过程,对人提供了一个制冷的按钮
- 面向对象的封装有四种方式(可见性)
  - Public
  - Private
  - Protected
  - Package



#### Rule:

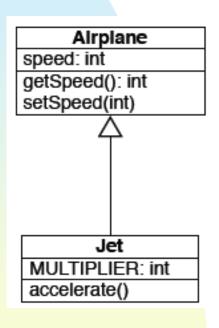
- An object should only reveal the interfaces needed to interact with it. Details not pertinent to the use of the object should be hidden from other objects.
  - This is a general rule of encapsulation

## Suggestion:

- Getters and Setters
  - supports the concept of data hiding
  - Because other objects should not directly manipulate data within another object
- For example,
  - if a user needs access to an attribute,
    - a method is called to return the attribute (a getter)

## 2)继承 Inheritance

- The basic principle is simple:
  - A class gets the state and behaviour of another class and adds additional state and behaviour



```
1 public class Airplane {
   private int speed;
   public Airplane(int speed) {
   this.speed = speed;
9 public int getSpeed() {
     return speed;
11 }
12
13 public void setSpeed(int speed) {
     this.speed = speed;
15 }
16
```

## 2)继承 Inheritance

```
public class Jet extends Airplane {
3 private static final int MULTIPLIER = 2;
5 public Jet(int id, int speed) {
6 super(id, speed);
7 }
8
9 public void setSpeed(int speed) {
10 super.setSpeed(speed * MULTIPLIER);
11 }
12
13 public void accelerate() {
14 super.setSpeed(getSpeed() * 2);
15 }
16
17 }
18
```

Note: extends keyword indicates inheritance super() and super keyword is used to refer to superclass

No need to define getSpeed() method; its inherited!

setSpeed() method overrides behavior of setSpeed() in Airplane

subclass can define new behaviors, such as accelerate()

## 2)继承 Inheritance

How do you express inheritance in C++?

# • 例 class Car : public Vehicle { public: ... };

We state the above relationship in several ways:

- Car is "a kind of a" Vehicle
- Car is "derived from" Vehicle
- Car is "a specialized" Vehicle
- Car is a "subclass" of Vehicle
- Car is a "derived class" of Vehicle
- Vehicle is the "base class" of Car
- Vehicle is the "superclass" of Car

# 3) 多态 Polymorphism

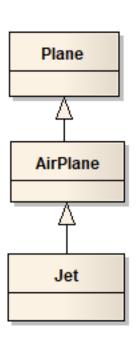
- ■本意:有多种形态 "Having many forms"
  - "When one class inherits from another, then polymorphism allows a subclass to stand in for the superclass." 当一个类从另一个类继承而来,多态使得子类可以代替父类
  - The sender of a stimulus doesn't need to know the receiver's class消息发送方不需要知道消息接收方属于那个子类
  - Different receivers can interpret the message in their own way同一类族的接收者可以按自己的方式处理消息
- 推论Consequence
  - different receivers can response to the same stimulus based on their interpretation同一类族的接收者可以按自 己的方式处理同一个消息
  - So we can have a variety of objects who process the same piece of data in different ways.有多种对象可以按自己的方式处理相同的数据

#### 计算机科学与 软件工程学院 School of Computer Science and Software Engineering

# 3) 多态 Polymorphism

- Implication: both of these are legal statements
  - Airplane plane = new Airplane()
  - Airplane plane = new Jet()

这是多态的核心思想,是设计模式的基础! "使用指向父类的指针或者引用,能够 调用子类的对象"



## 13) 多态 Polymorphism

```
public abstract class Shape{
   private double area;
   public abstract double getArea();
public class Circle extends Shape{
 double radius:
  public Circle(double r) {
    radius = r:
   public double getArea() {
      area = 3.14*(radius*radius);
      return (area);
public class Rectangle extends Shape{
  double length; double width;
  public Rectangle(double I, double w){
     length = I; width = w;
  public double getArea() {
     area = length*width;
     return (area);
```

```
Shape
                    area : Double
                                    School of Computer Science
                                    and Software Engineering
                     ♦getArea()
           Rectangle
                                       Circle
       length : Double
                                 radius : Double
       width : Double
                                  Circle()
        Rectangle()
                                  ♦getArea()
        <sup>♦</sup>aetArea()
//instantiate the Shape classes
Rectangle rectangle = new Rectangle(4,5);
```

```
Circle circle = new Circle(5);
//add these Shape classes to the stack:
stack.push(circle);
stack.push(rectangle);
```

```
care //about what kind of Shape classes
are in it:
while (!stack.empty()) {
  Shape shape = (Shape) stack.pop();
  System.out.println ("Area = " +
        shape.getArea());
```

//empty the stack, and do not have to



- 7、"面向对象"的思考方式的核心特征
  - 聚合/组合 Aggregation/Composition
  - 接口/实现 Interface /Implementation
  - 抽象 Abstraction

## 1) 聚合/组合 Aggregation/Composition

Inheritance means that one class inherits the characteristics of another class.
This is also called a "is a" relationship:

#### A car is a vehicle

A student is a person

 Aggregation describes a "has a" relationship. One object is a part of another object.

A car *has* wheels

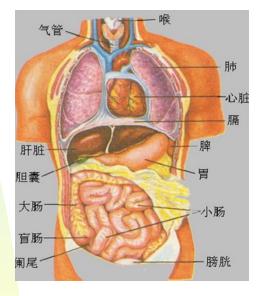
A person has legs

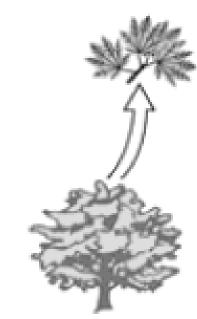
## 1)聚合/组合 Aggregation/Compositionschool of Computer Science Aggregation/Compositions School of Computer Science Aggregation (Composition of Computer Aggregation (Composition of Composition of Computer Aggregation (Composition of Computer Aggregation (Composition of Composition of Computer Aggregation (Composition of Computer Aggregation (Composition of Computer Aggregation (Composition of Computer Aggregation (Composition of Computer Aggregation (Computer Aggregation of Computer Aggregation (Computer Aggregation of Computer Aggregation (Computer Aggregation (Computer Aggregation of Computer Aggregation (Computer Aggregation (Computer Aggregatio

- Aggregations indicate that
  - one object contains a set of other objects
  - 如,大学 是由学生、老师组成的
- Aggregation relationships are transitive 传递的
  - if A contains B an B contains C, then A contains C
  - Aggregation relationships are asymmetric 不对称的
    - If A contains B, then B does not contain A
- A variant of aggregation is composition which adds the property of existence dependency 聚合 关系的一个变种, 称为组合。整体控制着部分的生命
  - if A composes B, then if A is deleted, B is deleted
  - 如手掌与手指

## 1)聚合/组合 Aggregation/Compositionschool of Computer Science Aggregation Aggre

- Composition is a type of Strong aggregation
  - 'whole' can control the life span of 'portion'
  - 部分对象只能存在于整体对象之中,整体对象控制部分对象的生命周期





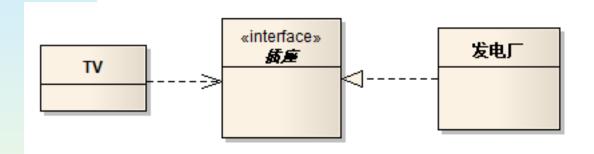
- 有些时候,聚合与组合的关系不是很明确
  - 如,房间与门

## |2)接口/实现 Interface /Implementating from the Engineering

- 生活中的例子,比比皆是
  - 电视机接口、水管接口等,举例:汽车里面有哪些接口?
- 对于软件系统
  - 软件系统的内部是由大量的互相关联的类构成的
  - 当对其中某一个类的局部进行修改的时候,不能影响其它的类
- 接口 interface
  - describe how users of the class interact with the class 描述一个 类的用户如何与这个类交互
- 实现 Implementation
  - 完成接口所定义的功能,如类、构件等完成的任务
  - A change to the implementation should not require a change to the user's code.
  - both the user and the implementation must conform to the interface specification

# |2)接口/实现 Interface /Implementation are Engineering

- For example
  - TV、电源插座、发电厂之间的关系?



- TV 是 顾客、用户
- 插座 是 接口
- 发电厂 是实现

## 3)抽象 Abstraction



- ■抽象是一种思维方式、一种思维能力
  - "抽取比较像的部分出来"
  - 客观世界的事物由各种各样的实体(对象)构成
  - 每个对象都有各自的内部状态和运动(状态)规律
  - 根据对象的属性和运动规律的相似性可以将对象分类
    - 复杂对象由相对简单的对象组成
    - 不同对象的组合及其之间的相互作用和联系构成了系统
    - 对象间的相互作用通过消息传递,对象根据所接收到的消息 做出自身的反应
  - "继承"的强大就在于它的抽象和组织技术
  - 毛主席提出的"统一战线"、"论持久战",都是高度抽象的结果
  - 抽象错了,危害很大! 如张国涛的"南下抗日!'



# 3)抽象 Abstraction



## 定义

- "抽象表示一个对象与其他所有对象相区别的基本特征,因此提 供了同观察者角度有关的清晰定义的概念界限。"
- A process allowing to focus on most important aspects while ignoring less important details
- allows us to manage complexity by concentrating on essential aspects making an entity different from others.
  - Booch
- 例如, An example of an <u>order processing</u> abstraction







- 简单地讲,
  - <u>抽象(abstr</u>action)就是过滤掉对象的一部分特性和操作,直到只 剩下你所需要的属性和操作。

65/33 2018/11/7

## 3) 抽象 Abstraction



- ■抽象与具体的比较
  - 你要到飞机场去。坐进出租车,该如何描述?
  - A: 师傅,请送我去飞机场!
  - B: 师傅, 走! 出门左拐、直行、过桥、右转、直行、 红绿灯左拐后继续直行.....
- 你觉得那个是抽象?你喜欢用哪种方式?
- 抽象是面向对象领域发现类的主要方法



■ 本讲结束