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



■ 1.5、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

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



课堂授课是 通过这些功 多个 一、学生、 一、学生、 一、等 、课桌

軟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)

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

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

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



■ 本讲结束