

# 软件测试 Software Testing

## 17. 面向对象软件测试

程适

cheng@snnu.edu.cn

计算机科学学院

2016 年 11 月 9 日



陕西师范大学  
SHAANXI NORMAL UNIVERSITY

# Outline

- 面向对象测试概述
- 面向对象的单元测试
- 面向对象的集成测试
- 基于客户角度的 Java 测试
- 基于程序角度的 Java 测试

# 面向对象软件测试

- 思维方式决定解决问题的方式。
- 传统软件开发采用面向过程、面向功能的方法，将程序系统模块化，在此基础上还可以再分成若干个单元，这些单元可以通过一系列程序的过程实现，也产生相应的单元测试、集成测试等方法。
- 面向对象程序的结构不再是传统的功能模块结构，它将开发分为面向对象分析 (OOA)、面向对象设计 (OOD) 和面向对象编程 (OOP) 三个阶段。

# 面向对象软件测试

- 分析阶段产生整个问题空间的抽象描述，在此基础上，进一步归纳出适用于面向对象编程语言的类和类结构，最后形成代码。
- 针对面向对象软件的开发特点，其测试方法和技术也必然要做相应的改变，从而形成面向对象的测试模型、测试的层次与数据流、面向对象的单元和集成测试方法等。

# 概述

- 1 类与子类的测试
- 2 分层与增量
- 3 面向对象层次结构测试重点

# 面向对象测试的问题

- 面向对象技术开发出的程序无需进行测试？
- 面向对象程序的结构不再是传统的功能模块结构，作为一个整体，原有集成测试所要求的逐步将开发的模块搭建在一起进行测试的方法已成为不可能。

# 面向对象测试的问题

- 面向对象软件测试的单元是什么？
- 面向对象软件的特征对测试的影响
  - 封装
  - 继承
  - 多态
  - 迭代的软件开发过程

# 对认定对象的测试

- OOA (面向对象分析测试) 中认定的对象是对问题空间中的结构、其他系统、设备、被记忆的事件、系统涉及的人员等实际实例的抽象。对它的测试可以从如下方面考虑：
  - 认定的对象是否全面，其名称应该尽量准确、适用，是否问题空间中所涉及的实例都反映在认定的抽象对象中。
  - 认定的对象是否具有多个属性。只有一个属性的对象通常应看作其他对象的属性而不是抽象为独立的对象。
  - 对认定为同一对象的实例是否有共同的、区别于其他实例的共同属性，是否提供或需要相同的服务，如果服务随着不同的实例而变化，认定的对象就需要分解或利用继承性来分类表示。
  - 如果系统没有必要始终保持对象代表的实例信息，提供或者得到关于它的服务，认定的对象也无必要。



# 对认定结构的测试

- 认定的结构指的是多种对象的组织方式，用来反映问题空间的复杂实例和复杂关系。
- 认定的结构分为两种：分类结构和组装结构。
- 分类结构体现了问题空间中实例的一般与特殊的关系；组装结构体现了问题空间中实例整体与局部的关系。

# 对认定结构的测试

## ■ 认定的分类结构测试要点：

- 对于结构中的一种对象，尤其是出于高层的对象，是否在问题空间中含有不同于下一层对象的特殊可能性，即是否能派生出下一次对象。
- 对于结构中的一种对象，尤其是处于同一低层的对象，是否能抽象出在现实中有意义的更一般的上层对象。
- 对所有认定的对象，是否能在问题空间内向上层抽象出在现实中有意义的对象。
- 高层对象的特性是否完全体现下层的共性，低层的对象是否有高层特性基础上的特殊性。

# 对认定结构的测试

## ■ 认定的组装结构测试要点：

- 整体（对象）和部件（对象）是否在考虑的问题空间中有实际应用，其组装关系是否符合现实的关系。
- 整体（对象）中是否遗漏了反映在问题空间中的有用部件（对象）。
- 部件（对象）是否能够在空间中组装新的有现实意义的整体（对象）。

# 对构造的类层次结构的测试

- 为了能充分发挥面向对象继承共享特性，OOD（面向对象设计测试）的类层级结构通常基于 OOA 中产生的分类结构原则来组织，着重体现父类和子类间的一般性和特殊性。在当前的问题空间，对类层次结构的主要要求是能了解空间构造实现全部功能的结构框架。为此测试要注意如下几个方面：
  - 类层次结构是否涵盖了所有定义的类。
  - 是否体现 OOA 中所定义的实例关联、消息关联。
  - 子类是否具有父类没有的新特性。
  - 子类间的共同特性是否完全在父类中得以体现。

# 面向对象层次结构测试重点

- 1 对认定对象的测试
- 2 对认定结构的测试
- 3 对构造的类层次结构的测试

# 面向对象的单元测试

- 1 方法层次的测试
- 2 类层次的测试
- 3 类树层次的测试

# 方法层次的测试

- 1 一般不会对类的每个成员及方法进行测试，例如，一般不会针对成员变量的定义进行单元测试，一般也不需要 `get/set` 方法进行单独测试，但对于核心或重要的方法需要进行全面的单元测试。
- 2 对单个方法的测试类似于对传统软件的单个函数的测试，相关的测试技术（如逻辑覆盖、路径覆盖等）都可以用在这里。

# 类层次的测试

- 1 类测试的方法
- 2 构建测试用例
- 3 Tester 类的设计举例



# 类树层次的测试

## ■ 多态服务测试

- 一般而言，子类对父类中的多态方法的覆盖应该保持父类对该方法的定义说明。
- 多态服务测试就是为了测试子类中多态方法的实现是否保持了父类对该方法的要求。
- 假设已存在父类的一个测试用例集，在对该子类的测试时，可以选取其中涉及相关多态方法的测试用例，并把子类的实例当作父类的实例来执行这些测试用例。

# 类树层次的测试

## ■ 展平测试

- 在最复杂的情况下，对于子类的测试可能需采用展平测试的策略。
- 所谓展平测试是指将子类自身定义的成员方法和成员变量以及从父类继承来的成员方法和成员变量全部放在一起组成一个新类（如果成员方法间存在覆盖关系，还需要确定哪些成员方法是子类真正拥有的），并对其进行测试。
- 展平后的类可能很大，测试的代价也较高，此时要尽可能地减少不必要的代价。

# 面向对象的集成测试

- 1 对象及其交互性测试
- 2 分布式对象测试

# 对象及其交互性测试

- 1 对象交互只不过是一个对象（发送者）对另一个对象（接收者）的请求，发送者请求接收者执行一个操作，而接收者进行的所有处理工作就是完成这个请求。
- 2 在大多数面向对象的语言中，对象交互涵盖了程序中的绝大部分活动。它包含了对象及其组件的信息，还包含了对象及其相关的其他对象之间的消息。

# 对象及其交互性测试

1 汇集类测试

2 协作类测试

# 分布式对象测试

- 1 分布式对象的概念和特点
- 2 测试中需要注意的情况

# 分布式对象的概念和特点

- 1 线程（Thread）是一个操作系统进程（Process）内能够独立运行的内容，它拥有自己的计数器和本地数据。线程是能够被调度执行的最小单元。
- 2 在设计中，基本的权衡是以线程的数量为中心。增加线程可以简化算法和技术，但线程执行的顺序出现问题的机会更大。减少线程的数量可以减少这种顺序问题，但可能会使软件更为刻板而且通常效率会更低。

# 测试中需要注意的情况

- 1 局部故障：由于以分布式系统为主的机器上的软件或硬件可能出错，分布式系统的部分代码也许就不能执行，而运行在单一机器上的应用程序是不会遇到这类问题的。
- 2 超时：当一个请求发送到另一个系统时，网络系统通过设置定时器来避免死锁。如果在指定的时间内没有得到任何的回应，系统就放弃这个请求。
- 3 结构的动态性：分布式系统通常具有依靠多种机器的加载来改变自身配置的能力，例如特定请求的动态定向。



# 基于客户角度的 Java 测试

- 1 空指针保护（Null Pointer Exception）
- 2 格式化数字错误（Number Format Exception）
- 3 字符串或数组越界错误（Out of Bounds Exception）
- 4 资源不合理使用
- 5 不当使用 `synchronized` 导致系统性能下降
- 6 调用不当方法导致结果出错

# 基于程序角度的 Java 测试

- 1 Action 层的单元测试
- 2 Biz 逻辑事务层的单元测试
- 3 Servlet 的单元测试
- 4 框架 Struts+ Spring + Hibernate 的测试

# 小结

- 面向对象测试概述
- 面向对象的单元测试
- 面向对象的集成测试
- 基于客户角度的 Java 测试
- 基于程序角度的 Java 测试

# 小结

- 1 面向对象软件与传统软件有什么不同？
- 2 面向对象中的类与子类、分层与增量是如何理解的？
- 3 面向对象软件单元测试主要做些什么？
- 4 面向对象软件集成测试常用哪些方法？
- 5 以 Java EE 软件为例，介绍各层的单元测试方法。
- 6 以 Java EE 软件为例，介绍常用工具的用途与参考资源。

# 致谢

谢谢，欢迎提问！