

# 面向对象测试技术研究

黄家伟, 曹谢东

(西南石油大学计算机科学学院, 成都 610500)

**摘要:** 面向对象程序设计语言的出现, 传统的测试方法已经不能满足面向对象软件测试的需求, 面向对象程序设计语言的特性增加了测试的复杂度, 提高了测试的难度。对于新的模式和结构, 需要通过面向对象的软件测试技术来改变传统测试的策略和方法。

**关键词:** 面向对象; 软件测试; 测试用例

## Research on object-oriented test technology

HUANG Jia-wei, CAO Xie-dong

(School of Computer Science, Southwest Petroleum University, Chengdu 610500, China)

**Abstract:** With the popularization of object-oriented programming language, the traditional test methods can not satisfy the needs of object-oriented software testing any more. The features of object-oriented programming language increase the test's complexity and the difficulty. The object-oriented software testing techniques to improve the traditional test strategies and methods for the new models and structures are necessary.

**Key words:** object-oriented; software testing; test case

## 0 引言

“程序测试是为了发现错误而执行程序的过程”<sup>[1]</sup>。软件测试是使用人工或自动手段来运行或测试某个系统的过程,其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别<sup>[2]</sup>。随着开发技术的进步,相对于传统的面向过程的软件开发技术,面向对象技术使得软件产品具有高质量、稳定性好、可重用性好和可维护性好等优点。然而,由于面向对象技术的多态、继承、封装等特性,不再是传统的功能模块结构,原有集成测试方法已成为不可能,使得传统的软件测试方法不能适应具有新特性的面向对象软件的测试。同时,面向对象技术开发的软件的代码重用率高,需要更加严格的软件测试来避免错误的繁衍。由此看来,面向对象的程序设计并没有降低软件测试的难度,相反使软件的测试变得更加复杂。面向对象开发使用的不是传统的开发模式,每个开发阶段有不同的要求和结果,已经不可能用功能细化的观点来检测面向对象分析和设计的结果。因此,传统的测试模型对

面向对象软件已经不再适用,应对面向对象的开发技术,需要提出新的测试策略和方法来支持面向对象的软件测试。

## 1 面向对象测试

面向对象测试的整体目标和传统软件测试的目标是一致的——以最小的工作量发现最多的错误。但是面向对象测试的策略和战术有很大不同,由于面向对象的特点是封装、继承和多态,测试的视角扩大到包括复审分析和设计模型,测试的焦点从过程构件转向了类。

### 1.1 面向对象测试模型

面向对象的开发模型突破了传统的瀑布模型,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。针对这种开发模型,面向对象的软件测试可分为:面向对象分析的测试,面向对象设计的测试,面向对象编程的测试,面向对象单元测试,面向对象集成测试,面向

收稿日期: 2010-03-08

作者简介: 黄家伟(1984-),男,硕士研究生,研究方向为计算机软件与理论。

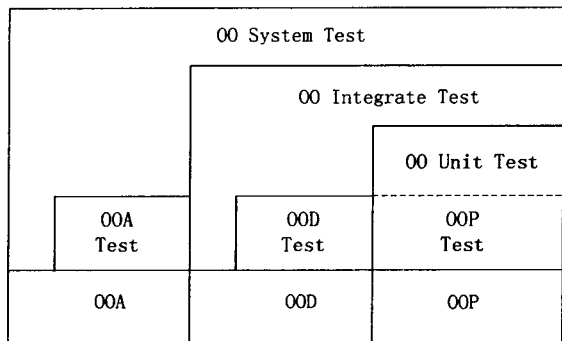


图1 面向对象测试模型

## 1.2 面向对象分析的测试

面向对象分析(OOA)是“把 E-R 图和语义网络模型,即信息造型中的概念,与面向对象程序设计语言中的重要概念结合在一起而形成的分析方法”,最后通常是得到问题空间的图表的形式描述。

OOA 直接映射问题空间,全面地将问题空间中实现功能的现实抽象化,测试重点在其完整性和冗余性,OOA 的结果是为后面阶段类的选定和实现,类层次结构的组织和实现提供平台。因此,对 OOA 的测试,应从以下方面考虑:

- (1)对认定的对象的测试
- (2)对认定的结构的测试
- (3)对认定的主题的测试
- (4)对定义的属性和实例关联的测试
- (5)对定义的服务和消息关联的测试

## 1.3 面向对象设计的测试

面向对象设计(OOD)采用“造型的观点”,以 OOA 为基础归纳出类,并建立类结构或进一步构造成类库,实现分析结果对问题空间的抽象。OOD 是 OOA 的进一步细化和更高层的抽象。OOD 确定类和类结构不仅是满足当前需求分析的要求,更重要的是通过重新组合或加以适当的补充,能方便实现功能的重用和扩增,以不断适应用户的要求。因此,对 OOD 的测试,应从如下三方面考虑:

- (1)对认定的类的测试
- (2)对构造的类层次结构的测试
- (3)对类库的支持的测试

## 1.4 面向对象编程的测试

面向对象程序具有继承、封装和多态的新特性,这使得传统的测试策略必须改变。封装是对数据的隐藏,外界只能通过被提供的操作来访问或修改数据,这样降低了数据被任意修改和读写的可能性,降低了传统程序中对数据非法操作的测试。继承是面向对象程序的重要特点,继承使得代码的重用率提高,同时也使错误传播的概率提高。多态使得面向

对象程序对外呈现出强大的处理能力,但同时却使得程序内“同一”函数的行为复杂化,测试时不得不考虑不同类型具体执行的代码和产生的行为。

## 2 面向对象测试实现

面向对象软件的最小的可测试单位是封装的类或对象,类包含一组不同的操作,并且某特殊操作可能作为一组不同类的一部分存在。面向对象的软件测试分为包括方法级测试、类级测试、类簇级测试和系统级测试。其中,类级测试是测试面向对象软件的关键。

### 2.1 示例

如图2所示类的层次结构中,基类是一个抽象类 Shape,Line 和 Quadrangle 是从 Shape 派生而来,并使用 Point 来定义其定点。其中,Quadrangle 还派生了三个类:Square、Rectangle 和 Other。类的设计如图2所示。

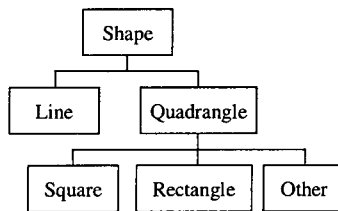


图2 Shape 及其子类结构

Point:表示二维平面上的一个点(x,y 坐标)。

Shape:表示二维平面上的基本形状,是一个抽象类,作为其余各类的基类,Point 除外,并提供平面上计算长度和面积的抽象方法,提供实现不同形状的对象之间转换的方法 ChangeTo() 和 RollBack()。

Line:表示线段,从 Shape 类派生而来,通过两个 Point 对象来定义,表示线段的两个顶点,并具体实现了所有抽象方法。

Quadrangle:表示四边形,从 Shape 派生而来,通过四个 Point 对象来定义其四个顶点,并实现了所有抽象方法。

Square:表示正方形,从 Quadrangle 派生而来。

Rectangle:表示长方形,从 Quadrangle 派生而来。

### 2.2 测试用例的设计

给定平面上的四个点 p1(1,1), p2(3,0), p3(4,4), p4(4,1),通过 Line(p1,p2), Line(p2,p3), Line(p3,p4), Line(p1,p4),由输入的四个点可以输出一个四边形:

(1)由 RollBack(p2,p3)将去掉点 p2,p3,四边形转换为由点 p1 和 p4 组成的线段。

(2)重置 p2,p3 的坐标为 p2(8,1),p3(8,4),由 ChangeTo(p2,p3)将 p2,p3 加入,正确输出是由 p1, p2,p3,p4 组成的长方形;若 p2(4,1), (下转第 102 页)

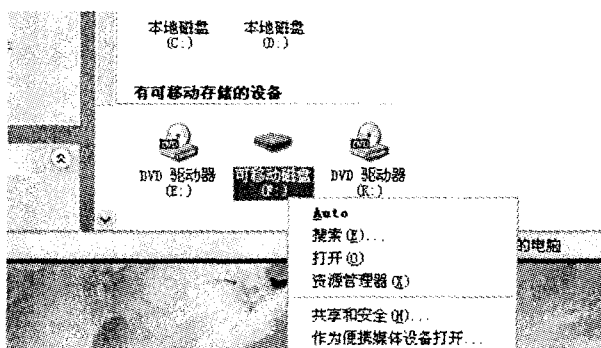


图4 右键菜单增加 Auto 选项

当双击 U 盘的时候,并不会打开 U 盘,每双击一次 U 盘,就会创建一个 U\_Virus.exe 的进程,如图 5 所示:

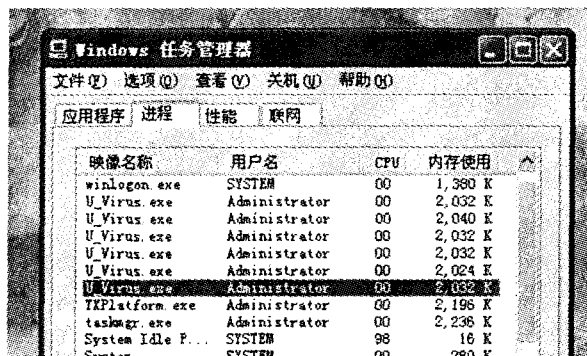


图5 病毒进程的创建

## 2 U 盘病毒的预防和删除方法

### 2.1 U 盘病毒的预防方法

目录在 Windows 下是一种特殊的文件,而两个

同一目录下的文件不能同名<sup>[3]</sup>。于是,新建一个目录“Autorun.inf”在可移动磁盘的根目录,可以防止这种情况存在的病毒创建 Autorun.inf,从而达到 U 盘病毒的预防。

### 2.2 U 盘病毒的删除方法

“开始”→“运行”→输入 CMD,之后进入 U 盘,输入 dir/a,应该可以看到 atuatorun.inf 这个文件,然后输入 attrib Autorun.inf-s-h-r,这个命令的意思是取消 Autorun.inf 文件的只读,系统只读,隐藏属性。然后输入 DEL Autorun.inf 删除这个文件。用同样的办法删除 U\_Virus.exe 和系统目录 C:\Windows\System32 下的 S\_Virus.exe 文件,之后安全删除 U 盘,再重新插入 U 盘,就可以使用了。

## 3 结束语

通过本文的实例,分析了 Autorun 病毒的基本原理,并针对本实例 Autorun 病毒提出了预防和删除方法,但是在现实生活中,U 盘病毒在不断地发展,这就需要寻求新的预防 Autorun 病毒的方法,只要了解了 Autorun 病毒的基本原理,就可以全面认识病毒,实现对病毒的预防。

### 参考文献:

- [1] 肖晓梅. 基于 U 盘的计算机病毒分析及防治[J]. 黑龙江科技信息, 2007(78).
- [2] 靳悦振. 浅谈 U 盘病毒(auto 病毒)及其防治措施[J]. 太原大学教育学院学报, 2008, 26(3): 100-103.
- [3] 范志力, 孙静丽, 包春芳, 等. 关于 U 盘及移动硬盘病毒的分析与处理[J]. 赤峰学院学报: 自然科学版, 2008, 24(4): 3-4.

责任编辑: 么丽苹

(上接第 99 页)

p3(4,4),则由 ChangeTo(p2,p3)将输出由 p1, p2,p3,p4 组成的正方形。

(3)置 p2. x = 2, p2. y = 2, p1, p2, p3 三点将变为特殊的共线关系, p2 位于线段 p1p3 的线段上, 四点无法组成四边形。

(4)置 p2. x = 1.5, 四点无法组成凸四边形。

全面的测试用例设计可以采用以下策略<sup>[4]</sup>:

第一,根据方法特性将被测类的方法划分为构造函数、功能函数和接口函数。

第二,对于构造函数,列出所有前置条件和后置条件的组合,根据不同的组合设计测试用例。

第三,对于功能函数而言,对所有公有方法列出前置条件和后置条件,根据各种有意义的组合设计测试用例;对所有受保护的方法,严格区分有访问权限和无访问权限的前置条件和后置条件设计测试用例;对所有私有方法,根据实际情况选用合适的策略进行测试。

第四,对于接口函数,应绘制类的状态转换图,根据该图设计测试用例,覆盖到每种类的状态和状

态的转换。

实际测试中,以上的情况都应结合多种基本的测试方法来选择测试数据。

## 3 结束语

面对面向对象技术开发出来的程序具有更好的结构更规范的编程风格,极大地优化了数据使用的安全性,提高了代码的重用率。同时,也影响了软件测试的方法和内容,增加了传统软件设计技术所不存在的错误,增加了软件测试的难度。面向对象测试技术能解决传统测试方法的不足,同时,更高效、快速、全面的测试技术以及自动化测试是面向对象测试技术所需解决的,以适应要求更高、功能更强大的软件系统。

### 参考文献:

- [1] Myers Glenford J, Badgett Tom, Thomas Todd M, et al. The Art of Software Testing[Z]. John Wiley & Sons Inc. 2005.
- [2] 路晓丽, 葛玮, 龚晓庆, 等. 软件测试技术[M]. 北京: 机械工业出版社, 2007.
- [3] 柳纯录. 软件评测师教程[M]. 北京: 清华大学出版社, 2005.
- [4] 武剑洁, 陈传波, 肖来元. 软件测试技术基础[M]. 武汉: 华中科技大学出版社, 2008.

责任编辑: 么丽苹