面向对象软件测试及测试用例设计

一、引言

软件测试是伴随着软件的产生而产生的。软件危机的频繁出现促使了软件测试的地位得到了大幅提升。软件测试已经不仅仅是局限于软件开发过程中的一个阶段，它已经开始贯穿于整个软件开发过程，成为软件产品质量控制与质量管理的重要手段之一。软件测试技术作为软件工程学科的一个分支，是保证软件质量和可靠性的关键，因此它也是软件开发过程中的一个重要环节。它的核心思想是：对于输入域的特定输入，观察软件的执行结果，验证该结果与期望结果是否一致，然后根据结果作相应的纠错和调整。在测试过程中，测试用例的选择决定测试的有效性，这也就直接影响到成本，是软件测试的关键和难点。目前，软件测试技术的发展还不是很成熟，测试人员在选择测试用例时通常根据直觉和经验进行，给测试带来很大的盲目性，最终导致的后果是使软件后期维护的费用在成本中居高不下。科学生成测试用例对提高软件质量不仅重要而且必要。

随着面向对象软件开发技术的广泛应用和软件测试自动化的要求，特别是基于的软件开发技术的逐渐普及，基于模型的软件测试逐渐得到了软件开发人员和软件测试人员的认可和接受。它是一种新兴的测试用例生成技术。有优于以前的测试技术的方面。其中模型以其定义良好、功能强大、普遍适用的优点，为基于模型的测试提供了非常好的契机。

二、面向对象特征对软件测试的影响

面向对象技术是一个全新的开发模式，具有以下特点　　：

(1)它要综合考虑软件开发过程所有阶段。

(2)在软件开发的整个生存周期中，每个阶段之间是连续的。

(3)开发过程分为面向对象分析　(00A)、面向对象设计　(OOD)、面向对象编程　(OOP)、面向对象测试　(OOT)四个连续的部分。　Coad 和　Yourdon 给面}向对象的概念下了一个定义　：

面向对象　=对象 +类 +继承 +通信

如果一个软件系统是使用这样　　4 个概念设计和实现的，则认为这个软件系统是面向对象的。一个而向对象的程序的每一个组成部分都是对象，计算是通过对象和对象之间的通信来执行的。

面向对象技术的本质是定义了类的抽象，将变量和与作用于它的操作封装到一块。然后用不同的类和方法组合成一个对象系统。面向对象软件将传统软件中的一个过程或一个方法内的复杂性转移到对象之间的交互中。面向对象语言一些本质特征形成了如下的一些新的故障、错误风险。

1、基本功能模块

在面向对象系统中，系统的基本构造单元是封装了数据和方法的类和对象，而不再是一个个能完成特定功能的功能模型。每个对象有自己的生存期，有自己的状态。消息是对象之间相互请示或协作的途径，是外界使用对象方法及获取对象状态的唯一方式。对象的功能是在消息的触发下，由对象所属类中定义的方法与相关对象的合作共同完成，并且对象在不同状态下对消息的响应可能完全同。

工作过程中，对象的状态可能被改变，产生新的状态，即发生状态的转移。对象中的数据和方法是一个有机的整体，在软件测试过程中，不能仅仅检查输入数据产生的输出结果是否与预期结果相吻合，还要考虑对象的状态变化。因此，除了要对对象的状态与方法间的相互影响进行测试，还要进行状态测试。

2、系统的功能实现

在面向对象系统中，系统的功能体现在对象间的协作上，而不再是简单的过程调用关系。面向对象程序的执行实际上是执行一个由消息连接起来的方法序列，方法的实现与所属对象本身的状态有关，各方法之间可能有相互作用。为实现某一特定的功能，可能要激活调用属于不同对象类的多个成员函数，形成成员函数的启用链。因此，基于功能分解的自顶向下或自底向上的集成测试策略不适用于面向对象软件系统的测试。

3、封装对测试的影响

封装是指在词法单位之中或之间决定名字可见性的访问控制机制。它支持信息的隐蔽和模块化，有助于防止全局变量访问的问题。尽管封装不会直接促成错误的发生，它却给测试带来了障碍。封装使对象的内部状态隐蔽，如果类中未提供足够的存取函数来表明对象的实现方式和内部状态，则类的信息隐蔽机制将给测试带来困难。

4、继承对测试的影响

继承也是面向对象语言中的一个本质特征。继承可用于一般与特殊关系，并目方便编码。但继承削弱了封装性，产生了类似于非面向对象语言中全局数据的错误风险。由于继承的作用，一个函数可能被封装在具有继承关系的多个类中，子类中还可以对继承的特征进行覆盖或重定义。

5、多态对测试的影响

多态性是指一个引用可以与多个对象绑定的能力。多态能减少代码的复杂性和规模，同时还可以实现动态绑定。但依赖于不规则的类层次的动态绑定可能产生编程人员没有想到的结果。某些绑定能正确的工作但并不能保证所有的绑定都能正确地运行。以后绑定的对象可能很容易将消息发送给错误的类，执行错误的功能，还可能导致一些与消息序列和状态相关的错误。

三、面向对象软件测试的层次划分及内容

面向对象软件测试的测试工作过程与传统的测试一样，分为以下几个阶段：制定测试计划、产生测试用例、执行测试和评价。目前，面向对象软件测试划分方法是：方法测试、类测试、类簇测试、系统测试。

1、方法测试

方法测试主要考察封装在类中的一个方法对数据进行的操作，它与传统的单元模块测试相对应，可以将传统成熟的单元测试方法。但是，方法与数据一起被封装在类中，并通过向所在对象发送消息来驱动，它的执行与对象状态有关，也有可能会改变对象的状态。因此，设计测试用例时要考虑设置对象的初态，使它收到消息时执行指定的路径。

2、类测试

主要考察封装在一个类中的方法与数据之间的相互作用。一个对象有它自己的状态和依赖于状态的行为，对象操作既与对象状态有关，又反过来可能改变对象的状态。普遍认为这一级别的测试是必须的。类测试时要把对象与状态结合起来，进行对象状态行为的测试。类测试可分以下两个部分：

（1）基于状态的测试

考察类的实例在其生命期各个状态下的情况。这类方法的优势是可以充分借鉴成熟

的有限状态自动机理论，但执行起来还很困难。一是状态空间可能太大，二是很难对一些类建立起状态模型，没有一种好的规则来识别对象状态及其状态转换，三是可能缺乏对被测对象的控制和观察机制的支持。

（2）基于响应状态的测试

从类和对象的责任出发，以外界向对象发送特定的消息序列来测试对象。较有影响的是基于规约的测试方法，和基于程序的测试。基于规约的测试往往可以根据规约自动或半自动地生成测试用例，但未必能提供足够的代码覆盖率。基于程序的测试大都是在传统的基于程序的测试技术的推广，有一定的实用性但方法过于复杂且效率不高。

3、系统测试

系统测试是对所有类和主程序构成的整个系统进行整体测试，以验证软件系统的正确性和性能指标等满足需求式样说明书和任务书所指定的要求。它与传统的系统测试一样，包括功能测试、性能测试、余量测试等，可套用传统的系统测试方法。

四、面向对象软件测试的覆盖准则

测试覆盖标准是对软件测试充分性的度量，任何测试策略都应该有相对应的覆盖标准，在此基础上选择测试用例，通过覆盖率来说明测试结果的可信性。传统的软件测试的覆盖标准对基于代码的测试而言，主要是代码覆盖，可细分为语句覆盖、路径覆盖、分支覆盖、判定分支覆盖，其中判定分支是最强的覆盖标准。对于面向对象软件的测试而言代码的覆盖只对方法级测试适用，为了测试面向对象程序设计机制带来的错误风险，必须引入新的覆盖标准。对于继承而言，要求达到继承的上下文覆盖；多态性要求覆盖所有可能的消息绑定；基于代数规约的测试要求能达到公理的覆盖；基于状态的测试要求能覆盖所有的状态和转换。

五　面向对象软件测试用例的设计

解决方案：

测试用例设计我们可以通过过程和方法两方面来考虑，并且借鉴程序设计过程和方法来设计我们的测试用例，让我们测试用例结构清晰、易读、可维护性强，提高设计测试用例质量和效率。

一、过程

我们可以参见程序设计过程跟我用例设计过程进行对比。请大家不要误会的是，这里的对应不是阶段、时间的对应，而是程序设计过程和测试用例设计过程中活动对照。

编号	开发过程	用例过程	我们做什么？
1	需求分析	测试需求	根据需求拆分、合并、整理成我们测试需求。
2	概要设计	初步设计	根据测试需求组装、拆分成我们测试用例大纲，并确定测试大纲包含各种测试有效并且是主导性分支。 要设计多少测试用例（模块），其中哪些测试用例可以做成公用用例（接口）
3	说细设计	用例编写	根据本文 3.2 方法和大家熟知的白盒、黑盒测试用例设计方法来设计我们的测试用例。
4	代码编写	用例执行	根据设计并且评审好的测试用例来执行测试。

我们可以将用例中各中元素， 根据当前系统业务， 加上自己一点想象力有机组合， 把它设计成一个个公用对象、公用函数、变量、结构体等，下表简单举例几种情况。

测试用例中内容	可设计对象	举例说明	
数据	变量	用户属性，病人属性，厂家属性等	实例 1
	结构体	药品列表，费用列表，用户列表等	实例 2
	……		
公用测试用例	公用函数	界面规范，安全规范，SQL 语句规范等	实例 3
	公用对象	查询控件，报表设置控件、折旧方法对象等	实例 4
	……		
……			

实例 1：病人属性医保性质，原来两种 "省医保 "，"市医保 "，我们测试用例编写快结束的时候，客户说要增加一个 "区医保 "，病人医保性质几乎要涉及到 60%的测试用例，要以前我们可能要花好几天来更新所有测试用例， 现在只要花上两小时就可以了， 只要在医保性质测试用例增加一个 "区医保 "政策及算法的子用例就行了。

实例 2：药品信息我们设计成一个数据集，有一次项目进行了设计变更药品信息表中要加一个批次的字段， 我们只要在药品信息数据集中增加批次即可， 更新测试用例只花几分钟的时间。

实例 3：如果我们有一个产品升级项目上，离上次升级已经有 3 年了，原有测试用例有好多涉及界面规范，安全规范， SQL 语句编写规范内容，这几年这些规范了也更新好几个版本，如果我采有面向对象的测试用例设计方法我们只要更新一下用例库规范内容就行了。

二 测试用例

该测试案例是以一个 B/S 结构的登录功能点位被测对象 , 该测试用例为黑盒测试用例。假设用户使用的浏览器为 IE6.0 SP4 。

功能描述如下：

1. 用户在地址栏输入相应地址，要求显示登录界面；
2. 输入用户名和密码，登录，系统自动校验，并给出相应提示信息；
3. 如果用户名或者密码任一信息未输入，登录后系统给出相应提示信息；
4. 连续 3 次未通过验证时，自动关闭 IE。

表 4-1 登录界面测试用例

用例 ID	XXXX-XX-XX	用例名称	系统登录
用例描述	系统登录 用户名存在、密码正确的情况下，进入系统 页面信息包含：页面背景显示 用户名和密码录入接口，输入数据后的登入系统接口		
用例入口	打开 IE，在地址栏输入相应地址 进入该系统登录页面		

测试用例 ID	场景	测试步骤	预期结果	备注
TC1	初始页面显示	从用例入口处进入	页面元素完整，显示与详细设计一致	
TC2	用户名录入－验证	输入已存在的用户： test	输入成功	
TC3	用户名－容错性验证	输入： aaaaabbbbbccccccddddddeeeee	输入到蓝色显示的字符时， 系统拒绝输入	输入数据超过规定长度范围
TC4	密码－密码录入	输入与用户名相关联的数据： test	输入成功	
TC5	系统登录－成功	TC2， TC4，单击登录按钮	登录系统成功	
TC6	系统登录－用户名、密码校验	没有输入用户名、密码，单击登录按钮	系统登录失败，并提示：请检查用户名和密码的输入是否正确	
TC7	系统登录－密码校验	输入用户名，没有输入密码，单击登录按钮	系统登录失败，并提示：需要输入密码	
TC8	系统登录－密码有效性校验	输入用户名，输入密码与用户名不一致，单击登录按钮	系统登录失败，并提示：错误的密码	
TC9	系统登录－输入有效性校验	输入不存在的用户名、密码，单击登录按钮	系统登录失败，并提示：用户名不存在	
TC10	系统登录－安全校验	连续 3 次未成功	系统提示： 您没有使用该系统的权限， 请与管理员联系！	
…	…		…	…