**快递管理系统（EMS）测试回顾**

本学期的项目实践已经接近尾声了，构造阶段完成之后，进入了测试阶段。本阶段进行的是系统测试，在系统测试前，首先对之前做过的单元测试和集成测试做一个回顾，从中可以发现一些问题，并适当地做出改进。

**单元测试的回顾**

之前在详细设计的时候写过JunitTest单元测试和mockobject，那个时候是根据详细设计文档中规定的各个类的各个方法接口来确定测试用例的。但是当时真正的代码还没有完成，文档中写的很多内容和后来的代码有许多不一致的地方，所以当时写的很多测试都是没有实际意义的，因为测试的那个方法可能已经被修改了。而且文档中写的内容不是很完整，后来在构造阶段，实际代码中新增了一些类和方法，这些新增的都是当时单元测试没有测试到的，所以单元测试覆盖率不是很高，导致了在后续测试中，还要回过头来重新进行部分单元测试来弥补之前的空缺。之所以出现这个情况，主要还是在设计阶段做得比较仓促，没有设计得很完善，漏洞比较多。还有就是当时对mockobject的认识还不是很深刻，看了书之后，也只是一知半解，因此当时写的mockobject没有做到尽可能的简单，导致了出现错误时，首先担心的是mockobject内部有没有问题。而实际上，mockobject应该是简单得不可能出错的，所以是无需检查的。

下面来说说单元测试中测出的问题。一个很常见的错误就是空指针异常，在测试中经常会发现没有对某个引用变量进行初始化而出现的错误。其次，一些边界值比较容易出问题。比如查询经营情况表要输入一个开始日期和一个结束日期，当时测试时忘记了开始日期等于结束日期的情况，后来发现当开始日期等于结束日期时无法得到正确的结果。在测试中还发现了一个很奇怪的问题，当时百思不得其解，后来上课听老师说了才恍然大悟。在计算成本收益表的时候，需要根据所有付款单计算总支出，根据所有收款单计算总收入，然后在计算利润。但是，当时测试时，发现计算利润的那个方法总是没能通过，后来进一步检查发现问题是出在计算总支出上，计算出的总支出应该是24.9，可是无论怎么算结果都是24.00000009。检查了好几遍都感觉代码没有任何问题。后来课上老师说double型变量计算时本身就会有问题，才终于知道了为何会产生这个问题。

**集成测试的回顾**

最早进行集成测试是在体系结构设计时，写的那些stub和driver程序。当时也没有很明白做这个工作的用处，只是模仿书上的例子写了一下。跟上面提到的问题类似，stub程序本身应该是要求很简单的，不会出错的，但是当时我们也没有注意到这个；还有那个时候，很多接口的规格跟后来实际代码相差较多，所以当时做的一些测试在后来并没有任何意义，同样地，也有很多后来加上来的内容，之前没有测试到。由于我们小组的集成频率比较低，没有做到持续集成，所以大部分时候都是大家各自写各自的代码。当要使用到别人写的模块的方法时，都是参照详细设计文档，但是并不知道实际代码有没有发生更改。比如在写查询经营情况表时，需要用到另一位成员写的付款单po、vo，根据最初的详细设计文档付款单没有institution这一项，但是后来被加上去了，而调用的时候仍然是根据原来的，所以产生了错误。类似的问题发生过好多次，所以后来我们加强了交流，有什么重大的变化必须及时通知其他人。由于不是持续集成，所以每次集成遇到一些交互比较复杂的类，问题就层出不穷，有时候debug要花费好多时间，才最终得以解决。当然有很大一部分原因还是因为当时各种接口的规格定义得不是很清晰，也不是很可靠，易于变化。这启示我们在做详细设计时，应该要加强交流，需要什么样的接口都讨论清楚，这样就可以减少接口的变化，为后续的集成测试做好铺垫。

以前自己写程序时，很少系统地进行过测试活动，所以这次项目实践中，由于缺乏经验，测试时遇到了很多问题。当然，这也并不是什么坏事，吃一堑长一智，有了这次的经验，好好反思一下，以后就会注意这些遇到的问题，在今后的实践中更上一层楼。