

由一个简单的程序谈起——之六（精华）

江苏 无锡 缪小东

本篇主要和大家讨论设计代码之外的一些观念、看法！只当作闲聊放松！

一、给你一道智力题

我估计在你看这篇的时候你已经研究了前面 3 篇文章，那是一个网友在群中为他的程序求助时我发现这个程序可能问题多多，先分析，然后就试图写一下这个程序！不管怎么样对一些要交毕业设计的朋友可能是一个帮助！同时通过这个简单的程序告诉初学者一些设计理念。希望大家会喜欢。

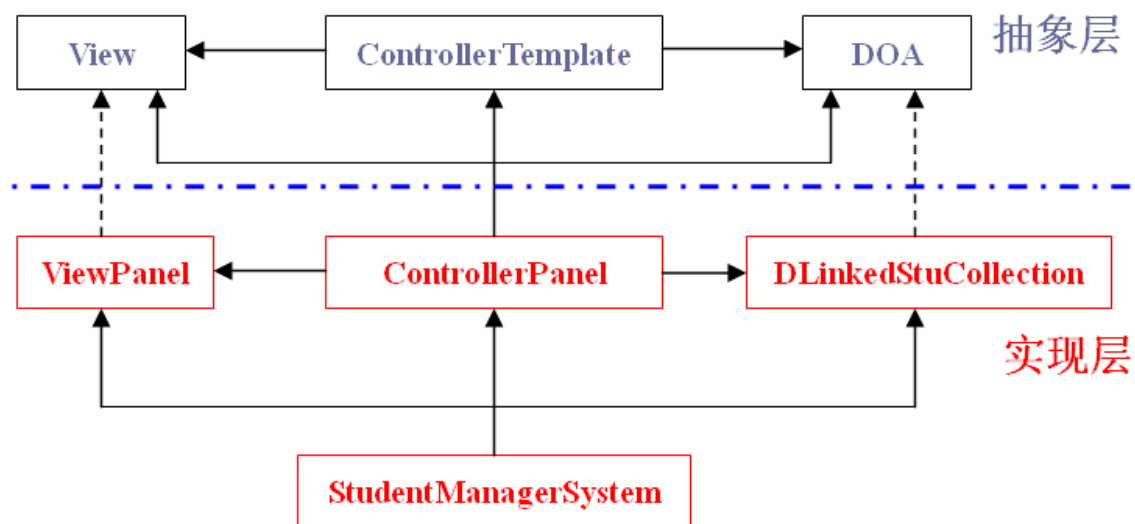
程序也许就是 20—30 页的代码，我两天的工作量！希望大家能有所所获！其实很明确地说写这个程序不是为了程序本身，而是告诉大家程序设计背后的东西！经过上面的阅读大家可能比较郁闷毕竟都是代码。下面给大家两道智力题吧！

1. 去玩过欧洲嘉年华的朋友都会看到这样一个游戏：一个平滑的圆圆的玻璃下面是一些化好的半径为 R 的圆圈，你可以使用 5 元钱买一个半径为 r 的游戏币，你向此玻璃上投，游戏币刚好在圆圈内你就可以拿到一个很好玩的毛茸茸的玩具——价值 30 元左右！假设某个人投射很准每次几乎都可以投到这个圆圈相连的地方，问投入圆圈中的几率！（这是我去嘉年华看到的游戏，一看到我就知道大家在上当！可惜很多北京、上海的有钱的公子哥都会去玩！痛心！悲哀！这个智力题会告诉大家一个道理“知识就是力量”）

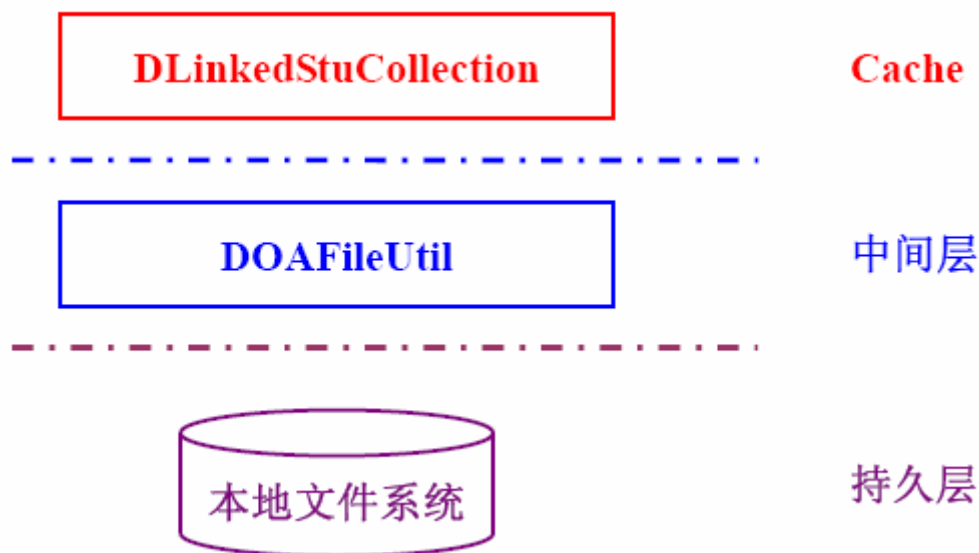
慢慢想吧！本篇的末尾会公布答案！

二、 系统的构建

我将此系统分为两层——所谓的抽象层和实现层。抽象层主要包含 View 和 DOA 两个接口以及一个表示控制器抽象业务流程的 Controller 类。实现层包括一个视图的实现、一个控制面板的实现以及一个特殊持久层 DLinkedStuCollection 的实现。下面是系统层次图：



上面就是他们的关系！结合代码仔细研究吧！下面是持久层的简要说明图：



DLinkedStuCollection 可以分为三部分：1.Cache，其中保存系统运行后存于内存中的记录，这个记录是根据场景变化的；2.中间层，主要完成将 Cache 中的元素保存入本地文件系统或者将本地文件系统中的记录加载到 Cache 中，这层主要是 DLinkedStuCollection 中的内部类 DOAFileUtil 类实现的；3.持久层，就是本地文件系统了！

在 2 和 3 之间我们根据记录的性质的不同将记录保存到不同的文件中，具体的可以多阅读代码体会吧！

三、本程序背后的东西

在构建本程序开始的时候，是没有这些很具体的想法！头一天写了 GUI 部分，并且各个模块都测试没有什么错误！然后第二天就是整个数据结构 DLinkedStuCollection 部分，刚开始也只是为了实现 DOA 接口，后来发现需要一些其它的方法，方法加入了，也可以运行了！发现程序的逻辑不是很清楚，所以就不停地 Refactor，其中也出现不少问题，例如链表的指针错误，链表的位置等等！最后发觉可以给大家一个有一定意义的程序，也就发布！不想再做

进一步的 Refactor，大家可以在此基础上继续改进吧！

3.1 程序中使用到的知识

一个简简单单的程序使用了很多方面的知识！如：Swing、IO、数据结构。

谈到数据结构 DLinkedListCollection 本身就是一个有趣的数据结构，其中使用了 Java Collections Framework 中的 List、LinkedList、Arrays、Collections 等几个类，您是否对这些类很熟悉很熟悉呢！（没有！那就想去仔细研究吧！基本得不能再基本了！除了会用他们，还得知道其内部的具体实现！这样在必要的时候可以自己建立某些特殊的数据结构啊！万丈高楼平地起——从基础开始吧！）

Swing 部分使用了 JButton、JComboBox、JSpinner 等等。其实整个 Swing 比这个多多了！我特意使用了大家很少用到的 JSpinner 和 JSplitPane 有意思吧！只要你掌握背后的理念就可以很熟练的使用他！在注释部分谈到 Swing 中使用比较多的“组合模式”和“MVC 架构”，这两个在 Swing 中太有代表性了，对了还有一个 BorderFactory 的使用了 FlightWeight 模式，我也在 Student 类中演示了一下，同时提及使用他的必要性，也就是使用场景了！好东西无处不在，关键是你有没有发觉！只有当你真正理解 oo，熟读 Swing 你才有可能将 Swing 的优点发挥得淋漓尽致！国外去年有一本关于 Swing 的比较高级的书籍《Swing Hacker》其实就是在理解了 Swing 的设计理念后，给读者一些简单的应用而已！看了几页，其实挺简单的都是一些表层的应用而已！你也想读懂，甚至能创建比他更好的程序吗？！那就请先研究 Design Pattern，然后看看 Swing 的 Source code！希望你早日成功！

在任何要保存数据的程序中都会使用 IO 的。任何语言或者系统的设计者在

设计语言或者系统时，IO 都是一个极其复杂的部分。这主要是因为 IO 的形式复杂多样很难有统一的标准。Java 是继 Linux 之后第一个采用将 IO 抽象为流的概念！java 的 IO 采用了装饰器模式 “Decorator” 看起来很怕人，其实是一种简单思想的强大应用而已，只要你知道背后的理念，你就会很好地使用他！最近又在看 IO 的源码，发觉比以前看有意思多了！一切都是那么简单、那么顺其自然、行云流水般……理解了 IO 后你就可以按照其理念任意地扩展其功能，为自己所用！爽吧！博客中其它文章已经介绍了一下 IO 的架构，请向前查找吧！IO 主要包含大量的流类和几个特殊的类。这些类的使用方法也是极其简单的！例子中给出了典型的构造流的方法，一般就是那样构造的！体会吧！不就博客中我们将深入了解 IO 包，不仅讲述其设计理念，还挖掘其源代码实现的细节，以及对其扩展等等。关注吧！！

程序中注册部分大量地谈到如何组织这个程序，如何设计他等等。可能你已经发现其中包含大量供调试用的 `System.println` 语句！不要奇怪，这么小的程序我也不太可能使用 JUnit 来写一些单元测试的代码，所以我选用最最基本的方法：观察程序执行的步骤和结果以及程序执行时内存中记录的一些状态。提醒初学者不要看到错误就害怕！慢慢查吗！！想办法查吗！！其它的方法不会，`System.println` 也该会吧！上面所谓的“学生管理系统”其实不能算一个系统，最最起码对程序的错误处理的支持不是很到位，最简单的程序一般要记录程序运行的状态、程序错误的原因等等——这就是软件设计的 Log 机制，应该必须的吧！

你可能发现我的程序中包括 n 多的所谓冗余的方法。其实这在软件设计中也是一个不可缺少的部分！不是说一定有，而是说可以有！经常阅读一些国外的软件发现：很多时候很多情况下那些冗余方法检测的错误可能在这段代码中

根本就不可能发生，但是作为一个一般的程序，程序的设计者还是加了这些代码！很有趣吧！

其实象这些小问题还是很多很多的！例如：在写方法的时候，方法的参数一般都是想要判定其是否为 `null`，不信你看看 `java.util` 包中的代码；一般都会在执行具体操作之前检查程序的状态，如在 `IO` 中一般会有 `ensureOpen` 之类的方法检查被 `Wrapper` 的流是否已经关闭、在 `net` 包中的类一般会有一些检查当前 `Socket` 是否关闭的方法、在数据库的源代码中也会有 `XXXOpen` 之类的代码……这样不仅逻辑明确，而且会减少出错的机会；对于写条件语句也是有一些想法的，比如一般总会将发生几率较高的放到最前面判断，或者按照次序进行，中不能一会儿东一会儿西吧！你写也麻烦别人读也麻烦！……太多太多了自己体会吧！

3.2 题外话

写一个程序甚至一个系统，首先要求你对语言必须比较熟悉！最重要的你必须要有想法，看到某种需求是否有灵感！灵感不会从天而降的，这就在于你平时的学习了！可能有很多人只学了一点基本知识，然后就开始写程序甚至开始搞什么什么项目了！不过你愿意你就那么干吧！编程和其它行业一样要有经验要有积累，一辈子闭门造车应该不可能写出好的程序！要善于学习学习别人的有意义的东西。教者授之以渔，学的人呢！最起码得之鱼吧！高明一点的应该会学之渔了！

学会别人知识固然很好，但是学会做人做事可能更为重要！经常在网络上海谢谢文章发发贴之类的！遇到形形色色的人可能在所难免！什么对模式知之甚少者，甚至可以说对模式根本就不了解的竟然大谈模式没有什么用处，竟然会

以 HelloWorld 程序中没有用到模式作为依据等等！更有甚者开口闭口就谈 RUP！甚至通过那种极为下流的“贬低别人抬高自己”的手法来做事！唉，难免让人恶心！只能劝你多读书！且少出来丢人现眼！

提醒大家一句——我一直推荐那些想高效使用或者研究 java 的人学学设计模式，这里还推荐大家在学精设计模式后继续阅读一些开发方法学之类的书籍！应该对你很有用处的！比如敏捷开发、极限编程、RUP、Refactor 等等！我读过一些，没有完全融会贯通所以当前不会给大家具体的指导！有朋友也推荐我看看《人件》、《人月神话》等等，有空我会看的！

我们做程序做的是一种理念、一种方法……请自强、自重、自爱吧！

四、题目的答案

这其实是道概率的题目。计算游戏币是在大圈之中的概念，其实就是计算游戏币的圆心在两者半径差 $R-r$ 的小圆中的概率！能明白吧！仔细想想，游戏币跑道两者半径差 $R-r$ 的小圆边界和外面是什么情形。一下就是可能的面积——圆圈的面积和有效面积——小圆的面积的比了！就是半径的平方比了！嘉年华比较恶心的就是游戏币的半径和大圆的半径几乎相近，且他的台子是玻璃的很多游戏币都跑到台子的边缘了——注意台子的边缘没有大圈了！小圆的半径很小，大圆半径相对较大，同时几率又是面积比！概率够小吧！同时又是玻璃的！概率更小更小的吧！——这叫坑你没商量！其中很多项目如两个球滚入一排有分数的空挡中也差不多！知识重要吧！有了知识就不会被人蒙了！20 天左右营业额 1—2 个亿啊！骗的都是大家的钱，也不知道地方政府得了多少好处！

下篇将 Collections 的算法吧！毕竟这个程序中使用了不少！

2006-12-28

更多精彩请关注：

<http://blog.163.com/miaoxiaodong78/>