

软件系统分析与设计课程论文

姓名：岳宇轩 学号：19020011038

第八小组组长 指导教师：张春海

在软件系统分析与设计这门课上，我主要学习了面向对象的思想，以及如何使用面向对象思想解决复杂问题，尤其是如何将面向对象思想与关系型数据库相结合的问题，并且将所学知识应用于实践，作为小组组长带领组员开发图书管理系统。

首先，若想对软件系统进行分析与设计，首先要对软件工程有一个较为明确的认知。软件是一种无形的和可塑的产品。软件的开发往往是一个学习的过程，软件的设计也经常随着知识的获得、项目与产品出现的范围而改变。传统的工程项目，比如一栋楼房的建造工程，它的参与人员必须是对相关知识和技术有非常全面的掌握才行的，而且在建造的过程中，楼房的设计是不能随意改变的。总体来说，这是一项非常“规矩”的工程。然而这并不是说软件工程的人员就可以没有专业素质，并不是说软件工程的设计就可以随意改变，只是相比之下，软件工程开发是一个更有弹性、更灵活的过程。在接一个项目之前，工程团队首先要对项目的可行性进行分析，通过调查研究和测试实践等方式来判断该项目是否可行。其中可能会用到某些从来没使用过的工具或者技术，但这并不代表该项目是不可行的。比如我们小组在进行图书管理系统的开发时，自己订立需求，并对该项目的可行性进行了分析。我们认为项目的难点在于实现 B+树建立索引以及实现不定长记录。虽然我们没有亲手在软件开发中实现以上两点的经验，

但基于网上搜集的相关资料以及过去所学的知识，我们认为这是可以实现的。在考虑如何存储用户借书信息这一问题时，我们首先想到的是使用数组存储。然而随着课程的深入学习，我们又采用了文件的存储方式，然后我们又学习到了对文件的抽象和封装。因此，软件工程的开发需要进行可行性分析，而且软件系统的设计也是会随着诸多因素的改变而改变的。软件系统的开发是一个相对来讲更加灵活的过程。

软件系统通常来讲都是一个相对复杂的工程。面向对象模型为解决软件工程中的复杂问题带来可行性。在关系型数据库的基础上引入面向对象思想，能有效的组织和管理不同类型的数据。那么，如何将面向对象的思想应用到我们图书管理系统的开发中呢？

首先，抽象是指对某一类事物的共同特征进行提取，譬如学生都有学号、姓名等属性，有借书、还书等操作，那我们就可以将所有的学生都抽象成一个类 **Student**。这是我在上这门课之前所理解的抽象。在这门课中，我真正领悟到了：抽象并不仅可以对自然界中客观存在的事物进行抽象，而是对任何具有共同特征的事物都可以进行抽象。像刚才举到的 **Student** 类的例子，就是对自然界中存在的客观事物进行抽象，与之相似的还有 **Book** 类等等。然而，使用文件进行信息存储的过程中，这一个个的文件难道不也是具有共同的特征吗？难道不也可以进行抽象吗？对于存储信息的文件（即数据库中的一张张 **table**），他们具有共同的属性，就是 **name**（属性名），**type**（类型），和 **length**（长度）；他们有共同的操作，就是增、删、查、改。因此，我们完全可以对文件进行抽象，将它们抽象成一个 **File** 类。我认为，

这是我上这门课中最大的收获。

其次，封装是指类通过访问控制权限保护类中的某些属性和操作，向外界提供接口，而屏蔽接口中的具体实现。比如 **File** 类的 **createTable** 操作，是创建一个文件（创建一张表），我们在软件系统分析时可以进行判断，说所有 **File** 类产生的对象的属性数量不会超过 256。存储 **Book** 类的属性有 **Isbn**、书名、作者、出版社等等（图 1），**User** 类的属性有姓名、性别、学号、专业等等。但不论是哪一个，都不会超过有 256 个属性。这样，我们可以在 **createTable** 函数中写入如下操作：在创建某一张表时，先进行文件的创建，然后在前 256 行中，每一行写入一个属性的属性名、属性类型、属性长度。比如我在第一行写入 **isbn char* 13**，在第二行写入 **book_price float 32**（图 2）。从第 257 行开始进行数据的存储（使用二进制文件进行存储），这样我在读取某一条数据的时候就可以知道我要按照什么格式去读取多长的数据，读出来的数据是代表什么属性的。这样，当我们想创建一张表时，直接调用 **createTable** 函数，并传递相应参数即可。在创建表的过程中，由于采用了面向对象的思想，我们并不与要对每一个参数都进行具体的含义解析，而是对他们一视同仁，进行相同的处理，即可得到我们想要的结果。这体现了面向对象模块化的特点。



	A	B	C	D
1	isbn	书名	作者	单价
2				
3				

图 1：数据库中一张表的表示



文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)
isbn	char*	13		
书名	char*	100		
作者	char*	100		
单价	float	32		

图 2：将图 1 所示表的控制信息写入文件头

面向对象模型还有其他特征。我们通常可以将模型分为五个层次：主题层、类与对象层、结构层、属性层、服务层。主题层用来控制读者同一时间所能考虑的模型规模，类与对象层是对事物的抽象，结构层描述类属成员的构成，属性层即数据元素，服务层是指对消息的响应。OOA 的第一步是标识对象，也就是说我们要分析事物的属性和操作，从而进行一定的抽象。比如，我可以抽象出 **User, Book, File**。第二步是标识结构，在问题空间里找出从一般到特殊的继承关系进行分类，找出整体和部分的关系进行组装。比如，可以使 **Student** 和 **Admin** 类都继承自 **User**。有了继承才叫面向对象，只有类与对象叫做基于对象。第三步是对五层结构的每一层进行具体的定义。将面向对象的思想运用到数据库存储中，可以使得数据存储更加持久化。我们在存储信息的时候，先把对象序列化成流存入数据库中，在读取信息时把流反序列化成对象。这样，对象信息的维持就不仅限于程序的运行时了，而是可以将它存储到外存中，可以在程序下一次运行时复现，实现对象持久化。我们也可以给对象设置一些并发控制属性来控制数据库文件的读写问题，从而使面向对象模型具有并发性的特征。

在这门课的学习过程中，我对面向对象思想有了更加深入的认知，比如面向对象模型的特征以及如何分析面向对象模型。举个例子，我

现在有一个类表示圆（**Circle**），有一个类表示长方形（**Rectangle**），这属于标识对象的过程。他们都可以继承自图形类（**Shape**），而如果我現在要想要一个圆形和正方形组合起来的图形类（**Combination**），让它直接继承自 **Circle** 和 **Rectangle** 是不合理的做法，因为其中没有 “is a” 的语义关系，将 **Circle** 和 **Rectangle** 进行组合显然更加合理，这属于标识结构的过程。在将面向对象模型运用到关系型数据库的实践——开发图书管理系统的过程中，我更加清晰的体会到了面向对象思想对于数据库信息存储的复杂性问题带来的帮助。对于可能会存储不同信息的文件，我们对他们进行抽象，通过在文件头写入控制信息字段的方式，实现对文件操作的统一管理。由此，可以将复杂的数据存储问题模块化，更利于我们去编程实现。同时，它也可以带来对象持久化、并发访问等方面的好处。本课程在要求我们开发一个图书管理系统，我认为其中其最核心的要求是采用面向对象模型自己实现一个关系型数据库。在进行软件系统的开发过程中，我不断思考文件结构，如何存储信息和如何对文件信息进行检索，这对于我理解面向对象思想和数据库设计都有着莫大的帮助。