

哈尔滨工业大学国家示范性软件学院

2015 本科《程序设计实践》报告

题 目： 图书管理系统

班 级： 1537105

项目组编号： 32

| 项目组成员： | 姓名 | 学号 |
|--------|------------|-------------------|
| | <u>薛仲豪</u> | <u>1153710506</u> |
| | <u>夏维邑</u> | <u>1153710515</u> |
| | <u>梁天材</u> | <u>1153710507</u> |
| | <u>葛俊岩</u> | <u>1153710523</u> |

指导教师： 郭勇

开始日期： 2015 年 07 月 06 日

完成日期： 2015 年 07 月 17 日

目 录

| | |
|------------------------|----|
| 第 1 部分 项目开发计划 | 1 |
| 第 2 部分 系统需求规格说明书 | 3 |
| 第 3 部分 系统设计说明书 | 11 |
| 第 4 部分 用户使用说明书 | 19 |
| 第 5 部分 项目总结报告 | 20 |

第 1 部分 项目开发计划

1 引言

1.1 编写目的

编写本项目开发计划书，旨在指导项目的开发，以里程碑和时间表的方式保证项目能够按时完成，并符合用户的要求。

1.2 背景

本项目要设计并实现的系统名称为：“图书管理系统”，项目由软件学院教研室提出。

图书馆数字化和网络化是新的发展方向，管理好藏书资料，有的放矢地不断补充新图书，让人们能更好更方便地从中获取新知识，提高技术水平，将更好地提高技术层次。要提高图书管理水平，就必须使图书管理从传统的手工管理服务模式走向计算机化。建立联机书目信息数据库，实现联机检索和资料共享是发展的第一步，开发图书管理系统软件正是为了实现这一步骤。

使用计算机对图书信息进行管理，具有手工管理所无法比拟的优点。例如：检索迅速、方便、准确、统计精确、便于分析等。这些优点能极大地提高图书管理的水平。

1.3 定义

评价标准：产生评语的依据，例如阶段性考试成绩，思想表现等。

2 实施计划

2.1 工作任务的分解与人员分工

薛仲豪：系统功能结构、java 与数据库连接，图书管理系统各种类的编写

夏维邑：借阅者界面及其自界面，实践报告

梁天材：登陆界面，管理员界面

葛俊岩：类的文档生成，数据收集，debug，代码规范

2.2 进度

进度安排如表 1-1 所示。

表 1-1 项目开发进度安排

| 日期 | 要完成的任务 | 任务执行人 |
|-----------|------------------------|-------|
| 7.5-7.6 | 数据库的建立 | 薛仲豪 |
| 7.6-7.8 | 图书管理系统各种类的编写 | 薛仲豪 |
| 7.8-7.10 | 图书管理系统登陆界面，管理员界面 | 梁天材 |
| 7.10-7.13 | 借阅者界面及其自界面，实践报告 | 夏维邑 |
| 7.13-7.15 | 类的文档生成，数据收集，debug，代码规范 | 葛俊岩 |

2.3 关键问题

(注：主要写出开发本系统的关键问题有哪些？比如查询算法的设计、文件结构的设计、类的类的合理划分等等。至少要写：300 字)

关键字：数据库，界面，类

数据库：构造一个初始化的 `LibraryDatabase` 类，用来自动与数据库进行连接，将收集到的书名，isbn 码，作者信息，读者姓名，id 号等信息录入数据库。建立好数据库，就可以很方便的调取删除或者新添信息，可以为后来的查询等其他事件提供很大的便利。

界面：界面主要分为登陆界面，借阅者界面，管理员界面还有一些子界面组成。通过登陆界面可以到达借阅者界面和管理员界面，其中管理员界面采用可视化编程并利用最适合的布局来排列组件，而登陆界面和借阅者界面采用的是常规编程。它们都使用了 `java swing` 组件来编写。

类：我们的图书馆里系统里定义了几个并列的类，比如：`Book` 类，`Frame` 类，`Event` 类，`Person` 类等，每个类具有各自的功能。比如：`Person` 类里主要都是一些方法，可以返回个人的一些个人信息。

第 2 部分 系统需求规格说明书

1 功能模块的确定

1.1 编写目的

本说明书描述“图书管理系统”的软件需求，目的在于明确说明系统的需求，说明系统需要实现的具体功能，系统运行所需的环境，系统对数据的输入输出要求。为系统数据库结构设计、概要设计、详细设计与编码实现提供基础的指导。

1.2 系统功能结构图

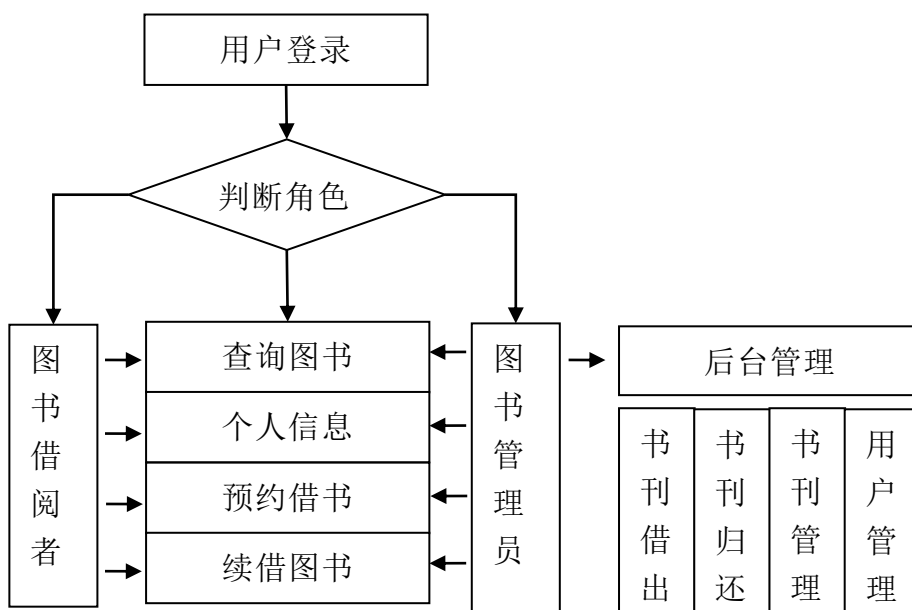


图 1-2

1.3 系统功能模块具体说明

用户登录：图书借阅者和图书管理员都从此处登陆。

判断角色：根据登陆者的 ID，判断是借阅者还是管理员。

查询图书：查询图书信息，包括图书的名字、作者、出版社、出版时间、ISBN 码、破损程度、是否被借出、是否被预约。

个人信息：查看个人的信息，包括姓名、证件类型、证件号、已借图书、还书日期，若超过限定日期，显示应付费用、历史记录。

预约借书：预约借出图书（最多提前三天）。

续借图书：增加借阅图书的时间。

书刊借出：借出图书，需要写借阅者 ID、借书时间、自动显示金额。

书刊管理：可以增加、删除图书、查看图书状态。

用户管理：添加、删除用户、查看用户记录。

2 任务概述编码规范的制定

2.1 目标

对于任何工程项目来说，统一的施工标准都是保证工程质量的重要因素。堪称当今人类最抽象、最复杂的工程——软件工程，自然更加不能例外。高品质、易维护的软件开发离不开清晰严格的编码规范。本文档详细描述使用 C++ 进行图书管理系统开发过程中的编码规范。本规范也适用于所有在文档中出现的源码。

2.2 具体编码规范

1 JAVA 源文件的组织结构

PACKAGE 的组织

Package 是组织相关类的一种比较方便的方法。Package 使我们能够更容易查找和使用类文件，并可以帮助我们在运行程序时更好的访问和控制类数据。类文件可以很容易的组织到 Package 中，只要把相关的类文件存放到同一个目录下，给该目录取一个与这些类文件的作用相关的名称。如果需要声明程序包，那么每个 JAVA 文件 (*.java) 都需要在顶部进行 Package 的声明，以反映出包的名称。

JAVA 源文件的内部结构

所有的 JAVA(*.java) 文件都应遵守如下的样式规则，如果 JAVA 源文件中出现以下相应的部分，应遵循如下的先后顺序。

编码时，即使某个类不是 `public` 类型的，也要在一个独立的 `JAVA` 文件（*.java）中声明，避免一个 `JAVA` 文件包含多个类声明。版权信息应在 `JAVA` 文件的开头，其他不需要出现在 `JavaDoc` 中的信息也可以包含在这里。

Package/Import

`Package` 行要在 `import` 行之前，`import` 中标准的包名要在本地的包名之前，而且按照字母顺序排列。如果 `import` 行中包含了同一个包中的不同子目录，应 `import` 到某一个指定的类，避免*类型的 `import`。

Class

类的注释一般是用来解释类的，建议使用文档注释。接下来是类的定义，有可能包含了 `extends` 和 `implements`。

Class Fields

`public` 的成员变量应使用 `JavaDoc` 注释，`protected`、`private` 和 `package` 定义的成员变量如果名字含义明确的话，可以没有注释。注释成员变量时，建议使用文档注释。

存取方法

如果存取方法只进行简单的赋值或取值操作，可以写在一行上，否则不要写在一行上。

构造函数

构造函数应该按照参数数目的递增顺序进行书写。每个类都应具有至少一个构造函数。

克隆方法

如果这个类是可以被克隆的，那么下一步就是 `clone` 方法：

类方法

对于类的具体方法，应将功能相似的方法放置在一起。

toString 方法

每一个类都应该定义 `toString` 方法。

main 方法

每个文件的主类中应有一个 `main` 方法。`main` 方法应该提供简单的测试和 `demo` 功能，它应该写在类的底部。

2 命名规则

JAVA 源文件的命名

JAVA 源文件名必须和源文件中所定义的类的类名相同。

Package 的命名

Package 名的第一部分应是小写 ASCII 字符，并且是顶级域名之一，通常是 com、edu、gov、mil、net、org 或由 ISO 标准 3166、1981 定义的国家唯一标志码。Package 名的后续部分由各组织内部命名规则决定，内部命名规则指定了各组件的目录名，所属部门名、项目名等。

Class/Interface 的命名

Class 名应是首字母大写的名词。命名时应该使其简洁而又具有描述性。异常类的命名，应以 Exception 结尾。Interface 的命名规则与 Class 相同。

常量的命名

常量名的字母应全部大写，不同的单词之间通过下划线进行连接，并且名字组合应该赋予含义。

普通变量

普通变量名的首字母小写，其它每个单词的首字母大写。命名时应该使其简短而又有特定含义，简洁明了的向使用者展示其使用意图。

约定变量

所谓约定变量，是指那些使用后即可抛弃（throwaway）的临时变量。通常 i、j、k、m 和 n 代表整型变量；c、d 和 e 代表字符型变量。

方法的命名

方法名的第一个单词应是动词，并且首字母小写，其它每个单词首字母大写。

方法参数的命名

应该选择有意义的名称作为方法的参数名。如果可能的话，选择和需要赋值的字段一样的名字。

样式结构

整体样式

缩进单位

一个缩进单位为四个空格，缩进排版时以缩进一个单位为最小缩进量。不要使用制表符（Tab 键），因为不同的系统对它的解释不尽相同。

缩进和对齐

缩进

当某行语句在逻辑上比下面的语句高一个层次时，该行下面的语句都要在该行的基础上缩进一个单位。

对齐

若干语句在逻辑上属于同一层次时，这些语句应对齐。

行宽

为了和 linux,unix 等字符界面的操作系统兼容，JAVA 代码行应限制在 80 个字符之内，多余部分应换行。

断行规则

当一句完整的语句大于 80 个字符时需要断行，断行时，应遵循下面规则。在逗号后换行

在操作符前换行

换行后，应和断行处的前一层次对齐

换行时尽量选择高层次的地方进行换行

在使用上述的规则换行后对齐时，如果次行的长度大于 80 个字符，应改用两个单位的缩进来代替层次对齐。

空白的使用

空格字符的使用

- 1) 关键字和括号()之间要用空格隔开
- 2) 参数列表中逗号的后面应该使用空格
- 3) 所有的二元运算符，除了"."，应该使用空格将之与操作数分开
- 4) 强制类型转换后应该跟一个空格
- 5) 左括号右边和右括号左边不能有空格
- 6) 方法名与其参数列表的左括号之间不能有空格
- 7) 一元操作符和操作数之间不应该加空格，比如：负号("-")、自增("++")和自减("--")

空白行的使用

空白行将逻辑相关的代码段分隔开，以提高可读性，有如下几种情形：

- 1) 一个源文件的两个片段(section)之间用两个空白行
- 2) 两个类声明或接口声明之间使用两个空白行
- 3) 两个方法的声明之间使用一个空白行
- 4) 方法内的局部变量和方法的第一条语句之间使用一个空白行
- 5) 块注释或单行注释之前使用一个空白行
- 6) 一个方法内的两个逻辑段之间应该用一个空白行

注释样式

JAVA 程序有两类注释，实现注释(implementation comments)和文档注释

(document comments)。

实现注释，就是使用 `/*...*/` 或 `//` 界定的注释。文档注释，又被称为 "doc comments" 或 "JavaDoc 注释"，是 JAVA 独有的，由 `/**...*/` 界定，并且文档注释可以通过 JavaDoc 工具转换成 HTML 文档。

在注释里，应该对设计决策中重要的或者不是显而易见的地方进行说明，但应避免对意思表达已经清晰的语句进行注释。

特别注意，频繁的注释有时反映出代码的低质量。当你觉得被迫要加注释的时候，考虑一下是否可以重写代码，并使其更清晰。

实现注释

块注释

块注释通常用于提供对文件，方法，数据结构和算法的描述。块注释被置于每个文件的开始处以及每个方法之前。它们也可以被用于其他地方，比如方法内部。在功能和方法内部的块注释应该和它们所描述的代码具有一样的缩进格式，并且，块注释之首应该有一个空白行，用于把块注释和代码分割开来。

单行注释

单行注释显示在一行内，并与其后的代码具有一样的缩进层次。如果一个注释不能在一行内写完，应采用块注释，且单行注释之前应该有一个空白行。

尾端注释

尾端注释与它们所要描述的代码位于同一行，但是应该有足够的空白来分开代码和注释。若有多个尾端注释出现于大段代码中，它们应该具有相同的缩进。

行末注释

行末注释的界定符是 `"/`"，它可以注释掉整行或者一行中的一部分，一般不用于连续多行的注释文本。但是，它可以用来注释掉连续多行的代码段。

文档注释

置于 `/**...*/` 之中的注释称之为文档注释。

文档注释用来描述 Java 的类、接口、构造器、方法以及字段(field)，一个注释对应一个类、接口或成员，该注释应位于声明之前，与被声明的对象有着相同的缩进层次。

在类的声明中，各种类、接口、变量、常量、方法之前都应该有相应注释。

变量的声明

一行只声明一个变量

声明变量时要对其进行初始化，如果是类中的变量，声明时不初始化，在构造函数中一定要初始化

临时变量放在其作用域内声明

声明应集中放在作用域的顶端

避免声明的局部变量覆盖上一级声明的变量

类和接口的声明

当编写类和接口时，应该遵守以下规则：

在方法名与其参数列表之前的左括号 "(" 间不要有空格

左大括号 "{" 位于声明语句同行的末尾，并与末尾之间留有一个空格

右大括号 "}" 另起一行，与相应的声明语句对齐。如果是一个空语句，"}" 应紧跟在 "{" 之后

方法与方法之间以空白行分隔

语句

简单语句

每行至多包含一条完整语句

在没有必要的情况下，不要在 `return` 语句中使用括号

复合语句

复合语句是包含在大括号中的语句序列，形如 "{ 语句 }"，其编码应有如下基本规则：

被括其中的语句应该比复合语句缩进一个层次

左大括号 "{" 应位于复合语句起始行的行尾，并且空一个空格，右大括号 "}" 应另起一行并与复合语句首行对齐

复合语句即使只有一个语句，也要有大括号作为界定

每行至多包含一条完整语句

判断语句

选择语句

在选择语句中应添加 `default` 情况，防止不可预知的情况发生。

当一个 `case` 在没有 `break` 语句的情况下，它将顺着往下执行。应在 `break` 语句的位置添加注释。[下面就含注释 `/* falls through */`]

循环语句

在 `for` 语句的初始化或更新子句中，如果存在多项，各项间应用逗号隔

开。同时，应避免使用三个以上子句，从而导致复杂度提高；若确实需要，可以在 for 循环之前放置初始化子句或在 for 循环末尾放置更新子句。

一个空的 for 语句和 while 语句只用一行

Try-Catch 结构语句

典型示例

- 1) 方法的返回值表达应尽量简单。在 return 语句后一般不使用括号，但是如果返回中包含复杂表达式，则要使用括号。
- 2) 比较对象用 equals() 方法代替操作符 “==”。特别是不能用 “==” 去比较字符串类型的变量。
- 3) if、while 等语句中的条件判断部分，应将常量（如果有）写在 “==” 运算符的左边，而把变量写在右边。
- 4) 在每一个 package 中包含一个 package.html 文件用于对包的简要说明。

第 3 部分 系统设计说明书

1 编写目的

本说明书描述“图书管理系统”的设计，目的在于明确系统的总体设计方案，定义系统的逻辑结构、数据结构、接口以及异常或出错处理方法，、编码实现与测试提供基础的指导。

本说明书的预期读者为系统开发人员、系统测试人员。

2 系统设计

2.1 系统规定

2.1.1 硬件环境

软件运行所需的硬件设备如表 3-1 所示。

表 3-1 系统运行所需硬件

| 硬件要求 | 计算机 | 业务服务器 |
|------|---------------|---------------|
| 处理器 | 奔腾 II200 以上 | 奔腾 II450 以上 |
| 内存 | 64MB 以上 | 256MB 以上 |
| 网络 | 10/100M 自适应网卡 | 10/100M 自适应网卡 |
| 硬盘 | 3G 以上 | 10G 以上 |

2.1.2 处理的功能要求

本项目主要以数据文件为基础，开发“图书管理系统”

2.2 系统总体结构

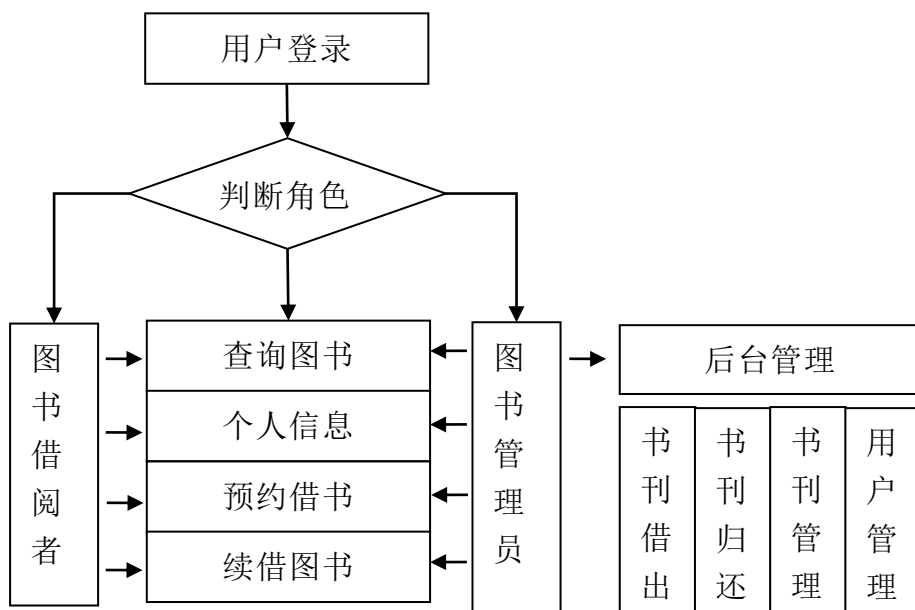


图 2-2

2.3 类的设计

将开发“图书管理系统”所用的类的详细信息列在下表中。

表 3-2 系统类的设计

表格 2-5

Book

| 限定符和类型 | 方法和说明 |
|------------------|---|
| java.lang.String | <code>getAuthor()</code> 获取作者 |
| double | <code>getIntegrity()</code> 获取书的完好程度 |
| java.lang.String | <code>getIsbn()</code> 获取书的 isbn 码 |

续表 2-5

| | |
|------------------|--|
| boolean | getLent() 获取书是否被借出 |
| int | getLentDate() 获取借出书的日期 |
| java.lang.String | getLentPerson() 获取借出书的人 |
| java.lang.String | getName() 获取书名 |
| boolean | getOrdered() 获取书是否被预定 |
| int | getOrderedDate() 获取预定书的日期 |
| java.lang.String | getOrderedPerson() 获取预定书的人 |
| java.lang.String | getPress() 获取出版社 |
| int | getPublicationDate() 获取出版时间 |
| void | setAuthor (java.lang.String author) 设置作者（默认为空字符串""） |
| void | setIntegrity (double integrity) 设置完好程度（默认为-1，是一个不合法的值） |

续表 2-5

| | |
|------|--|
| void | setIsbn (java.lang.String isbn) 设置书的 isbn 码 |
| void | setLent (boolean lent) 设置是否被借出，若被借出，则应当继续设置 lentPerson 、 lentDate ；若未被借出， lentPerson 、 lentDate 的值不需考虑 |
| void | setLentDate (int lentDate) 设置借出时间（默认为-1） |
| void | setLentPerson (java.lang.String id) 设置借书的人（的 id） |
| void | setName (java.lang.String name) 设置书名（默认为空字符串""） |
| void | setOrdered (boolean ordered) 设置是否被预定，若被预定，则应当继续设置 orderedPerson 、 orderedDate ；若未被借出， orderedPerson 、 orderedDate 的值不需考虑 |
| void | setOrderedDate (int orderedDate) 设置预定时间（默认为-1） |
| void | setOrderedPerson (java.lang.String id) 设置预定的人（的 id） |
| void | setPress (java.lang.String press) 设置出版社（默认为空字符串""） |
| void | setPublicationDate (int publicationDate) 设置出版时间（默认为-1） |

Event 类

| 限定符和类型 | 字段和说明 |
|------------------|---|
| static int | BORROW 事件的类型之一，表示借书，值为 0 |
| static int | RENEW 事件的类型之一，表示续借，值为 2 |
| static int | RETURN 事件的类型之一，表示还书，值为 1 |
| 限定符和类型 | 方法和说明 |
| double | <code>getAfterIntegrity()</code> 获取事件发生之后书的完好程度 |
| double | <code>getBeforeIntegrity()</code> 获取事件发生之前书的完好程度 |
| int | <code>getDate()</code> 获取事件的时间 |
| int | <code>getId()</code> 获取事件的序号 |
| java.lang.String | <code>getIsbn()</code> 获取事件所对应的书的 isbn 码 |
| java.lang.String | <code>getManagerId()</code> 获取事件所对应的管理员的 id |
| int | <code>getType()</code> |

| | |
|------------------|---|
| | 获取事件的类型 |
| java.lang.String | getId() 获取事件所对应的借阅者的 id |
| void | setAfterIntegrity (double afterIntegrity) |
| void | setBeforeIntegrity (double beforeIntegrity) |
| void | setDate (int date) |
| void | setId (int id) |
| void | setIsbn (java.lang.String isbn) |
| void | setManagerId (java.lang.String managerId) |
| void | setType (int type) |
| void | setUserId (java.lang.String userId) |

LibraryDatabase 类

| 限定符和类型 | 方法和说明 |
|-----------------------------|--|
| Book | getBook (java.lang.String isbn) 从数据库中获取 isbn 码对应的图书，并写入一个 Book 类，返回此 Book 类 |
| java.util.LinkedList<Event> | getEventByBook (java.lang.String isbn) 续表 2-5 从数据库中获取 isbn 对应的图书的所有历史事件 |
| java.util.LinkedList<Event> | getEventByPerson (java.lang.String id) 从数据库中获取 id 对应的人的所有历史事件 |
| java.lang.String | getPasswd (java.lang.String id) 获取一个用户 id 对应的密码 |

| | |
|---|---|
| Person | <code>getPerson(java.lang.String id)</code> 从数据库中获取 id 对应的用户，并写入一个 Person 类，返回此 Person 类 |
| boolean | <code>saveBook(Book book)</code> 将一个书的信息存入数据库（只用于新建图书之时即可） |
| boolean | <code>savePerson(Person person, java.lang.String passwd)</code> 将一个人的信息存入数据库（只用于新建用户即可） |
| <code>java.util.LinkedList<Book></code> | <code>searchBook(java.lang.String st)</code> 模糊搜索图书 |

2.4 类之间的关系图

画出类之间的继承关系。

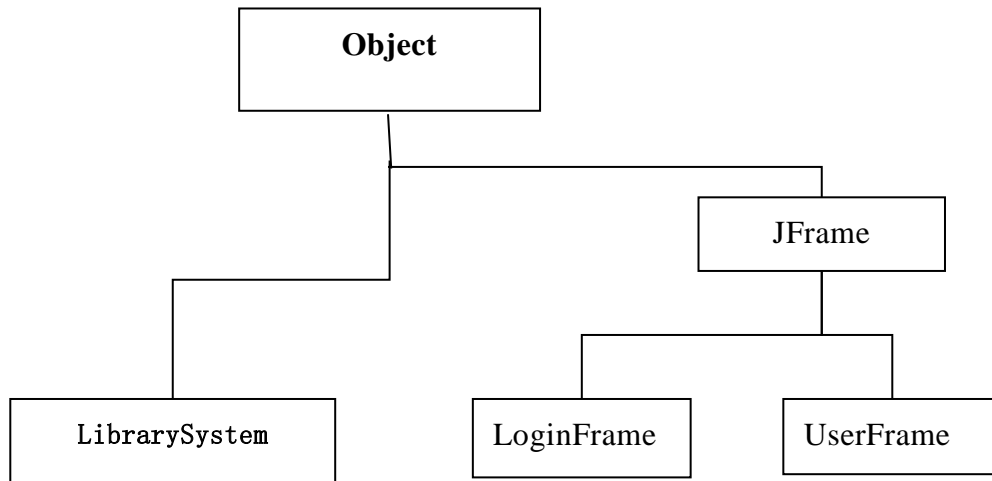


图 2-4

2.5 数据存储设计

| 表名 | 字段 | 数据类型 |
|--------|------------------|---------|
| book | Isbn | Bigint |
| | Name | Varchar |
| | Author | Varchar |
| | Press | Varchar |
| | Publication_date | Int |
| | Integrity | Double |
| | Lent | Tinyint |
| | Lent_person | Bigint |
| | Lent_date | int |
| | Ordered | Tinyint |
| | Ordered_person | Bigint |
| | Ordered_date | Int |
| person | Id | Bigint |
| | Passwd | Varchar |
| | Id_type | Tinyint |
| | Name | Char |
| | Lent_a | Bigint |
| | Lent_a_date | Int |
| | Lent_b | Bigint |
| | Lent_b_date | Int |
| | Lent_c | Bigint |
| | Lent_c_date | Int |
| event | Id | Int |
| | type | Tinyint |
| | Isbn | Bigint |
| | User_id | Bigint |
| | Manager_id | Bigint |
| | Before_integrity | Double |
| | After_integrity | Double |

采用人工的方式整理测试工作得到的数据，并基于这些数据得出测试分析报告。

第 4 部分 用户使用说明书

1 软件安装配置方法

写出所开发的“图书管理系统”的安装方法。
双击安装包，选中同意选项即可安装。

2 系统使用说明书

在本节分模块与功能介绍本系统的使用方法。

2.1 系统登录功能

系统登录界面如图 7-5 所示。

图 7-5



图 7-5 系统登录界面

在系统登录功能界面中，输入“用户名”与“密码”，点击登录按钮即可完成登录功能。

第 5 部分 项目总结报告

1 编写目的

为从项目中遇到的问题获取教训，用本报告总结项目中遇到的问题，并表述项目开发的系统的功能与性能，总结在项目开发周期中犯的错误，为今后项目的开发积累技术与经验。

本项目开发总结报告的预期读者为：项目组成员，项目提出者。

2 开发工作评价

3.1 对产品质量的评价

主要写一下你们小组开发的系统是否满足预期的要求，使用是否方便，灵活性怎样。

这个图书馆里系统的功能符合最开始的预期，基本功能具备。比如：图书查询具有模糊搜索的功能，可以更加方便借阅者进行查找目标书籍，在个人信息的界面还可以进行续借。界面精简，使用起来比较方便，容易上手。

3.2 对技术方法的评价

对本项目的开发中所用的技术进行评价。

本次图书馆里系统的设计底层方面主要使用了我们从没接触过的数据库，在学习数据库的使用方面付出了较大努力。在最开始的构想上思路清晰，类的设计比较合理，为之后的设计减少了不必要的工作量。并且利用编程时利用 `github` 可以让小组成员间的代码配合更容易，提高了合作效率。

3 经验与教训

写一下通过本项目的开发，获得的经验及教训。

通过本次的项目开发，小组成员对于 `java` 语言的理解加深了，学习到了之前没通过的知识，对于以前没学懂的知识又一次的学习并且熟练了。对于 `eclipse` 的使用更加熟练，还提前接触到了以后要学的数据库的知识。最重要的是学会了团队配合，大家相互支持，彼此鼓励，完成项目的同时也加深了友谊，对于软件工程专业的热情更加高涨。同时我们也都认识到了各自

身上的不足之处，比如小编的编程水平还需要提高，算法能力太低，查找相关知识的搜索能力还需要增强。