**总体方案设计说明**

**2015211313班 2015211516李子茜 2015211516吕滢**

1.软件开发环境CFree

2.总体结构图

**Main\_v5.h**

**user.cpp**

**Main.cpp**

**admin.cpp**

**books.cpp**

**Log.cpp**

**Kmp.cpp**

3.功能模块划分

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **模块文件** | **模块说明** | | **模块包含的函数名** | | **函数功能** | |
| Main.cpp | 1. 用户数据，书籍数据初始化 2. 创建时间轴进程，再无操作时向前推进 3. 跳转进入用户界面 | | 无 | | 无 | |
| Admin.cpp | | 1. 管理员模块 2. 管理员函数 3. 管理员页面跳转声明 | | void admin\_page() | | 管理员用户界面函数 | |
| void admin\_choices() | | 管理员选择界面函数，根据用户的不同选择调用不同的函数 | |
| admin::admin() | | Admin类无参数构造函数 | |
| void admin::add\_books() | | Admin类函数增加图书函数 | |
|  | |  | | void admin::delete\_books() | | admin类删除图书函数 | |
|  | |  | | void admin::add\_user() | | Admin类增加使用者函数 | |
|  | |  | | void admin::delete\_user() | | Admin类删除使用者函数 | |
|  | |  | | void admin::search\_user() | | Admin类查找用户函数 | |
| Books.cpp | | 1. 全部图书类函数声明 2. 初始化图书信息 函数 3. 从文件种穿件图书信息函数 | | books::books() | | Books类无参数构造函数 | |
| books::books(string set\_ID,string set\_name,string set\_author,string set\_press,string set\_isbn,int set\_time,int set\_price,int set\_total\_num) | | Books类有参数的初始化函数 | |
| books::books(string set\_ID, string set\_name, string set\_author, string set\_press, string set\_isbn, int set\_time, int set\_price, int set\_total\_num ,int set\_rent\_num, int set\_rent\_count, int set\_appointment\_num, list<APPOINTMENT> set\_appointment\_list) | | Books类带参数的初始化函数 | |
| string books::get\_name() | | Books类获取图书书名函数 | |
|  | |  | | string books::get\_author() | | Books类获取书的作者函数 | |
|  | |  | | string books::get\_press() | | Books类获取图书出版社函数 | |
|  | |  | | string books::get\_isbn() | | Books类函数获取图书ISBN号 | |
|  | |  | | int books::get\_time() | | Books类获取购入时间函数 | |
|  | |  | | int books::get\_price() | | Books类获取图书价格函数 | |
|  | |  | | int books::get\_total\_num( | | Books类获取全部馆藏数价格 | |
|  | |  | | int books::get\_rent\_num | | Books类获取已借出图书数函数 | |
|  | |  | | int books::get\_in\_num | | Books类获取仍可借图书数函数 | |
|  | |  | | int books::get\_count\_num | | Books类获取借出图书序列函数 | |
|  | |  | | int books::get\_appointment\_num | | Books类获取预约数函数 | |
|  | |  | | list<APPOINTMENT> books::get\_appointment\_list | | Books类获取预约人员序列函数 | |
|  | |  | | int books::get\_rent\_rate() | | Books类获取借出率函数 | |
| User.cpp | | 1. 全部User类成员函数声明 2. 根据各种属性对成员进行查找的函数定义 | | user::user() | | User类无参数构造函数 | |
| user:: user(string set\_ID, string | | User类有参数的初始化函数 | |
| user:: user(string set\_ID, string set\_name, int set\_rent\_number, rent\_map\_type set\_rent\_record, int set\_appointment\_number, list<APPOINTMENT> set\_appointment\_list, int set\_fine,int set\_credit) | | User类带参数的初始化函数 | |
| string user:: get\_ID() | | User类获取使用者ID号函数 | |
|  | |  | | string user:: get\_name() | | User类获取使用者姓名函数 | |
|  | |  | | int user::get\_rent\_num() | | User类获取已借图书数函数 | |
|  | |  | | rent\_map\_type user::get\_rent\_map() | | User类获取使用者已借图书记录的map容器地址函数 | |
|  | |  | | int user::get\_appointment\_num() | | User类获取已预约数函数 | |
|  | |  | | list<APPOINTMENT> user::get\_appointment\_list() | | User类获取已预约图书LIst函数 | |
|  | |  | | int user::get\_priviledge() | | User类获取使用者类别权限函数 | |
|  | |  | | int user::get\_limit\_date() | | User类获取使用者可最长借阅图书天数函数 | |
|  | |  | | int user::get\_limit\_rent\_num() | | User类获取使用者可最大借阅图书本数函数 | |
|  | |  | | int user::get\_limit\_appointment() | | User类获取使用者可最大预约图书本数函数 | |
|  | |  | | int user::get\_fine() | | User类获取使用者罚款金额函数 | |
|  | |  | | int user::get\_credit() | | User类获取使用者违约次数函数 | |
|  | |  | | void user::add\_appointment(string book\_ID) | | User类增加预约图书函数 | |
|  | |  | | void user::delete\_appointment(string book\_ID) | | User类删除预约图书函数 | |
|  | |  | | void user::refresh\_appointment () | | User类刷新预约记录函数 | |
|  | |  | | void user::search\_book() | | User类查询图书函数 | |
|  | |  | | void user::return\_book(string book\_ID) | | User类进行还书操作的函数 | |
|  | |  | | void user::rent\_book(string book\_ID) | | User类进行借书操作的函数 | |
|  | |  | | void user::user\_information() | | User类打印使用者信息函数 | |
|  | |  | | void user::user\_signup( ) | | User类使用者注册页面函数 | |
|  | |  | | void search\_by\_name() | | 查找页面依书名查询函数 | |
|  | |  | | void user\_interface() | | 使用者登陆页面含糊 | |
|  | |  | | void search\_by\_ISBN() | | 使用者依据ISBN号查找函数 | |
|  | |  | | void search\_by\_author() | | 使用者依据图书作者查询函数 | |
|  | |  | | void search\_by\_press() | | 使用者依据图书出版社查询函数 | |
|  | |  | | void search\_by\_ID() | | 使用者依据图书的ID号进行查询函数 | |
|  | |  | | void estab\_user\_index() | | 从文件创建用户容器函数 | |
|  | |  | | void record\_user\_index() | | 从容器 写入文件 函数 | |

4.数据结构说明

Class user

|  |  |
| --- | --- |
| **类属性** | **类属性功能** |
| string user\_name; | 使用者的姓名 |
| int rent\_number; | 借书数量 |
| rent\_map\_type rent\_record; | 借书记录，是一个map容器 |
| int appointment\_number | 预约书籍本数 |
| int fine | 罚款金额 |
| int credit | 诚信记录 每逾期一次增加1 |
| list <APPOINTMENT> appointment\_list | 记录预约的数目。 |
| string ID | 使用者的ID |

Class books

|  |  |
| --- | --- |
| **类属性** | **类属性功能** |
| book\_ID | 书的ID |
| string book\_name | 书名 |
| string book\_author | 作者 |
| string book\_press | 出版社 |
| string book\_isbn | ISBN号 |
| int book\_time | 购入时间 |
| int book\_price | 价格 |
| int total\_num | 书的总数 |
| int rent\_num | 借出去的数量 |
| int rent\_count | 统计借阅总人次数 |
| int appointment\_num | 目前记录的预约人数量 |
| list <APPOINTMENT> appointment\_list | 第一位为预约人的ID 第二位为 预约人可以借书的开始时间。如果借书人还不能借书，就设置为-1 |

5.数据文件

|  |  |  |
| --- | --- | --- |
| **数据文件名** | **使用数据函数** | **用途** |
| today\_date.txt | Main() | 记录时间文件 |
| book\_ID.txt | Main() | 每本书都会有的一个文件，记录借阅情况 |
| user\_record.txt | void estab\_user\_index()  record\_user\_index() | 记录全部用户信息 |
| books\_information.txt | void admin::add\_books() | 图书信息文件 |
| books\_record.txt | void estab\_books\_index()  void record\_books\_index() | 图书借阅记录文件 |

1. 数据字典

|  |  |
| --- | --- |
| **全局变量** | **功能** |
| user\_map\_type user\_map | 使用者map容器 |
| books\_map\_type books\_map | 图书map容器 |
| int today\_date | 当前时间 |
| int opration | 操作类型 |
| DWORD WINAPI time\_move(LPVOID pPararneter) | 时间轴线程 |
| DWORD WINAPI library(LPVOID pPararneter) | 图书馆页面进程 |