

功能分析

账户相关操作

账户的注册

功能：注册账户。

操作：输入一个指令，输入账户类型、用户名、密码，完成注册。

账户的登录

功能：登录账户。

操作：输入一个指令，输入账户用户名和密码，完成登录。

账户的登出

功能：登出账户。

操作：输入一个指令，登出当前账户。

书籍相关操作

书籍的录入与进货

功能：将书籍信息存储到系统中，修改其库存。

当一个账户为店员和店主时可以执行该操作。

操作：输入一个指令，选择该次为第一次录入信息，或是已有录入信息。如果为第一次录入信息，那么在块状链表中新增内容。否则，修改块状链表中对应的进货量。同时，修改对应内容的数据。

书籍的查询

功能：查询满足要求的书籍，显示在屏幕上。

操作：输入一个指令，并且输入待查询书籍的信息。随后返回相应内容。

书籍的购买

功能：购买需要的书籍。

操作：输入一个指令，并且输入欲购买书籍的ISBN号以及购买数量。

日志

功能：将保存的操作内容输出。

上述进行每条指令时，都会存储至另外一个文件中，并且记录经办人。

操作：当账户身份是店长时，可以执行该操作。输入指令和欲查询的内容，则会显示相对应的日志。

用户交互设计

注册

```
Sign up [账户类型] [用户名] [用户密码]
```

如果注册成功，会输出"Sign up successfully!"

如果用户名已经被使用，会输出"Error!The user's name has been used."

登录

```
Login [用户名] [用户密码]
```

如果登录成功，会输出"Welcome![用户名]!"

如果用户名不存在，会输出"Error!The user's name doesn't exist."

如果密码错误，会输出"Sorry!Please check your password."

登出

```
Logout
```

如果登出成功，输出"Goodbye![用户名]!"

如果当前已经没有登录状态的用户，将会输出"Error!There is no account to log out!"

书籍的录入

```
New [ISBN] [书名] [作者] [关键字1],[关键字2],... [数量] [单价] [进货价格]
```

当第一次录入某一本书籍时，使用new指令。

如果添加成功，会输出"Add the book successfully."

如果该书已存在，会输出"Error!The book is existed."

```
Add [ISBN] [数量] [进货价格]
```

非第一次录入书籍时，使用Add指令。

如果添加成功，会输出"Add the book successfully."

如果该书不存在，会输出"Error!The book doesn't exist."

书籍的查询

```
Query [ISBN] [书名] [作者] [关键字1],[关键字2],...
```

部分信息可以不填，默认生成一个"~"表示留空。

查询成功时，会将所有书籍的信息以

```
[ISBN] [书名] [作者] [关键字1],[关键字2],... [库存量] [单价]
```

的格式输出所有满足条件的书籍。

如果不存在这样的书籍，将会输出"Sorry!There is no books match your requirements!"

书籍的购买

```
Buy [ISBN] [购买数]
```

如果购买成功，将会显示"The transaction is success!"

如果购买书籍不存在，将会输出"Error!The book doesn't exist."

如果购买书籍数超过库存数，将会输出"Sorry!There aren't enough books!"

多种日志的生成

```
Log_purchase
```

这样可以生成采购日志。将会按照采购顺序输出采购图书的

```
[ISBN] [书名] [作者] [关键字1],[关键字2],... [数量] [单价] [进货价格]
```

```
Log_transaction
```

这样可以生成交易日志。将会按照交易顺序输出

```
[ISBN] [书名] [作者] [关键字1],[关键字2],... [交易数量] [单价] [进货价格]
```

```
Log_Profit
```

这样可以生成盈利日志。将会输出

"The income is..."

"The cost is..."

"The profit is..."

```
Log_employee [用户名]
```

这样可以生成操作日志。将会按照这名成员的操作顺序生成和上述指令格式一样的日志。

Log_system

将会生成全局日志。将会依次输出执行过的每一条指令。

数据库设计及类设计

```
class Book {
private:
    char ISBN[20];
    char book_name[60];
    char author[60];
    char key_word[60];
    int price;
    int in_price;
    int amount;
    long long block_nxt;
}
```

这是数据将要存储的信息我们将它的ISBN号存在另外多个文件中，存在它的关键字/作者下。随后再找对应的书籍。block_nxt 用于块状链表的存储。我们可以注意到。

```
class Statement {
    int type;
}
```

Statement是一个存储命令的抽象类。所有的类将会在Statement上派生。

```
class Account {
    char user[60];
    char password[60];
    int type;
}
```

Account将会存储账户。

这些信息都将以块状链表存储。从而便于查找。登录的情况将会存在一个栈中，从而提示当前的登录状态。

测试设计
