

# 《数据结构课程设计》

## 各模块设计说明

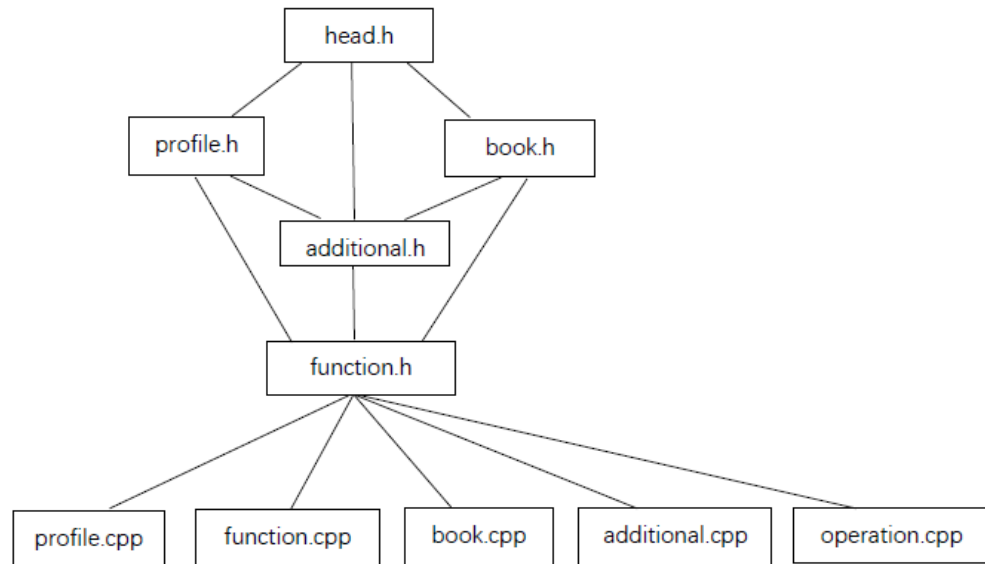
实验名称     图书馆管理系统

班     级     2015211314

姓     名     罗暄澍、郝燕如、郭子宁

## 一、 总体结构和功能模块划分：

### 1. 总体结构：



### 2. 功能模块划分：

- 1) 模块名称：profile：  
模块功能简要概述：定义老师、学生的类并实现其接口及使用；
- 2) 模块名称：book：  
模块功能简要概述：定义书的类并实现其接口及使用；
- 3) 模块名称：additional：  
模块功能简要概述：定义时间、借书单的类并实现其接口及使用；
- 4) 模块名称：function：  
模块功能简要概述：实现登录、查询、预约、借书、还书、添加书目、修改图书信息的功能
- 5) 模块名称：operation：  
模块功能简要概述：实现管理员、老师、学生登录的不同情况及其权限；

## 二、 各模块设计说明：

### 1. 模块名称：profile：

定义老师、学生的类并实现其接口及使用；

```
class User {  
private:  
    string name;           //姓名  
    string ID;             //工作证号  
    string academy;        //学院  
    int authority;          //权限为 0--老师， 权限为 1--学生  
    int lent_number;        //已借书数目
```

```

        double money_delay; //已欠书款
public:
    User();
    ~User();
    User(string name_, string ID_, string academy_, int authority_, int
lent_number, double money_delay);
    string get_name() { return name; }
    string get_ID() { return ID; }
    string get_academy() { return academy; }
    int get_authority() { return authority; }
    int get_lent_number() { return lent_number; }
    double get_money_delay() { return money_delay; }

    void set_lent_number(int n) { lent_number = n; }
    void set_money_delay(double n) { money_delay = n; }
};

User::User()
{
}
User::~~User()
{
}
User::User(string name_, string ID_, string academy_, int authority_, int
lent_number_,double money_delay_)
{
    name = name_;
    ID = ID_;
    academy = academy_;
    authority = authority_;
    lent_number = lent_number_;
    money_delay = money_delay_;
}

```

## 2. 模块名称 : book :

定义书的类并实现其接口及使用 ;

```

class Book {
private:
    string ID;           //图书编码, 要自动生成
    string ISBN;         //ISBN
    string name;          //书名
    string author;        //作者
    string publisher;     //出版社
    string introduction;  //简介

```

```

        string bought_time;    //购入时间
        int origin_num;        //最初的数量
        int remain_num;        //数量
    public:
        Book();
        ~Book();
        Book(string ID_, string ISBN_, string name_, string author_, string publisher_,
string introduction_, string bought_time_, int origin_num_, int remain_num_);
        string get_ID() { return ID; }
        string get_ISBN() { return ISBN; }
        string get_name() { return name; }
        string get_author() { return author; }
        string get_publisher() { return publisher; }
        string get_introduction() { return introduction; }
        string get_bought_time() { return bought_time; }
        int get_origin_num() { return origin_num; }
        int get_remain_num() { return remain_num; }

        bool isForeign();
        void ShowAllInfo();

        void set_remain_num(int n) { remain_num = n; }
};

Book::Book()
{
}

Book::~Book()
{
}

Book::Book(string ID_, string ISBN_, string name_, string author_, string
publisher_, string introduction_, string bought_time_, int origin_num_, int
remain_num_)
{
    ID = ID_;
    ISBN = ISBN_;
    name = name_;
    author = author_;
    publisher = publisher_;
    introduction = introduction_;
    bought_time = bought_time_;
    origin_num = origin_num_;
    remain_num = remain_num_;
}

```

```

    }

    bool Book::isForeign()
    {
        string a = author;
        if (a[0] == '(' || a[0] == '[')
            return true;
        return false;
    }

    void Book::ShowAllInfo()
    {
        double percent = (double)(origin_num - remain_num) / origin_num;
        cout << "书名：" << name << "\tISBN：" << ISBN << "\t作者：" << author
        << "\t出版社：" << publisher << "\t介绍：" << introduction << "\t购买时间："
        << bought_time << "\t 剩余本数：" << remain_num << "\t 借阅率：" <<
        setiosflags(ios::fixed) << setprecision(2) << percent * 100 << "%" << endl;
    }

```

### 3. 模块名称 : additional :

定义时间、借书单类并实现其接口及使用；

```

class Time {
private:
    int year = 0;
    int month = 0;
    int day = 0;
public:
    Time();
    ~Time();
    Time(int year_, int month_, int day_);
    Time(int n);

    int get_year() { return year; }
    int get_month() { return month; }
    int get_day() { return day; }

    void set_time(int year_, int month_, int day_);

    int get_absdays(Time a);
    int get_minus(Time a, Time b); // b-a
    bool compare(Time a, Time b); // a 是否大于 b
    bool is_leap();
    void tomorrow();
    void show();

```

```

};

class additional { //借书单
private:
    string ID;
    int borrowed_num = 0;
    vector<Time> borrow;
    vector<string> borrowed_ISBN;
    int wanted_num = 0;
    vector<string> wanted_ISBN;
public:
    additional();
    ~additional();
    additional(string ID_, int borrowed_num_, vector<Time> borrow_,
vector<string> borrowed_ISBN_, int wanted_num_, vector<string> wanted_ISBN_);

    string get_ID() { return ID; }
    int get_borrowed_num() { return borrowed_num; }
    vector<Time> get_borrow_time() { return borrow; }
    vector<string> get_borrowed_ISBN() { return borrowed_ISBN; }
    int get_wanted_num() { return wanted_num; }
    vector<string> get_wanted_ISBN() { return wanted_ISBN; }

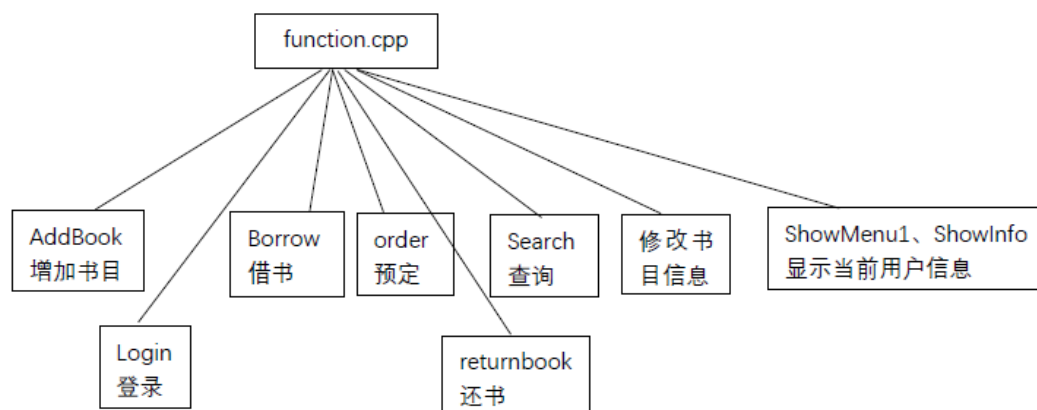
    void set_borrowed_num(int n) { borrowed_num = n; }
    void set_borrow_time(vector<Time> n) { borrow = n; }
    void set_borrowed_ISBN(vector<string> n) { borrowed_ISBN = n; }
    void set_wanted_num(int n) { wanted_num = n; }
    void set_wanted_ISBN(vector<string> n) { wanted_ISBN = n; }

};

```

#### 4. 模块名称 : function :

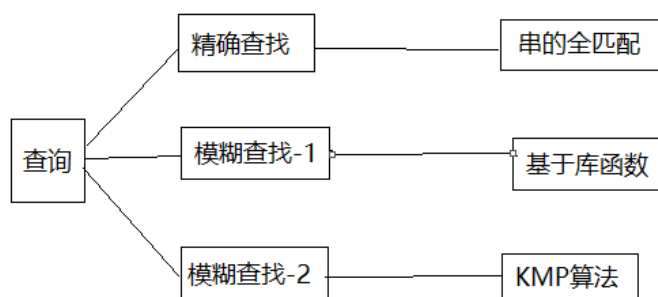
实现登录、显示个人信息、查询、预约、借书、还书、添加书目、修改图书信息的功能



与其他模块调用关系：

功能	函数	参数	调用模块	调用模块功能
登录系统	int login()	vector<User> user	profile	老师、学生的类及接口使用
显示用户信息	void ShowInfo()	User user,	profile	老师、学生的类及接口使用
		vector<Book> book	book	书的及其接口使用
		additional addinfo	additional	时间、借书单的类及其接口使用
增加书目	void AddBook ()	vector<Book> &book	book	书的及其接口使用
查询	vector<int> Search	vector<Book> book	book	书的及其接口使用
借书	void Borrow ()	Book &book	book	书的及其接口使用
		User &user	profile	老师、学生的类及接口使用
		additional &addinfo	additional	时间、借书单的类及其接口使用
预约	void order ()	additional &addinfo	additional	时间、借书单的类及其接口使用
		Book book	book	书的及其接口使用
还书	void returnbook	Book &book	book	书的及其接口使用
		User &user	profile	老师、学生的类及接口使用
		additional &addinfo	additional	时间、借书单的类及其接口使用

函数使用算法：



## 5. 模块名称：operation：

实现管理员、老师、学生登录的不同情况及其权限；

与其他模块调用关系：

功能	函数	参数	调用模块	调用模块功能
实现管理员、老师、学生登录的不同情况及其权限	void operation ()	User user,	profile	老师、学生的类及接口使用
		vector<Book> book	book	书的及其接口使用
		additional addinfo	additional	时间、借书单的类及其接口使用