

 未标题-1

**《面向对象编程》**

**课程设计报告**

**题 目 ：**

**专 业 班 级 ：**

**学 生 姓 名 ：**

**学 号 ：**

**指 导 教 师 ：**

**课程设计时间：**

1 需求分析

核心需求：实现教室管理系统，完成对学校教室的维护、查询、预约、统计功能。程序拥有权限控制，分为管理员和普通用户。

管理员需求：查看某栋楼教室的概要信息、每个教室的详细信息（包括教室基本信息、预约情况、上课情况）、修改某教室基本信息、增加教室、删除教室、增加某教室上课情况、统计某类教室的综合使用情况（周上课率、周预约率）。

普通用户需求：查询符合条件的空闲教室、预约空闲教室。

其他需求：程序具有良好的交互性。

2 概要设计

程序运用面向对象编程思想进行设计。核心有教室类、预约信息类、上课信息类。

教室类：

成员：校区号、教学楼号、教室号、教室类型、座位数、预约状态、预约数、预约信息、上课数、上课信息。

函数：默认构造函数、成员获取函数、成员设置函数。

预约信息类：

成员：课时时间、学号/教工号、姓名、电话。

函数：默认构造函数、构造函数、成员获取函数、重载流运算符。

上课信息类：

成员：教师姓名、教工号、总上课次数、上课时间。

函数：默认构造函数、构造函数、成员获取函数、重载流运算符。

课时时间类：

成员：课时周、周、节

函数：重载相等、不等、流运算符

程序结构图如图1所示。

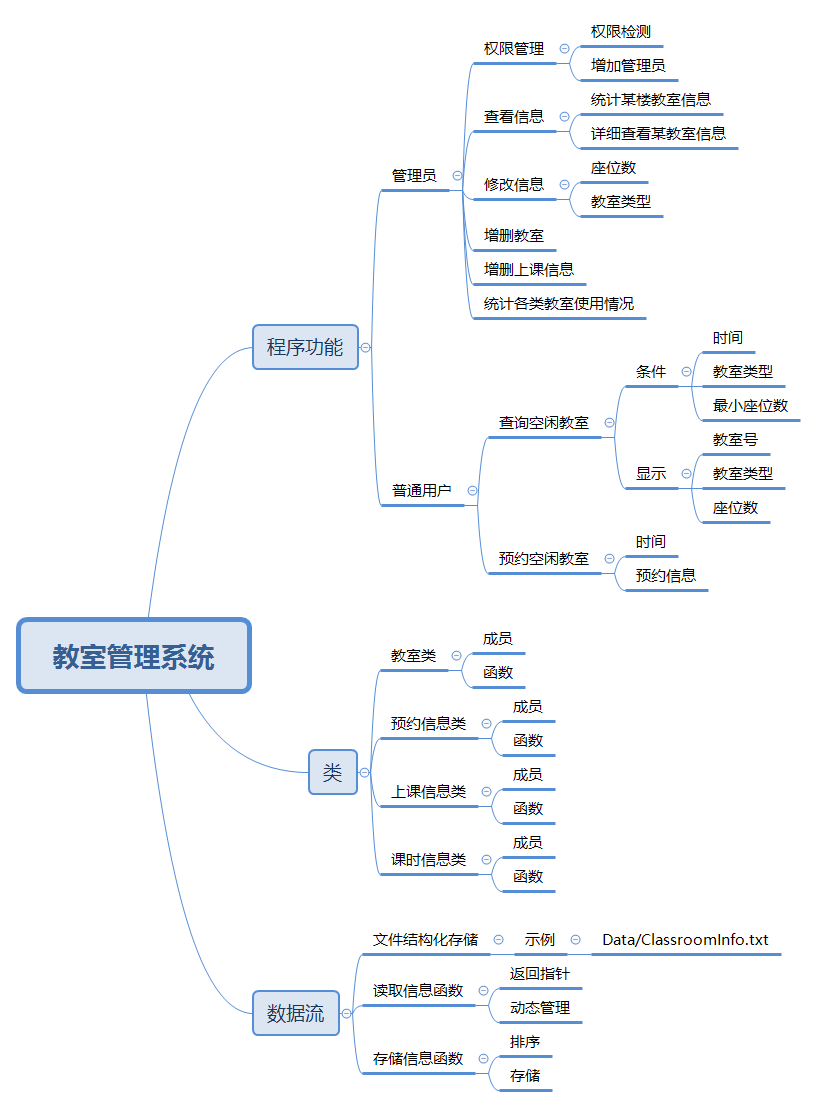


图1 程序结构图

3 运行环境

硬件环境

处理器：Intel® Core™ i7-6500U CPU @ 2.50GHz 2.59GHz

内存：8.00GB

系统类型：64位操作系统，基于x64的处理器

软件环境：

Windows 10

4 开发工具和编程语言

开发工具：

Microsoft Visual Studio Community 2017

编程语言：

C++11

5 详细设计

教室信息类：

该类封装有教室的基本属性和方法，用于存储每个教室的信息集合及修改其中的信息，构成教室管理系统的核心。

class Classroom

{

public:

Classroom() = default; //默认构造函数

void setCampusType(uint); //设置函数

void setBuildID(uint);

void setRoomID(uint);

void setRoomType(uint);

void setSeats(uint);

void setOrderStatus(uint);

void setOrderCnt(uint);

void setClassCnt(uint);

void addOrderInfo(OrderInfo);

void addClassInfo(ClassInfo);

uint getCampusType() const; //获取函数

uint getBuildID() const;

uint getRoomID() const;

uint getRoomType() const;

uint getSeats() const;

uint getOrderStatus() const;

uint getOrderCnt() const;

uint getClassCnt() const;

vector <OrderInfo> getOrderInfo() const;

vector <ClassInfo> getClassInfo() const;

private:

uint m\_campusType = 0; //校区

uint m\_buildID = 0; //楼号

uint m\_roomID = 0; //教室号

uint m\_roomType = 0; //教室类型

uint m\_seats = 0; //座位数

uint m\_orderStatus = 0; //预约状态

uint m\_orderCnt = 0; //预约数

uint m\_classCnt = 0; //课堂数

vector <OrderInfo> m\_orderInfo; //预约信息数组

vector <ClassInfo> m\_classInfo; //课堂信息数组

};

预约信息类：

主要用来存取每间教室的一条预约信息，含有预约时间、预约账号、预约者姓名及电话等基本信息，提供构造函数进行初始化，提供获取函数用于获取私有成员。

class OrderInfo

{

friend ostream& operator<< (ostream &, const OrderInfo&);

public:

OrderInfo() = default;

OrderInfo(const classTime &time, const string &ID,

const string &name, const string &phoneNum) : m\_time(time),

m\_ID(ID), m\_name(name), m\_phoneNum(phoneNum) { }

string getID() const;

classTime getClassTime() const;

string getName() const;

string getPhoneNum() const;

private:

classTime m\_time;

string m\_ID = "";

string m\_name = "";

string m\_phoneNum = "";

};

上课信息类：

用来存取当前教室的一条上课信息，如上课时间、上课老师姓名、ID。

class ClassInfo

{

friend ostream& operator<< (ostream &, const ClassInfo &);

public:

ClassInfo() = default;

ClassInfo(const string &teacherName, const string &teacherID, uint sumClassCnt, vector< classTime > &classTime)

:m\_teacherName(teacherName),m\_teacherID(teacherID),

m\_sumClassCnt(sumClassCnt), m\_classTime(classTime) {}

string getName() const;

string getID() const;

uint getSumClassCnt() const;

vector<classTime> getClassTime() const;

private:

string m\_teacherName = "";

string m\_teacherID = "";

uint m\_sumClassCnt = 0;

vector< classTime > m\_classTime; //三元组 （课时周，周，节）

};

查看信息：

查看每座楼的所有教室基本信息，如教室号、教室类型、座位数、预约数、课程数。查看每个教室的详细信息，包括基本信息、预约信息、上课信息。

修改信息：

对某个教室的信息进行修改。

修改教室：

可增加、删除一个教室。

增加管理员：

增加一个管理员账号并存储。

增加上课信息：

为某个教室增加上课信息。

统计使用情况：

用于统计各类教室的使用率情况。

查询空闲教室：

普通用户可通过输入课时时间来查询某号楼所有的空闲教室，同时可按照座位数、教室类型进行筛选。

预约空闲教室：

普通用户可通过输入课时时间来预约教室、如果预约时间该教室有课或已有预约将预约失败。

6 调试分析

问题1：程序所有功能都集中一起。

解决：增加用户权限，分为管理员和普通用户。管理员需要密码登陆，并对其进行校验，可对教室信息进行直接管理维护。对于普通用户仅开放预约及查询教室功能。

问题2：预约无校验功能。

解决：为系统添加时间属性、采用课时时间的方式对预约信息、上课信息进行时间管理。预约时可检测是否当前时间已被上课或预约占用。

问题3：交互逻辑使用不便。

解决：改进交互逻辑，提供跳转功能、选择错误处理。

程序改进：

1. 改进管理员对教室进行维护的操作逻辑，使得操作更加方便。
2. 增加真实时间、自动对已过期的预约进行删除。
3. 采用更为合理的信息管理方式、使教室信息管理更加高效易行。
4. 增加用户更多功能，如查询自己的预约、删除自己的预约。
5. 增加程序鲁棒性。
6. 改进交互。

7 测试结果

管理员功能列表如下图2。



图2 管理员功能列表

查询教室详细信息如下图3。

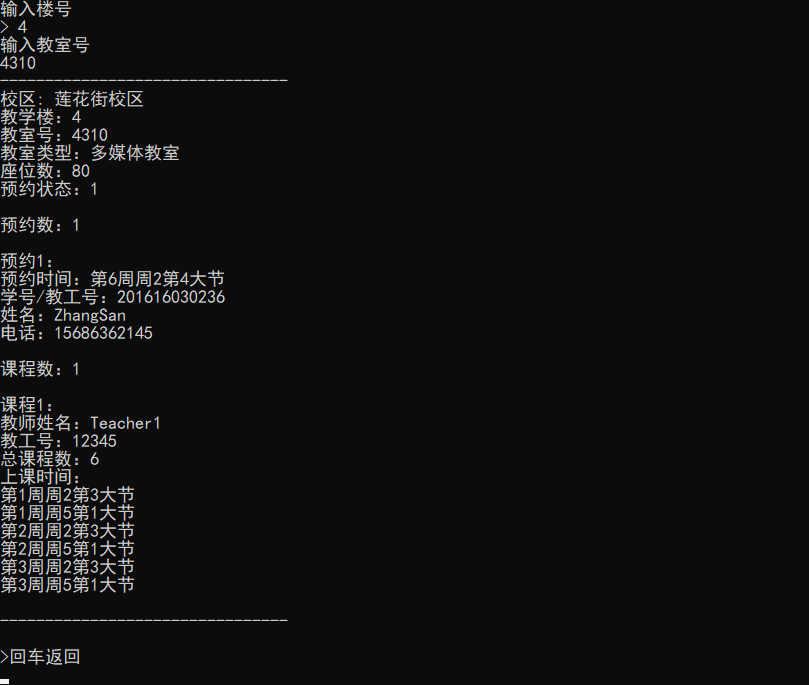


图3 查询教室详细信息

普通用户界面如下图4。



图4 普通用户界面

查询空闲教室如下图5。

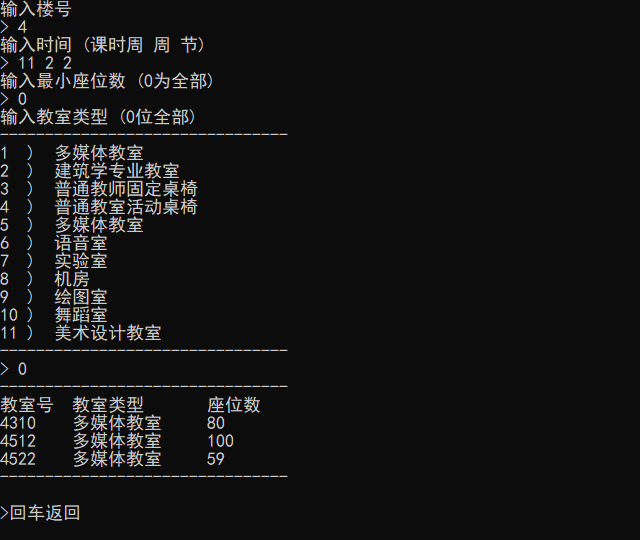


图5 查询空闲教室

预约空闲教室（预约成功）如下图6。

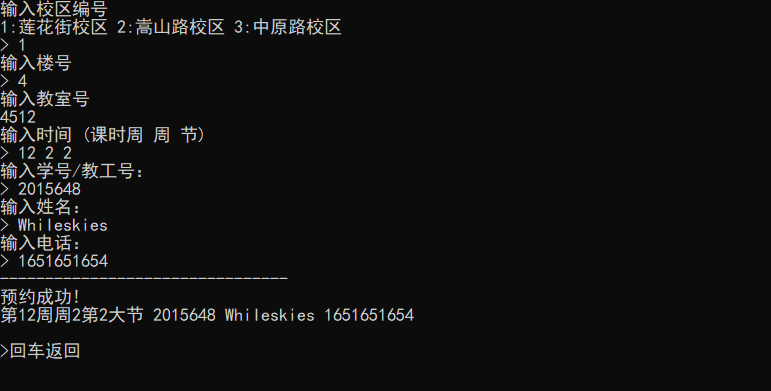


图6 预约空闲教室（成功）

预约空闲教室（预约失败）如下图7。

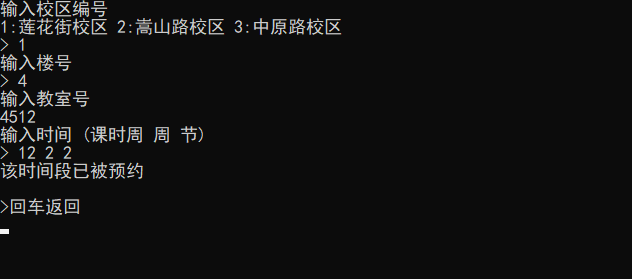


图7 预约空闲教室（失败）

参考文献

[1] H.M.Deitel.C++大学基础教程（第五版）[M]. 北京: 电子工业出版社, 2011.

[2] 沈学东等. C++面向对象程序设计实用教程[M]. 上海: 上海交通大学出版社,2012.

[3] 谭浩强等. C++程序设计 [M]. 北京: 清华大学出版社, 2011.

[4] Stanley B.Lippman等. C++ Primer中文版[M]. 北京：电子工业出版,2017.

[5] Bjarne Stroustrup. C++程序设计语言[M]. 北京：机械工业出版社,2016.

**心得体会**

本次的OOP课程设计在19周，赶上各种考试,在时间上很是紧迫。初看题目时想了好久，由于这是一个稍大的系统，不仅要从面向对象思想上去思考代码实现，更要联系现实从而解决现实问题。刚开始时并不简单，要设计程序的类、框架，有时没想清楚就去敲代码最后不得不重写。现实问题总是复杂的，要考虑到方方面面，当然也不可能设计一个完美的程序，只能尽可能的考虑到多的方面。对比之前做过的C语言面向过程的编程方法，面向对象提供了一种全新的视角去看待问题——去更多关注解决问题的方法，而不是关注一步一步的过程怎么做。对于这种稍大点的程序，可极大提高程序的编码效率和简化自己的思维过程。通过这次课程实践，我对面向对象思想有了更进一步的认识，认识到了如何将面向对象编程思想运用到实际的开发中。也通过这次实践提高了自己的编码能力，认识到只有将理论和实践结合起来我们才能真正学好编程。