深圳大学考试答题纸

(以论文、报告等形式考核专用)

二〇 一九 ～二〇 二零 学年度第 二 学期

课程编号	1500220004R	课序号	01	课程名称	面向对象程序设计(荣誉)	主讲教师	张艳	评分	

学号	2019092121	姓名	沈晨珀	专业年级	2019 级计算机科学与技术 04

	教师评语：

题目： 基于 STL 实现机房预约系统

一、 题目描述

本课程的教学目的是掌握并熟练使用 STL。我的期末大作业是基于 STL 实现机房预约系统。

该系统主要用于实现机房用户（学生/老师/管理员）的帐号注册/登录/特殊功能。

学生：
1. 申请预约
2. 查看已申请预约
3. 查看机房使用情况
4. 取消预约
5. 注销登录

老师：
1. 查看机房使用情况
2. 审核预约
3. 注销登录

管理员：
1. 注册账号
2. 查看账号
3. 查看机房信息
4. 清空预约
5. 新增机房
6. 注销登录

二、 开发环境

Visual Studio 2019

三、 详细设计

1. 程序结构

a) 程序结构

i. 主程序

```
int main() {
    while (true) {
        system("cls");
        Menu1();
        int opt;
        cin >> opt;
        switch (opt) {
        case 1:StudentFunction(); break;
        case 2:TeacherFunction(); break;
        case 3:AdminFunction(); break;
        case 0:ExitProgram(); break;
        default:cout << "输入错误，请重新选择！" << endl; system("pause"); break;
        }
    }
    return 0;
}

void Menu1() {
    cout << "---------------------------------------------" << endl;
    cout << "          机房预约系统登陆界面               " << endl;
    cout << "---------------------------------------------" << endl;
    cout << "                                             " << endl;
    cout << "              1. 学    生                    " << endl;
    cout << "                                             " << endl;
    cout << "              2. 老    师                    " << endl;
    cout << "                                             " << endl;
    cout << "              3. 管 理 员                    " << endl;
    cout << "                                             " << endl;
    cout << "              0. 退    出                    " << endl;
    cout << "                                             " << endl;
    cout << "---------------------------------------------" << endl;
    cout << "请输入你的选择:" << endl;
}
//退出程序
void ExitProgram() {
    cout << "欢迎下次使用！" << endl;
    exit(-1);
}
```

ii. 用户功能界面菜单（以学生为例，老师/管理员类似）

思路：验证帐号—>创建对象—>根据输入选择功能

```
//学生功能界面
void StudentFunction() {
    Identidy* p;
    //登录学生账号并验证
    if (Login(1, p)) {
        //登陆成功
        Student* obj = (Student*)p;
        while (true) {
            system("cls");
            obj->Menu();
            int opt;
            cin >> opt;
            //申请预约
            if (opt == 1) {
                obj->apply();
            }
            //查看已申请预约
            else if (opt == 2) {
                obj->showMyApply();
            }
            //查看机房使用情况
            else if (opt == 3) {
                obj->showAllApply();
            }
            //取消预约
            else if (opt == 4) {
                obj->cancelApply();
            }
            //注销账号
            else if (opt == 5) {
                delete obj;
                cout << "注销成功" << endl;
                system("pause");
                system("cls");
                return;
            }
            else {
                cout << "输入错误，请重新选择！" << endl;
                system("pause");
                system("cls");
            }
        }
    }
    else {
        return;
```

b) 类

i. 身份基类

```
//身份基类
class Identidy {
public:
    //用户名
    string Name;
    //密码
    string Password;
    virtual void Menu() = 0;
};
```

1. 学生类

```
class Student :public Identidy {
public:
    //学号
    int studentID;
    //默认构造
    Student();
    //有参构造 学号/姓名/密码
    Student(int id, string name, string password);
    //菜单界面
    void Menu();
    //申请预约
    void apply();
    //查看已申请预约
    void showMyApply();
    //查看机房使用情况
    void showAllApply();
    //取消预约
    void cancelApply();
    //机房信息
    vector<ComputerRoom> vecComputerRoom;
};
```

2. 老师类

```
class Teacher :public Identidy {
public:
    //职工号
    int teacherID;
    //默认构造
    Teacher();
    //有参构造 学号/姓名/密码
    Teacher(int id, string name, string password);
    //菜单界面
    void Menu();
    //查看机房使用情况
    void showAllApply();
    //审核预约
    void checkApply();
};
```

3. 管理员类

```
class Admin :public Identidy {
public:
    //学号
    string studentID;
    //默认构造
    Admin();
    //有参构造 学号/姓名/密码
    Admin(string name, string password);
    //菜单界面
    void Menu();
    //注册帐号
    void Register();
    //查看帐号
    void showAccount();
    //查看机房信息
    void showInfo();
    //清空预约信息
    void clear();
    //新增机房
    void addComputerRoom();
};
```

ii. 机房类

```
class ComputerRoom {
public:
    //机房名字
    string RoomName;
    //房间号
    int RoomNum;
    //机房数量
    int ComputerNum;
    //开放起始时间
    int StartTime;
    //开放结束时间
    int EndTime;
    //默认构造
    ComputerRoom() {
        RoomName = "";
        RoomNum = 0;
        ComputerNum = 0;
        StartTime = 0;
        EndTime = 0;
    }
    //有参构造
    ComputerRoom(string s, int a, int b, int c, int d) {
        RoomName = s;
        RoomNum = a;
        ComputerNum = b;
        StartTime = c;
        EndTime = d;
    }
};
```

2. 主要功能（对实现的系统功能进行阐述，哪个函数完成什么功能）

a) 主程序

i. 帐号登录验证

思路：打开对应的数据文件，依次读取帐号数个信息（管理员：用户名+密码，学生：学号+用户名+密码，老师：职工号+用户名+密码），与用户输入的信息进行对比，知道有一组帐号信息完全匹配，否则不存在用户输入的帐号（登陆失败）

```
//登陆界面
bool Login(int type, Identidy*& p) {
    int id = 0;
    string name, password;
    switch (type)
    {
        //学生
    case 1:cout << "请输入学号:" << endl; cin >> id; break;
```

```
        //老师
    case 2:cout << "请输入职工号:" << endl; cin >> id; break;
    }
    cout << "请输入用户名:" << endl;
    cin >> name;
    cout << "请输入密码:" << endl;
    cin >> password;
    //验证学号/职工号，用户名，密码是否匹配
    switch (type)
    {
        //学生
    case 1:{
        if (checkLogin(1, id, name, password)) {
            cout << "学生账号登陆成功！" << endl;
            system("pause");
            system("cls");
            p = new Student(id, name, password);
            return true;
        }
        else {
            cout << "学生账号登陆失败" << endl;
            system("pause");
            return false;
        }
    }
        //老师
    case 2:{
        if (checkLogin(2, id, name, password)) {
            cout << "老师账号登陆成功！" << endl;
            system("pause");
            system("cls");
            p = new Teacher(id, name, password);
            return true;
        }
        else {
            cout << "老师账号登陆失败" << endl;
            system("pause");
            return false;
        }
    }
        //管理员
    case 3:{
        if (checkLogin(3, name, password)) {
            cout << "管理员账号登陆成功！" << endl;
            system("pause");
            system("cls");
            p = new Admin(name, password);
            return true;
```

```
        else {
            cout << "管理员账号登陆失败" << endl;
            system("pause");
            return false;
        }
    }
    }
    return false;
}
```

教师/学生账号登陆需要三个信息，管理员账号登录需要两个信息。所以用函数重载来实现。

```
//根据帐号信息，验证登录(学生/老师)
bool checkLogin(int type, int id, string name, string password) {
    ifstream ifs;
    int fid;
    string fname, fpassword;
    //学生验证
    if (type == 1) {
        ifs.open(StudentFile, ios::in);
        if (!ifs.is_open()) {
            cout << "文件不存在" << endl;
            ifs.close();
            system("pause");
            return false;
        }
        while (ifs >> fid >> fname >> fpassword) {
            if (id == fid && name == fname && password == fpassword) {
                return true;
            }
        }
        return false;
    }
    //老师验证
    else if (type == 2) {
        ifs.open(TeacherFile, ios::in);
        if (!ifs.is_open()) {
            cout << "文件不存在" << endl;
            ifs.close();
            system("pause");
            return false;
        }
        while (ifs >> fid >> fname >> fpassword) {
            if (id == fid && name == fname && password == fpassword) {
                return true;
            }
        }
        return false;
```

```
//根据帐号信息，验证登录(管理员)
bool checkLogin(int type, string name, string password) {
    ifstream ifs;
    string fname, fpassword;
    ifs.open(AdminFile, ios::in);
    if (!ifs.is_open()) {
        cout << "帐号文件不存在" << endl;
        ifs.close();
        system("pause");
        return false;
    }
    while (ifs >> fname >> fpassword) {
        if (name == fname && password == fpassword) {
            return true;
        }
    }
    return false;
}
```

b) 管理员类实现

i. 辅助函数（具体用途见方法实现代码注释）

1. 容器初始化

```
//初始化已注册学生老师所有信息容器
void setVecAllData(vector<Student>& v1, vector<Teacher>& v2) {
    v1.clear();
    v2.clear();
    string fileName;
    int id;
    string name;
    string password;
    ifstream ifs;
    ifstream ifs1;
    fileName = StudentFile;
    //打开学生信息文件
    ifs.open(fileName);
    while (ifs >> id >> name >> password)
        v1.push_back(Student(id, name, password));
    ifs.close();
    //打开老师信息文件
    fileName = TeacherFile;
    ifs1.open(fileName);
    while (ifs1 >> id >> name >> password)
        v2.push_back(Teacher(id, name, password));
    return;
}
```

```
//初始化学号/职工号容器内容
void setVecIdData(vector<int>& v1, vector<int>& v2) {
    v1.clear();
    v2.clear();
    string fileName;
    int id;
    string name;
    string password;
    ifstream ifs;
    //打开学生信息文件
    fileName = StudentFile;
    ifs.open(fileName);
    while (ifs >> id >> name >> password)
        v1.push_back(id);
    ifs.close();
    ifstream ifs2;
    fileName = TeacherFile;
    //打开老师信息文件
    ifs2.open(fileName);
    while (ifs2 >> id >> name >> password)
        v2.push_back(id);
    ifs2.close();
    return;
}
```

```
//初始化已注册学生老师用户名容器
void setVecNameData(vector<string>& v1, vector<string>& v2) {
    v1.clear();
    v2.clear();
    string fileName;
    int id;
    string name;
    string password;
    ifstream ifs;
    ifstream ifs2;
    fileName = StudentFile;
    //打开学生信息文件
    ifs.open(fileName);
    while (ifs >> id >> name >> password)
        v1.push_back(name);
    ifs.close();
    fileName = TeacherFile;
    //打开老师信息文件
    ifs2.open(fileName);
    while (ifs2 >> id >> name >> password)
        v2.push_back(name);
    ifs2.close();
    return;
```

```
//初始化已开放机房房间号容器
void setVecRoomNum(vector<int>& v1) {
    ifstream ifs;
    string str1, str;
    //打开机房信息文件
    ifs.open(ComputerRoomFile);
    while (getline(ifs, str1)) {
        //截取房间号信息
        str = str1.substr(str1.find("楼") + 2, str1.find('\t') - str1.find("楼"));
        v1.push_back(stoi(str));
    }
    return;
}
```

2. 重复检查

```
//检查重复，重复返回 true，无重复返回 false
//检查 ID 是否重复
bool checkRepeat(int newID, vector<int>& v1) {
    if (find(begin(v1), end(v1), newID) != end(v1))
        return true;
    else
        return false;
}

//重载 检查用户名重复
bool checkRepeat(string newName, vector<string>& v1) {
    if (find(begin(v1), end(v1), newName) != end(v1))
        return true;
    else
        return false;
}
```

ii. 类方法实现

1. 注册账号（根据提示输入对应信息，并将账号信息存入数据文档）

```
//添加账号
void Admin::Register()
{
    //已注册学号/职工号容器，用于检测是否有重复注册
    vector<int> ExistedStudentID;
    vector<int> ExistedTeacherID;
    vector<string> ExistedStudentName;
    vector<string> ExistedTeacherName;
    //初始化容器内容
    setVecIdData(ExistedStudentID, ExistedTeacherID);
    setVecNameData(ExistedStudentName, ExistedTeacherName);
    system("cls");
    this->Menu();
```

```
int opt;
    string fileName;
    ofstream ofs;
    int id;
    string name, password;
    while (true) {
        cout << "请输入注册账号类型:" << endl;
        cout << "1: 学生账号" << endl;
        cout << "2: 老师账号" << endl;
        cin >> opt;
        //注册学生账号
        if (opt == 1) {
            fileName = StudentFile;
            cout << "请输入学号:" << endl;
            cin >> id;
            //检查 ID 是否重复注册
            if (checkRepeat(id, ExistedStudentID)) {
                cout << "该学号已经存在。" << endl << "注册失败" << endl;
                system("pause");
                return;
            }
            break;
        }
        else if (opt == 2) {
            fileName = TeacherFile;
            cout << "请输入职工号:" << endl;
            cin >> id;
            //检查 ID 是否重复注册
            if (checkRepeat(id, ExistedTeacherID)) {
                cout << "该职工号已经存在。" << endl << "注册失败" << endl;
                system("pause");
                return;
            }
            break;
        }
        else {
            cout << "输入错误，请重新选择！" << endl;
            system("pause");
            system("cls");
            this->Menu();
        }
    }
    //打开对应文件
    ofs.open(fileName, ios::out | ios::app);
    if (!ofs.is_open()) {
        cout << "文件不存在" << endl;
        ofs.close();
```

```
            system("pause");
            return;
        }
        cout << "请输入用户名:" << endl;
        cin >> name;
        //检查用户名是否重复
        if (opt == 1) {
            if (checkRepeat(name, ExistedStudentName)) {
                cout << "该用户名已经存在。" << endl << "注册失败" << endl;
                system("pause");
                return;
            }
        }
        else if (opt == 2) {
            if (checkRepeat(name, ExistedTeacherName)) {
                cout << "该用户名已经存在。" << endl << "注册失败" << endl;
                system("pause");
                return;
            }
        }
        cout << "请输入密码:" << endl;
        cin >> password;
        //添加信息
        ofs << id << " " << name << " " << password << endl;
        cout << "账号注册成功！" << endl;
        system("pause");
        system("cls");
        ofs.close();
        return;
}
```

2. 查看账号（根据用户选择，打开对应数据文档，并展示已注册帐号信息）

```
//查看账号
void Admin::showAccount()
{
    //已注册学生老师所有信息容器，用于遍历
    vector<Student> ExistedStudentInfo;
    vector<Teacher> ExistedTeacherInfo;
    //初始化容器内容
    setVecAllData(ExistedStudentInfo, ExistedTeacherInfo);
    string name, password;
    int opt;
    while (true) {
        cout << "请输入需要查看的账号类型:" << endl;
        cout << "1: 学生账号" << endl;
        cout << "2: 老师账号" << endl;
        cin >> opt;
```

```
        //注册学生账号
        if (opt == 1) {
            cout << "学生帐号信息如下:" << endl;
            //打印学生信息
            for_each(begin(ExistedStudentInfo), end(ExistedStudentInfo), [](Student& s) {
                cout << "学号: " << setiosflags(ios::left) << setw(15) << s.studentID << "
用户名: " << setw(15) << s.Name << "密码: " << setw(15) << s.Password << endl;
                });
            break;
        }
        else if (opt == 2) {
            cout << "老师账号信息如下:" << endl;
            //打印老师信息
            for_each(begin(ExistedTeacherInfo), end(ExistedTeacherInfo), [](Teacher& t) {
                cout << "职工号: " << setiosflags(ios::left) << setw(15) << t.teacherID << "
用户名: " << setw(15) << t.Name << "密码: " << setw(15) << t.Password << endl;
                });
            break;
        }
        else {
            cout << "输入错误，请重新选择！" << endl;
            system("pause");
            system("cls");
            this->Menu();
        }
    }
    system("pause");
    return;
}
```

3. 查看机房信息（打开机房信息文件，输出所有信息）

```
//查看机房
void Admin::showInfo()
{
    ifstream ifs;
    string str;
    //打开机房信息文件
    ifs.open(ComputerRoomFile);
    if (!ifs.is_open()) {
        cout << "文件不存在" << endl;
        ifs.close();
        system("pause");
        return;
    }
    cout << "机房信息如下:" << endl;
    //打印机房信息
    while (getline(ifs, str))
```

```
        cout << str << endl;
    system("pause");
    return;
}
```

4. 清空预约（清空预约信息汇总中的所有信息）

```
//清空预约
void Admin::clear()
{
    cout << "是否确认清空预约信息？" << endl;
    cout << "1: 确认" << endl;
    cout << "2: 返回" << endl;
    int opt;
    cin >> opt;
    if (opt == 1) {
        //覆盖原文件（清空）
        ofstream ofs(OrderStaticFile, ios::trunc);
        ofs.close();
        cout << "清空成功！" << endl;
    }
    system("pause");
    return;
}
```

5. 新增机房（根据提示输入信息，并将机房信息存入数据文档）

```
//新增机房
void Admin::addComputerRoom()
{
    vector<int> RoomNumVec;
    setVecRoomNum(RoomNumVec);
    int RoomNum, ComputerNum, startTime, endTime;
    cout << "请输入机房房间号码:" << endl;
    cin >> RoomNum;
    //检查房间号是否重复
    if (checkRepeat(RoomNum, RoomNumVec)) {
        cout << "已经存在该机房信息。添加失败" << endl;
        system("pause");
        return;
    }
    cout << "请输入电脑数量:" << endl;
    cin >> ComputerNum;
    cout << "请输入开放时间:" << endl;
    cout << "开始时间（第 xx 节课）:";
    cin >> startTime;
    cout << "结束时间（第 xx 节课）:";
    cin >> endTime;
```

```
    if (startTime < 1 || endTime>12 || endTime < startTime) {
        cout << "时间输入错误。添加失败" << endl;
        system("pause");
        return;
    }
    ofstream ofs;
    //打开新房信息文件
    ofs.open(ComputerRoomFile, ios::out | ios::app);
    //接在后面添加信息
    ofs << "机房名称：南区计算机大楼" << RoomNum << '\t' << "电脑数量：" << ComputerNum << '\t'
<< "开放时间：" << setw(2) << setfill('0') << startTime << '-' << setw(2) << setfill('0') << endTime
<< "节课" << endl;
    cout << "添加成功！" << endl;
    system("pause");
    return;
}
```

i. 辅助函数（具体用途见方法实现代码注释）

1. 截取字符串（拆分信息）

```
//截取字符串
//例：
//origin：申请人:test_student
//则 a->申请人 b->test_student
void mySubstr(string& a, string& b, string& origin) {
    if (origin.find(':') != string::npos) {
        a = origin.substr(0, origin.find(':'));
        b = origin.substr(origin.find(':') + 1, origin.size() - origin.find(':') - 1);
    }
}
```

2. 容器初始化（初始化预约信息汇总容器）

```
//初始化预约信息汇总容器
//键值：第 x 条信息
//实值：信息标题+信息内容
void setOrderStatic(map<int, map<string, string>>& m) {
    ifstream ifs;
    ifs.open(OrderStaticFile, ios::in);
    //申请人用户名，日期，开始时间，结束时间，房间名，状态
    string applicantName, date, startTime, endTime, roomName, status;
    //第 x 条信息
    int num = 0;
    while (ifs >> applicantName>> roomName >> date >> startTime >> endTime >> status) {
        num++;
        string key, value;
        map<string, string> temp;
```

```
        //获取预约信息字符串
        mySubstr(key, value, applicantName);
        temp.insert(pair<string, string>(key, value));
        mySubstr(key, value, roomName);
        temp.insert(pair<string, string>(key, value));
        mySubstr(key, value, date);
        temp.insert(pair<string, string>(key, value));
        mySubstr(key, value, startTime);
        temp.insert(pair<string, string>(key, value));
        mySubstr(key, value, endTime);
        temp.insert(pair<string, string>(key, value));
        mySubstr(key, value, status);
        temp.insert(pair<string, string>(key, value));
        m.insert(make_pair(num, temp));
    }
    ifs.close();
}
```

3. 更新预约信息汇总数据文件

```
//更新预约信息汇总数据文件
void updateOrderFile(map<int, map<string, string>>& m) {
    ofstream ofs;
    //清空文件内容
    ofs.open(OrderStaticFile, ios::out | ios::trunc);
    for (int i = 1; i <= m.size(); i++) {
        //依次输入容器信息
        ofs << "申请人:" << m[i]["申请人"] << " " << "机房名:" << m[i]["机房名"] << " " << "日期:" << m[i]["日期"] << " " << "开始时间:" << m[i]["开始时间"] << " " << "结束时间:" << m[i]["结束时间"] << " " << "状态:" << m[i]["状态"] << endl;
    }
    ofs.close();
}
```

ii. 类方法实现

1. 构造函数

```
Student::Student(int id, string name, string password)
{
    studentID = id;
    Name = name;
    Password = password;
    //初始化机房信息
    string RoomName;
    int RoomNum;
    int ComputerNum;
    int StartTime;
    int EndTime;
    ifstream ifs;
```

```
    string str1, str;
    //初始化机房信息
    ifs.open(ComputerRoomFile);
    while (getline(ifs, str1)) {
        RoomName = str1.substr(str1.find("南"), str1.find('\t') - str1.find("南"));
        RoomNum = stoi(str1.substr(str1.find("楼") + 2, str1.find('\t') - str1.find("楼")));
        ComputerNum = stoi(str1.substr(str1.find("量") + 4, str1.find("开") - 5 - str1.find("量")));
        StartTime = stoi(str1.substr(str1.find("间") + 4, str1.find("-") - str1.find("间") - 4));
        EndTime = stoi(str1.substr(str1.find("-") + 1, str1.length() - str1.find("-") - 5));
        vecComputerRoom.push_back(ComputerRoom(RoomName, RoomNum, ComputerNum, StartTime, EndTime));
    }
    ifs.close();
}
```

2. 申请预约（根据提示输入预约信息，并将预约信息存入数据文档）

```
//申请预约
void Student::apply()
{
    int date, startTime, endTime, roomNum;
    string roomName;
    cout << "----------------------------------------" << endl;
    cout << "有以下机房开放：" << endl;
    //显示可申请机房
    for (int i = 1; i <= vecComputerRoom.size(); i++) {
        cout<<i<<"、"<<vecComputerRoom[i-1].RoomName<<'\t'<<"电脑数量："<< vecComputerRoom[i - 1].ComputerNum << '\t' << "开放时间：" << setw(2) << setfill('0') << vecComputerRoom[i - 1].StartTime << '-' << setw(2) << setfill('0') << vecComputerRoom[i - 1].EndTime << "节课" << endl;
    }
    //选择机房
    while (true) {
        cout << "请选择你要预约的机房号码：" << endl;
        cin >> roomNum;
        if (roomNum >= 1 && roomNum <= vecComputerRoom.size()) {
            roomName = vecComputerRoom[roomNum - 1].RoomName;
            break;
        }
        cout << "输入有误，请重新输入。" << endl;
    }
    //选择时间
    cout << "----------------------------------------" << endl;
    cout << "机房开放时间为周一至周五。" << endl;
    cout << "1. 周一" << endl;
    cout << "2. 周二" << endl;
```

```
    cout << "3. 周三" << endl;
    cout << "4. 周四" << endl;
    cout << "5. 周五" << endl;
    while (true) {
        cout << "请输入申请预约的时间：" << endl;
        cin >> date;
        if (date >= 1 && date <= 5)
            break;
        cout << "输入有误，请重新输入。" << endl;
    }
    //选择时间 2
    cout << "----------------------------------------" << endl;
    cout << "机房开放时间为第" << vecComputerRoom[roomNum - 1].StartTime << "节课至第" <<
vecComputerRoom[roomNum - 1].EndTime << "节课。" << endl;
    while (true) {
        cout << "请输入申请预约的时间段：" << endl;
        cout << "开始时间（第 xx 节课）:";
        cin >> startTime;
        cout << "结束时间（第 xx 节课）:";
        cin >> endTime;
        //检测时间 2 是否符合
        if (startTime >= 1 && endTime <= 12 && endTime >= startTime && (startTime >=
vecComputerRoom[roomNum - 1].StartTime && endTime <= vecComputerRoom[roomNum - 1].EndTime)) {
            break;
        }
        else if (startTime < 1 || endTime > 12 || endTime < startTime) {
            cout << "输入有误，请重新输入。" << endl;
            continue;
        }
        else if (!(startTime >= vecComputerRoom[roomNum - 1].StartTime && endTime <=
vecComputerRoom[roomNum - 1].EndTime)) {
            cout << "不在开放时间。预约失败。" << endl;
            return;
        }
    }
    cout << "----------------------------------------" << endl;
    cout << "预约成功！请等待审核。" << endl;
    ofstream ofs;
    //打开预约信息文件，并追加预约记录
    ofs.open(OrderStaticFile, ios::app);
    ofs << "申请人:" << this->Name << " " << "机房名:" << roomName << " " << "日期:" << date <<
" " << "开始时间:" << startTime << " " << "结束时间:" << endTime << " " << "状态:" << 1 << endl;
    ofs.close();
    system("pause");
    return;
}
```

3. 查看已申请预约（打开预约信息汇总，按格式输出此帐号所有的申请记录）

```
//查看已申请预约
void Student::showMyApply()
{
    //获取所有预约信息;
    map<int, map<string, string>> AllOrderStatic;
    setOrderStatic(AllOrderStatic);
    if (AllOrderStatic.size() == 0) {
        cout << "暂无预约记录。" << endl;
        system("pause");
        return;
    }
    int n = 0;
    for (int i = 1; i <= AllOrderStatic.size(); i++) {
        if (this->Name == AllOrderStatic[i]["申请人"]) {
            n++;
            cout << n << "、" << "机房名:" << AllOrderStatic[i]["机房名"] << "\t" << "日期:";
            //数字转周 x
            string date;
            if (AllOrderStatic[i]["日期"] == "1") {
                date = "周一";
            }
            else if (AllOrderStatic[i]["日期"] == "2") {
                date = "周二";
            }
            else if (AllOrderStatic[i]["日期"] == "3") {
                date = "周三";
            }
            else if (AllOrderStatic[i]["日期"] == "4") {
                date = "周四";
            }
            else if (AllOrderStatic[i]["日期"] == "5") {
                date = "周五";
            }
            cout << date << "\t" << "开始时间:第" << AllOrderStatic[i]["开始时间"] << "节
课" << "\t" << "结束时间:第" << AllOrderStatic[i]["结束时间"] << "节课" << "\t" << "状态:";
            //1：审核中 2：审核通过 3：审核失败 4：已取消
            string status;
            if (AllOrderStatic[i]["状态"] == "1") {
                status = "审核中";
            }
            else if (AllOrderStatic[i]["状态"] == "2") {
                status = "审核通过";
            }
            else if (AllOrderStatic[i]["状态"] == "3") {
                status = "审核失败";
```

```
            }
            else if (AllOrderStatic[i]["状态"] == "4") {
                status = "已取消";
            }
            cout << status << endl;
        }
    }
    if (n == 0) {
        cout << "暂无预约信息。" << endl;
    }
    system("pause");
    return;
}
```

4. 查看机房使用情况（打开预约信息汇总，按格式输出所有的申请记录）

```
//查看机房使用情况
void Student::showAllApply()
{
    map<int, map<string, string>> AllOrderStatic;
    setOrderStatic(AllOrderStatic);
    if (AllOrderStatic.size() == 0) {
        cout << "暂无预约记录。" << endl;
        system("pause");
        return;
    }
    for (int i = 1; i <= AllOrderStatic.size(); i++) {
        cout << i << "、" << "申请人:" << AllOrderStatic[i]["申请人"] << "\t" << "机房名:" <<
AllOrderStatic[i]["机房名"] << "\t" << "日期:";
        //数字转周 x
        string date;
        if (AllOrderStatic[i]["日期"] == "1") {
            date = "周一";
        }
        else if (AllOrderStatic[i]["日期"] == "2") {
            date = "周二";
        }
        else if (AllOrderStatic[i]["日期"] == "3") {
            date = "周三";
        }
        else if (AllOrderStatic[i]["日期"] == "4") {
            date = "周四";
        }
        else if (AllOrderStatic[i]["日期"] == "5") {
            date = "周五";
        }
        cout << date << "\t" << "开始时间:第" << AllOrderStatic[i]["开始时间"] << "节课" << "\t"
<< "结束时间:第" << AllOrderStatic[i]["结束时间"] << "节课" << "\t" << "状态:";
```

```
        //1：审核中 2：审核通过 3：审核失败 4：已取消
        string status;
        if (AllOrderStatic[i]["状态"] == "1") {
            status = "审核中";
        }
        else if (AllOrderStatic[i]["状态"] == "2") {
            status = "审核通过";
        }
        else if (AllOrderStatic[i]["状态"] == "3") {
            status = "审核失败";
        }
        else if (AllOrderStatic[i]["状态"] == "4") {
            status = "已取消";
        }
        cout << status << endl;
    }
    system("pause");
    return;
}
```

5. 取消预约（打开预约信息汇总，输出所有此账号可以取消申请记录，供用户选择）

```
//取消预约
void Student::cancelApply()
{
    //获取预约记录
    map<int, map<string, string>> AllOrderStatic;
    setOrderStatic(AllOrderStatic);
    if (AllOrderStatic.size() == 0) {
        cout << "暂无预约记录。" << endl;
        system("pause");
        return;
    }
    cout << "注：审核中或审核通过的预约记录可以被取消。" << endl;
    //键值：记录个人第 n 条/实值：总共第 i 条的对应关系
    map<int, int> recordNum;
    int n = 0;
    for (int i = 1; i <= AllOrderStatic.size(); i++) {
        if (this->Name == AllOrderStatic[i]["申请人"] && (AllOrderStatic[i]["状态"] == "1" ||
AllOrderStatic[i]["状态"] == "2")) {
            n++;
            recordNum.insert(make_pair(n, i));
            cout << n << "、" << "机房名:" << AllOrderStatic[i]["机房名"] << "\t" << "日期:";
            //数字转周 x
            string date;
            if (AllOrderStatic[i]["日期"] == "1") {
                date = "周一";
```

```
            }
            else if (AllOrderStatic[i]["日期"] == "2") {
                date = "周二";
            }
            else if (AllOrderStatic[i]["日期"] == "3") {
                date = "周三";
            }
            else if (AllOrderStatic[i]["日期"] == "4") {
                date = "周四";
            }
            else if (AllOrderStatic[i]["日期"] == "5") {
                date = "周五";
            }
            cout << date << "\t" << "开始时间:第" << AllOrderStatic[i]["开始时间"] << "节课
" << "\t" << "结束时间:第" << AllOrderStatic[i]["结束时间"] << "节课" << "\t" << "状态:";
            //1：审核中 2：审核通过 3：审核失败 4：已取消
            string status;
            if (AllOrderStatic[i]["状态"] == "1") {
                status = "审核中";
            }
            else if (AllOrderStatic[i]["状态"] == "2") {
                status = "审核通过";
            }
            cout << status << endl;
        }
    }
    if (n == 0) {
        cout << "暂无可以被取消的预约申请。" << endl;
        system("pause");
        return;
    }
    cout << "请选择你需要取消预约记录。" << endl;
    int select;
    cin >> select;
    if (select < 1 || select > n) {
        cout << "选择错误！" << endl;
        system("pause");
        return;
    }
    cout << "是否确认取消第" << select << "条预约？" << endl;
    cout << "1: 确认" << endl;
    cout << "2: 取消" << endl;
    int select2;
    cin >> select2;
    if (select2 == 1) {
        //更新状态为已取消
        AllOrderStatic[recordNum[select]]["状态"] = "4";
```

```
        //更新数据文件
        updateOrderFile(AllOrderStatic);
        cout << "取消成功！" << endl;
        system("pause");
        return;
    }
    else {
        cout << "取消失败！" << endl;
        system("pause");
        return;
    }
    return;
}
```

d) 教师类实现

i. 辅助函数（同学生类辅助函数，不再重复展示）

ii. 类方法实现

1. 查看机房使用情况（打开预约信息汇总，按格式输出所有的申请记录）（同学生类实现，不再重复展示）
2. 审核预约（打开预约信息汇总，输出所有需要被申请记录（状态为审核中），供用户选择是否审核通过）

```
//审核预约
void Teacher::checkApply()
{
    //键值：信息标题
    //实值：信息内容
    map<int, map<string, string>> AllOrderStatic;
    setOrderStatic1(AllOrderStatic);
    if (AllOrderStatic.size() == 0) {
        cout << "暂无预约记录。" << endl;
        system("pause");
        return;
    }
    cout << "注：审核中的预约申请可以被操作。" << endl;
    //键值：记录个人第 n 条/实值：总共第 i 条的对应关系
    map<int, int> recordNum;
    int n = 0;
    for (int i = 1; i <= AllOrderStatic.size(); i++) {
        if (AllOrderStatic[i]["状态"] == "1") {
            n++;
            recordNum.insert(make_pair(n, i));
            cout << n << "、" << "机房名:" << AllOrderStatic[i]["机房名"] << "\t" << "日
期:";
            //数字转周 x
            string date;
            if (AllOrderStatic[i]["日期"] == "1") {
                date = "周一";
```

```
            }
            else if (AllOrderStatic[i]["日期"] == "2") {
                date = "周二";
            }
            else if (AllOrderStatic[i]["日期"] == "3") {
                date = "周三";
            }
            else if (AllOrderStatic[i]["日期"] == "4") {
                date = "周四";
            }
            else if (AllOrderStatic[i]["日期"] == "5") {
                date = "周五";
            }
            cout << date << "\t" << "开始时间:第" << AllOrderStatic[i]["开始时间"] << "节课
" << "\t" << "结束时间:第" << AllOrderStatic[i]["结束时间"] << "节课" << "\t" << "状态:";
            //1：审核中 2：审核通过 3：审核失败 4：已取消
            string status;
            if (AllOrderStatic[i]["状态"] == "1") {
                status = "审核中";
            }
            cout << status << endl;
        }
    }
    //总数为零
    if (n == 0) {
        cout << "暂无需要审核的预约申请。" << endl;
        system("pause");
        return;
    }
    cout << "请选择你需要审核的预约申请。" << endl;
    int select;
    cin >> select;
    if (select < 1 || select > n) {
        cout << "选择错误！" << endl;
        system("pause");
        return;
    }
    cout << "请输入第" << select << "条预约记录的审核结果。" << endl;
    cout << "1: 通过" << endl;
    cout << "2: 不通过" << endl;
    int select2;
    cin >> select2;
    if (select2 == 1) {
        //更新状态为审核通过
        AllOrderStatic[recordNum[select]]["状态"] = "2";
        //更新数据文件
        updateOrderFile1(AllOrderStatic);
```

```
            cout << "审核通过！" << endl;
            system("pause");
            return;
        }
        else if (select2 == 2) {
            //更新状态为审核失败
            AllOrderStatic[recordNum[select]]["状态"] = "3";
            //更新数据文件
            updateOrderFile1(AllOrderStatic);
            cout << "审核不通过！" << endl;
            system("pause");
            return;
        }
        return;
}
```

四、 程序使用说明

***注：压缩文件中有三个文件夹（“数据”，“空数据”，“测试数据”）

每个文件夹中应该包含五个文件：

1. 管理员帐号信息.txt
2. 老师帐号信息.txt
3. 学生帐号信息.txt
4. 机房信息.txt
5. 预约信息汇总.txt

“数据”：请将需要进行操作的文件放进此文件夹中，执行过程也将对此文件夹中的 txt 文件进行操作。

“空数据”：此文件夹中除了提供了一个管理员帐号

（帐号：test_admin 密码：123456）

其他文件均为空文件，可以从零开始进行测试。

“测试数据”：此文件夹中提供了多组帐号（见下），含有几十组数据，可以用作测试数据进行测试。

以下演示实验均使用的时“测试数据”。

以下提供三个测试账号：

管理员 用户名：test_admin 密码：123456

学生 学号：2019150000 用户名：test_student 密码：123456

老师 职工号：10001 用户名：test_teacher 密码：123456

（其他账号见文件）

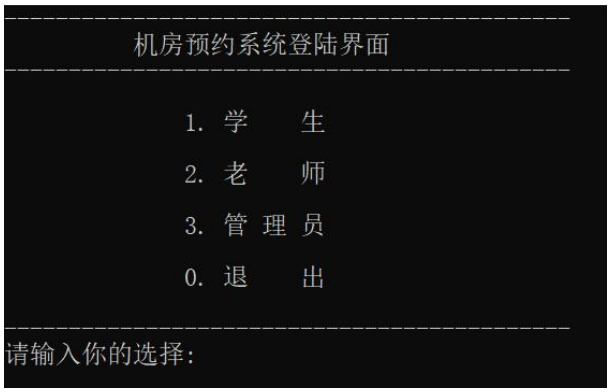
五、 测试及结果截图

用户使用手册：（如果用户选择了非界面提示的选项或内容，均会报错，效果类似，不作重复演示）

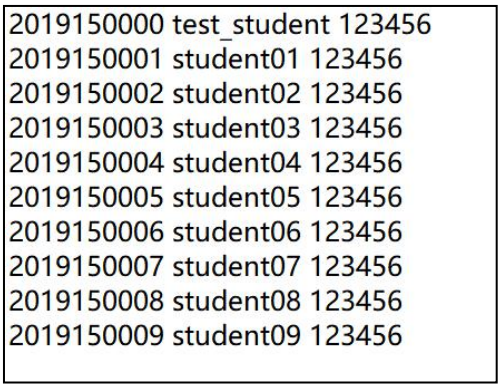
1. 登陆系统

(1) 登录功能：（以登录学生为例，老师/管理员效果类似）

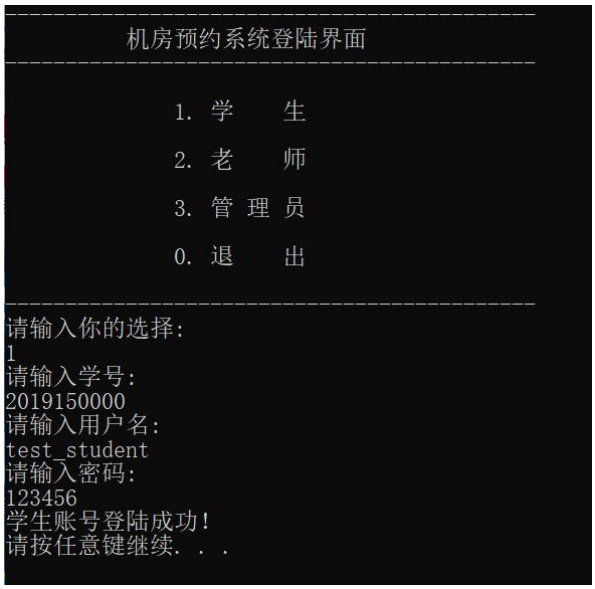
主界面：



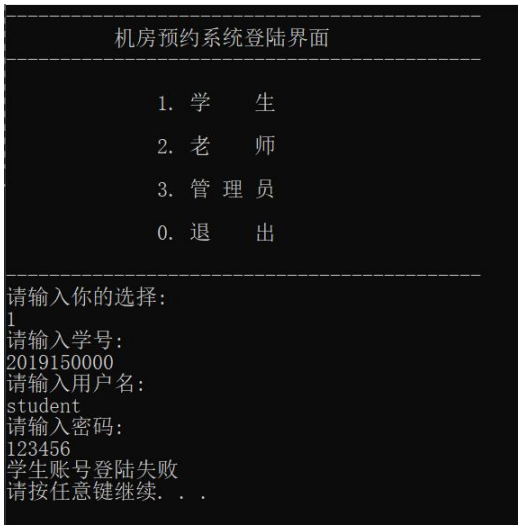
学生帐号信息：



情况 1：输入信息与账号列表匹配，登陆成功，并跳转学生功能界面



情况 2：学号/用户名/密码与账号列表不匹配，登陆失败，返回主菜单重新选择。



(2) 注销功能：（以注销学生帐号为例，老师/管理员效果类似）

选择对应选项，注销账号，返回主页。

```
欢迎学生test_student!
------------------------------------------
              学生功能界面
------------------------------------------

              1. 申请预约
              2. 查看已申请预约
              3. 查看机房使用情况
              4. 取消预约
              5. 注销登录

------------------------------------------
请输入你的选择:
5
注销成功
请按任意键继续. . .
```

2. 管理员功能

(1) 注册账号

（以注册学生账号为例，老师账号效果相同）
原始学生账号信息：

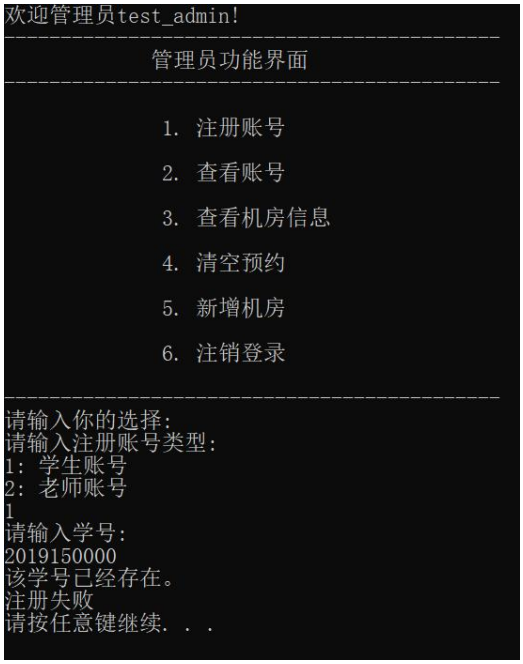
```
2019150000 test_student 123456
2019150001 student01 123456
2019150002 student02 123456
2019150003 student03 123456
2019150004 student04 123456
2019150005 student05 123456
2019150006 student06 123456
2019150007 student07 123456
2019150008 student08 123456
2019150009 student09 123456
```

情况 1：（正常注册）根据提示，依次输入学号/用户名/密码，完成注册。

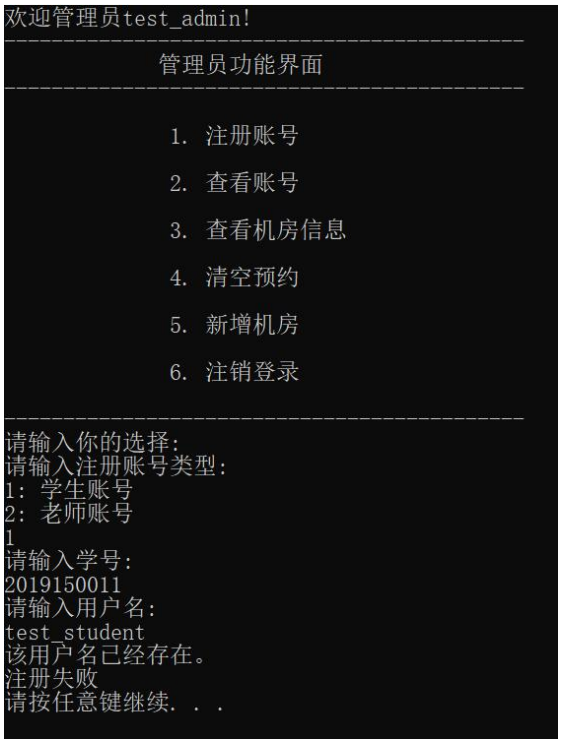
```
欢迎管理员test_admin!
------------------------------------------
              管理员功能界面
------------------------------------------

              1. 注册账号
              2. 查看账号
              3. 查看机房信息
              4. 清空预约
              5. 新增机房
              6. 注销登录

------------------------------------------
请输入你的选择:
请输入注册账号类型:
1: 学生账号
2: 老师账号
1
请输入学号:
2019150010
请输入用户名:
test01
请输入密码:
654321
账号注册成功!
请按任意键继续. . .
```

```
2019150000 test_student 123456
2019150001 student01 123456
2019150002 student02 123456
2019150003 student03 123456
2019150004 student04 123456
2019150005 student05 123456
2019150006 student06 123456
2019150007 student07 123456
2019150008 student08 123456
2019150009 student09 123456
2019150010 test01 654321
```

情况 2：（学号已经存在（职工号重复效果相同））
根据提示，输入学号，若重复，会报错，停止注册，返回上级菜单。



情况 3：（用户名已经存在）
注：当学号重复检查通过后，才会检查用户名是否重复。
根据提示，输入学号/用户名，若重复，会报错，停止注册，返回上级菜单。



情况 2/3 均不会对数据文件产生影响。

(2) 查看账号

根据提示，可以查看已成功的学生/老师帐号。

```
请输入你的选择:
2
请输入需要查看的账号类型:
1: 学生账号
2: 老师账号
1
学生帐号信息如下:
学号: 2019150000      用户名: test_student    密码: 123456
学号: 2019150001      用户名: student01       密码: 123456
学号: 2019150002      用户名: student02       密码: 123456
学号: 2019150003      用户名: student03       密码: 123456
学号: 2019150004      用户名: student04       密码: 123456
学号: 2019150005      用户名: student05       密码: 123456
学号: 2019150006      用户名: student06       密码: 123456
学号: 2019150007      用户名: student07       密码: 123456
学号: 2019150008      用户名: student08       密码: 123456
学号: 2019150009      用户名: student09       密码: 123456
学号: 2019150010      用户名: test01          密码: 654321
请按任意键继续. . .
```

```
2019150000 test_student 123456
2019150001 student01 123456
2019150002 student02 123456
2019150003 student03 123456
2019150004 student04 123456
2019150005 student05 123456
2019150006 student06 123456
2019150007 student07 123456
2019150008 student08 123456
2019150009 student09 123456
2019150010 test01 654321
```

```
请输入你的选择:
2
请输入需要查看的账号类型:
1: 学生账号
2: 老师账号
2
老师账号信息如下:
职工号: 10001         用户名: test_teacher    密码: 123456
职工号: 10002         用户名: teacher1        密码: 123456
职工号: 10003         用户名: teacher2        密码: 123456
职工号: 10004         用户名: teacher4        密码: 123456
职工号: 10005         用户名: teacher5        密码: 123456
职工号: 10006         用户名: teacher6        密码: 123456
职工号: 10007         用户名: teacher7        密码: 123456
职工号: 10008         用户名: teacher8        密码: 123456
请按任意键继续. . .
```

```
10001 test_teacher 123456
10002 teacher1 123456
10003 teacher2 123456
10004 teacher4 123456
10005 teacher5 123456
10006 teacher6 123456
10007 teacher7 123456
10008 teacher8 123456
```

如果无账号信息，会有提示反馈。

```
请输入你的选择:
2
请输入需要查看的账号类型:
1: 学生账号
2: 老师账号
1
学生帐号信息如下:
暂无学生帐号信息。
请按任意键继续. . .
```

(3) 查看机房信息

根据提示，可以查看已开放的机房信息。

```
请输入你的选择:
3
机房信息如下:
机房名称: 南区计算机大楼325    电脑数量: 50    开放时间: 01-12节课
机房名称: 南区计算机大楼326    电脑数量: 70    开放时间: 04-12节课
机房名称: 南区计算机大楼327    电脑数量: 80    开放时间: 05-12节课
机房名称: 南区计算机大楼328    电脑数量: 45    开放时间: 03-12节课
机房名称: 南区计算机大楼401    电脑数量: 23    开放时间: 05-08节课
机房名称: 南区计算机大楼402    电脑数量: 60    开放时间: 02-07节课
请按任意键继续. . .
```

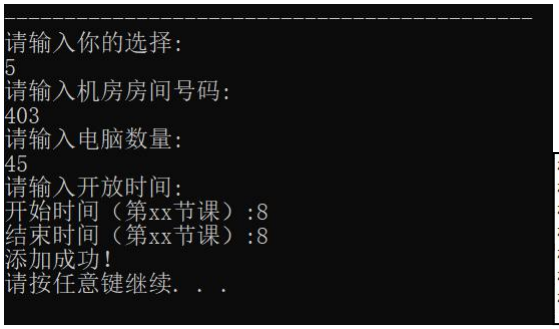
```
机房名称: 南区计算机大楼325 电脑数量: 50    开放时间: 01-12节课
机房名称: 南区计算机大楼326 电脑数量: 70    开放时间: 04-12节课
机房名称: 南区计算机大楼327 电脑数量: 80    开放时间: 05-12节课
机房名称: 南区计算机大楼328 电脑数量: 45    开放时间: 03-12节课
机房名称: 南区计算机大楼401 电脑数量: 23    开放时间: 05-08节课
机房名称: 南区计算机大楼402 电脑数量: 60    开放时间: 02-07节课
```

如果无机房信息，会有提示反馈。

(4) 新增机房

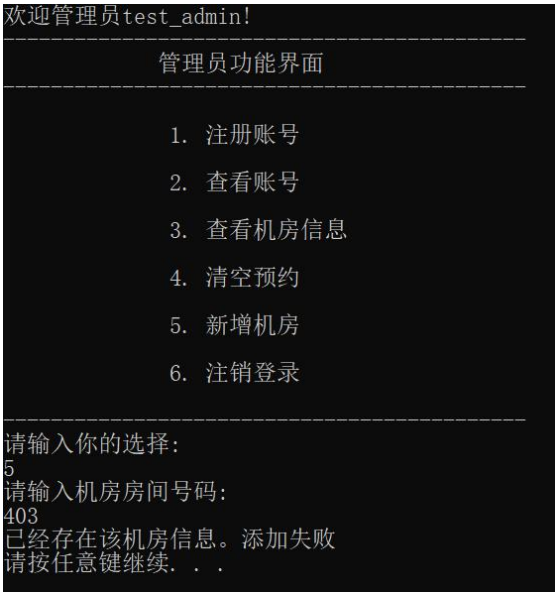
原始机房信息：

情况 1：（正常添加）根据提示，依次输入房间号码/电脑数量/开放时间，完成注册。



情况 2：（房间号码重复）

根据提示，输入房间号码，若重复，会报错，停止注册，返回上级菜单。



情况 3：（开放时间设置错误）

注：当房间号重复检查通过后，才会检查用户名是否重复。

根据提示，输入房间号/开放时间，若不符合，会报错，停止注册，返回上级菜单。

以下两种情况会报错：1. 开始时间晚于结束时间　　2. 开放时间不属于 01-12 节课

```
请输入你的选择:
5
请输入机房房间号码:
404
请输入电脑数量:
10
请输入开放时间:
开始时间（第xx节课）:5
结束时间（第xx节课）:1
时间输入错误。添加失败
请按任意键继续. . .
```

```
请输入你的选择:
5
请输入机房房间号码:
404
请输入电脑数量:
10
请输入开放时间:
开始时间（第xx节课）:5
结束时间（第xx节课）:13
时间输入错误。添加失败
请按任意键继续. . .
```

情况 2/3 均不会对数据文件产生影响。

(5) 清空预约

根据提示，选择操作。

```
请输入你的选择:
4
是否确认清空预约信息?
1: 确认
2: 返回
1
清空成功!
请按任意键继续. . .
```

原始预约信息汇总：

操作后预约信息汇总：

3. 学生功能

(1) 查看已申请预约

根据提示，选择操作，可以查看该用户所有申请过的预约记录。

```
请输入你的选择:
2
1、机房名:南区计算机大楼325    日期:周五    开始时间:第2节课    结束时间:第12节课    状态:审核中
2、机房名:南区计算机大楼401    日期:周一    开始时间:第5节课    结束时间:第6节课    状态:审核通过
3、机房名:南区计算机大楼327    日期:周三    开始时间:第10节课    结束时间:第10节课    状态:审核通过
4、机房名:南区计算机大楼402    日期:周一    开始时间:第4节课    结束时间:第4节课    状态:审核通过
5、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
6、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:审核通过
7、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核失败
8、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:已取消
9、机房名:南区计算机大楼401    日期:周二    开始时间:第5节课    结束时间:第5节课    状态:审核通过
请按任意键继续. . .
```

如果无预约记录信息，会有提示反馈。

```
--------------------------------------------
请输入你的选择：
2
暂无预约记录。
请按任意键继续. . .
```

(2) 申请预约

情况 1：（预约成功）根据提示，依次输入机房号码/日期/时间段，完成申请。

```
--------------------------------------------
请输入你的选择：
1
--------------------------------------------
有以下机房开放：
1、南区计算机大楼325    电脑数量：50    开放时间：01-12节课
2、南区计算机大楼326    电脑数量：70    开放时间：04-12节课
3、南区计算机大楼327    电脑数量：80    开放时间：05-12节课
4、南区计算机大楼328    电脑数量：45    开放时间：03-12节课
5、南区计算机大楼401    电脑数量：23    开放时间：05-08节课
6、南区计算机大楼402    电脑数量：60    开放时间：02-07节课
7、南区计算机大楼403    电脑数量：45    开放时间：08-08节课
请选择你要预约的机房号码：
5
--------------------------------------------
机房开放时间为周一至周五。
1. 周一
2. 周二
3. 周三
4. 周四
5. 周五
请输入申请预约的时间：
4
--------------------------------------------
机房开放时间为第5节课至第8节课。
请输入申请预约的时间段：
开始时间（第xx节课）:5
结束时间（第xx节课）:6
--------------------------------------------
预约成功！请等待审核。
请按任意键继续. . .
```

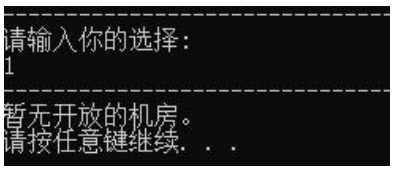
```
--------------------------------------------
请输入你的选择：
2
1、机房名:南区计算机大楼325    日期:周五    开始时间:第2节课    结束时间:第12节课    状态:审核中
2、机房名:南区计算机大楼401    日期:周一    开始时间:第5节课    结束时间:第6节课    状态:审核通过
3、机房名:南区计算机大楼327    日期:周三    开始时间:第10节课    结束时间:第10节课    状态:审核通过
4、机房名:南区计算机大楼402    日期:周一    开始时间:第4节课    结束时间:第4节课    状态:审核通过
5、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
6、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:审核通过
7、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核失败
8、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:已取消
9、机房名:南区计算机大楼401    日期:周二    开始时间:第5节课    结束时间:第5节课    状态:审核通过
10、机房名:南区计算机大楼401    日期:周四    开始时间:第5节课    结束时间:第6节课    状态:审核中
请按任意键继续. . .
```

情况 2：（预约失败）根据提示，依次输入机房号码/日期/时间段，若时间段不在开放时间内或者时间段输入错误（1. 开始时间晚于结束时间　2. 时间段不属于 01-12 节课），显示错误信息，预约失败。

```
--------------------------------------------
请输入你的选择：
1
--------------------------------------------
有以下机房开放：
1、南区计算机大楼325    电脑数量：50    开放时间：01-12节课
2、南区计算机大楼326    电脑数量：70    开放时间：04-12节课
3、南区计算机大楼327    电脑数量：80    开放时间：05-12节课
4、南区计算机大楼328    电脑数量：45    开放时间：03-12节课
5、南区计算机大楼401    电脑数量：23    开放时间：05-08节课
6、南区计算机大楼402    电脑数量：60    开放时间：02-07节课
请选择你要预约的机房号码：
5
--------------------------------------------
机房开放时间为周一至周五。
1. 周一
2. 周二
3. 周三
4. 周四
5. 周五
请输入申请预约的时间：
4
--------------------------------------------
机房开放时间为第5节课至第8节课。
请输入申请预约的时间段：
开始时间（第xx节课）:6
结束时间（第xx节课）:9
不在开放时间。预约失败。
请按任意键继续. . .
```

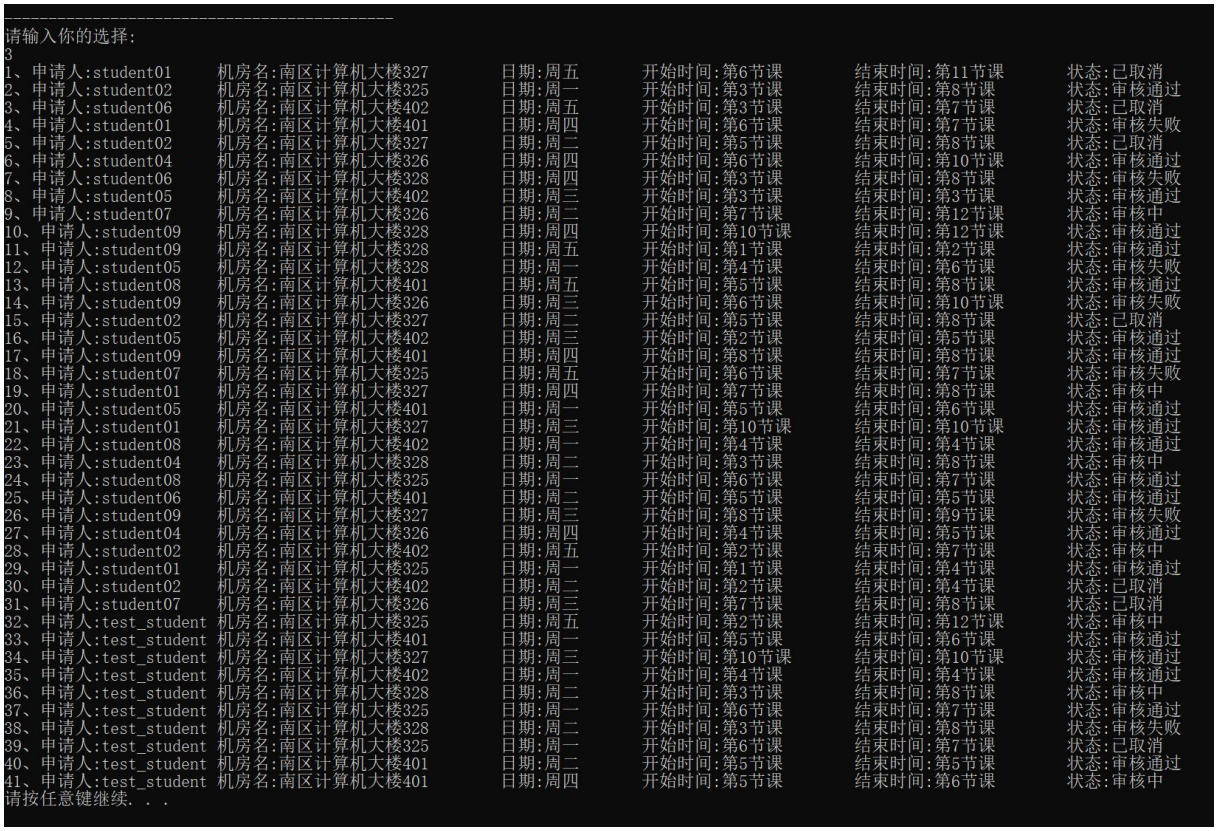
```
--------------------------------------------
请输入你的选择：
1
--------------------------------------------
有以下机房开放：
1、南区计算机大楼325    电脑数量：50    开放时间：01-12节课
2、南区计算机大楼326    电脑数量：70    开放时间：04-12节课
3、南区计算机大楼327    电脑数量：80    开放时间：05-12节课
4、南区计算机大楼328    电脑数量：45    开放时间：03-12节课
5、南区计算机大楼401    电脑数量：23    开放时间：05-08节课
6、南区计算机大楼402    电脑数量：60    开放时间：02-07节课
请选择你要预约的机房号码：
5
--------------------------------------------
机房开放时间为周一至周五。
1. 周一
2. 周二
3. 周三
4. 周四
5. 周五
请输入申请预约的时间：
4
--------------------------------------------
机房开放时间为第5节课至第8节课。
请输入申请预约的时间段：
开始时间（第xx节课）:6
结束时间（第xx节课）:13
输入有误，请重新输入。
```

如果无开放的机房，会有提示反馈。



(3) 查看机房使用情况

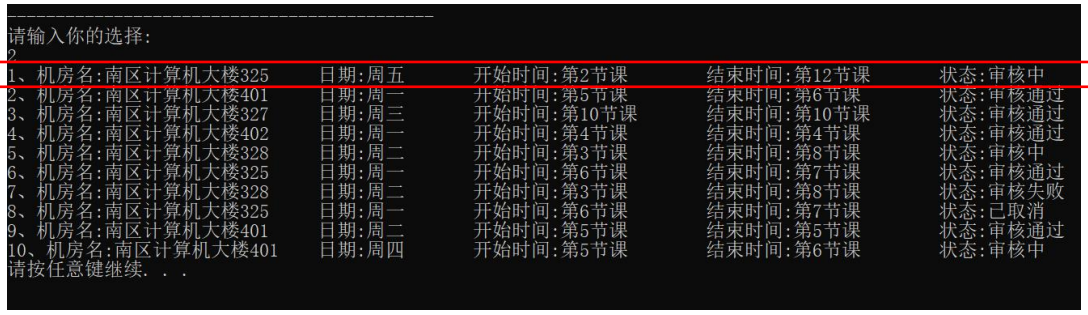
根据提示，选择操作，可以查看所有申请预约记录，从而帮助判断自己何时预约。



如果无预约记录信息，会有提示反馈。

(4) 取消预约

根据提示，选择操作，只可以取消状态为审核中或审核通过的预约记录。
原始预约记录：



操作：

```
------------------------------------------
请输入你的选择:
4
注：审核中或审核通过的预约记录可以被取消。
1、机房名:南区计算机大楼325    日期:周五    开始时间:第2节课    结束时间:第12节课    状态:审核中
2、机房名:南区计算机大楼401    日期:周一    开始时间:第5节课    结束时间:第6节课    状态:审核通过
3、机房名:南区计算机大楼327    日期:周三    开始时间:第10节课    结束时间:第10节课    状态:审核通过
4、机房名:南区计算机大楼402    日期:周一    开始时间:第4节课    结束时间:第4节课    状态:审核通过
5、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
6、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:审核通过
7、机房名:南区计算机大楼401    日期:周二    开始时间:第5节课    结束时间:第5节课    状态:审核通过
8、机房名:南区计算机大楼401    日期:周四    开始时间:第5节课    结束时间:第6节课    状态:审核中
请选择你需要取消预约记录。
1
是否确认取消第1条预约?
1: 确认
2: 取消
1
取消成功!
请按任意键继续. . .
```

结果：

```
------------------------------------------------
请输入你的选择:
2
1、机房名:南区计算机大楼325    日期:周五    开始时间:第2节课    结束时间:第12节课    状态:已取消
2、机房名:南区计算机大楼401    日期:周一    开始时间:第5节课    结束时间:第6节课    状态:审核通过
3、机房名:南区计算机大楼327    日期:周三    开始时间:第10节课    结束时间:第10节课    状态:审核通过
4、机房名:南区计算机大楼402    日期:周一    开始时间:第4节课    结束时间:第4节课    状态:审核通过
5、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
6、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:审核通过
7、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核失败
8、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:已取消
9、机房名:南区计算机大楼401    日期:周二    开始时间:第5节课    结束时间:第5节课    状态:审核通过
10、机房名:南区计算机大楼401    日期:周四    开始时间:第5节课    结束时间:第6节课    状态:审核中
请按任意键继续. . .
```

4. 老师功能

(1) 查看机房使用情况

根据提示，选择操作，可以查看所有申请预约记录，从而帮助判断自己何时预约。

```
------------------------------------------
请输入你的选择:
3
1、申请人:student01    机房名:南区计算机大楼327    日期:周五    开始时间:第6节课    结束时间:第11节课    状态:已取消
2、申请人:student02    机房名:南区计算机大楼325    日期:周一    开始时间:第3节课    结束时间:第8节课    状态:审核通过
3、申请人:student06    机房名:南区计算机大楼402    日期:周五    开始时间:第3节课    结束时间:第7节课    状态:已取消
4、申请人:student01    机房名:南区计算机大楼401    日期:周四    开始时间:第6节课    结束时间:第7节课    状态:审核失败
5、申请人:student02    机房名:南区计算机大楼327    日期:周二    开始时间:第5节课    结束时间:第8节课    状态:已取消
6、申请人:student04    机房名:南区计算机大楼326    日期:周四    开始时间:第6节课    结束时间:第10节课    状态:审核通过
7、申请人:student06    机房名:南区计算机大楼328    日期:周四    开始时间:第3节课    结束时间:第8节课    状态:审核失败
8、申请人:student05    机房名:南区计算机大楼402    日期:周三    开始时间:第3节课    结束时间:第3节课    状态:审核通过
9、申请人:student07    机房名:南区计算机大楼326    日期:周二    开始时间:第7节课    结束时间:第12节课    状态:审核中
10、申请人:student09    机房名:南区计算机大楼328    日期:周四    开始时间:第10节课    结束时间:第12节课    状态:审核通过
11、申请人:student09    机房名:南区计算机大楼328    日期:周五    开始时间:第1节课    结束时间:第2节课    状态:审核通过
12、申请人:student05    机房名:南区计算机大楼328    日期:周一    开始时间:第4节课    结束时间:第6节课    状态:审核失败
13、申请人:student08    机房名:南区计算机大楼401    日期:周五    开始时间:第5节课    结束时间:第8节课    状态:审核通过
14、申请人:student09    机房名:南区计算机大楼326    日期:周三    开始时间:第6节课    结束时间:第10节课    状态:审核失败
15、申请人:student02    机房名:南区计算机大楼327    日期:周二    开始时间:第5节课    结束时间:第8节课    状态:已取消
16、申请人:student05    机房名:南区计算机大楼402    日期:周三    开始时间:第2节课    结束时间:第5节课    状态:审核通过
17、申请人:student09    机房名:南区计算机大楼401    日期:周四    开始时间:第8节课    结束时间:第8节课    状态:审核通过
18、申请人:student07    机房名:南区计算机大楼325    日期:周五    开始时间:第6节课    结束时间:第7节课    状态:审核失败
19、申请人:student01    机房名:南区计算机大楼327    日期:周四    开始时间:第7节课    结束时间:第8节课    状态:审核中
20、申请人:student05    机房名:南区计算机大楼401    日期:周一    开始时间:第5节课    结束时间:第6节课    状态:审核通过
21、申请人:student01    机房名:南区计算机大楼327    日期:周三    开始时间:第10节课    结束时间:第10节课    状态:审核通过
22、申请人:student08    机房名:南区计算机大楼402    日期:周一    开始时间:第4节课    结束时间:第4节课    状态:审核通过
23、申请人:student04    机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
24、申请人:student08    机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:审核通过
25、申请人:student06    机房名:南区计算机大楼401    日期:周二    开始时间:第5节课    结束时间:第5节课    状态:审核通过
26、申请人:student09    机房名:南区计算机大楼327    日期:周三    开始时间:第8节课    结束时间:第9节课    状态:审核失败
27、申请人:student04    机房名:南区计算机大楼326    日期:周四    开始时间:第4节课    结束时间:第5节课    状态:审核通过
28、申请人:student02    机房名:南区计算机大楼402    日期:周五    开始时间:第2节课    结束时间:第7节课    状态:审核中
29、申请人:student01    机房名:南区计算机大楼325    日期:周一    开始时间:第1节课    结束时间:第4节课    状态:审核通过
30、申请人:student02    机房名:南区计算机大楼402    日期:周二    开始时间:第2节课    结束时间:第4节课    状态:已取消
31、申请人:student07    机房名:南区计算机大楼326    日期:周三    开始时间:第7节课    结束时间:第8节课    状态:已取消
32、申请人:test_student    机房名:南区计算机大楼325    日期:周五    开始时间:第2节课    结束时间:第12节课    状态:审核中
33、申请人:test_student    机房名:南区计算机大楼401    日期:周一    开始时间:第5节课    结束时间:第6节课    状态:审核通过
34、申请人:test_student    机房名:南区计算机大楼327    日期:周三    开始时间:第10节课    结束时间:第10节课    状态:审核通过
35、申请人:test_student    机房名:南区计算机大楼402    日期:周一    开始时间:第4节课    结束时间:第4节课    状态:审核通过
36、申请人:test_student    机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
37、申请人:test_student    机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:审核通过
38、申请人:test_student    机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核失败
39、申请人:test_student    机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:已取消
40、申请人:test_student    机房名:南区计算机大楼401    日期:周二    开始时间:第5节课    结束时间:第5节课    状态:审核通过
41、申请人:test_student    机房名:南区计算机大楼401    日期:周四    开始时间:第5节课    结束时间:第6节课    状态:审核中
请按任意键继续. . .
```

如果无预约记录信息，会有提示反馈。

```
--------------------------------
请输入你的选择:
1
暂无预约记录。
请按任意键继续. . .
```

(2) 审核预约

以 test_student 为例，展示变化。

test_student 查看已申请预约界面：

```
--------------------------------------------
请输入你的选择:
2
1、机房名:南区计算机大楼325    日期:周五    开始时间:第2节课    结束时间:第12节课    状态:审核中
2、机房名:南区计算机大楼401    日期:周一    开始时间:第5节课    结束时间:第6节课    状态:审核通过
3、机房名:南区计算机大楼327    日期:周三    开始时间:第10节课    结束时间:第10节课    状态:审核通过
4、机房名:南区计算机大楼402    日期:周一    开始时间:第4节课    结束时间:第4节课    状态:审核通过
5、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
6、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:审核通过
7、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核失败
8、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:已取消
9、机房名:南区计算机大楼401    日期:周二    开始时间:第5节课    结束时间:第5节课    状态:审核通过
请按任意键继续. . .
```

test_teacher 审核预约界面：

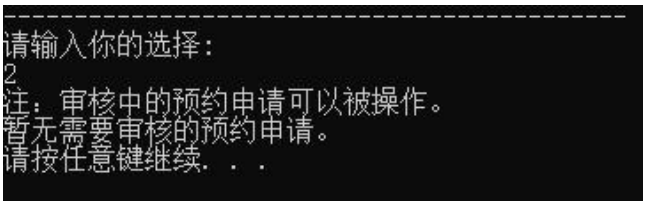
```
------------------------------------------
请输入你的选择:
2
注：审核中的预约申请可以被操作。
1、申请人:student07 机房名:南区计算机大楼326    日期:周二    开始时间:第7节课    结束时间:第12节课    状态:审核中
2、申请人:student01 机房名:南区计算机大楼327    日期:周四    开始时间:第7节课    结束时间:第8节课    状态:审核中
3、申请人:student04 机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
4、申请人:student02 机房名:南区计算机大楼402    日期:周五    开始时间:第2节课    结束时间:第7节课    状态:审核中
5、申请人:test_student 机房名:南区计算机大楼325 日期:周五    开始时间:第2节课    结束时间:第12节课    状态:审核中
6、申请人:test_student 机房名:南区计算机大楼328 日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
请选择你需要审核的预约申请。
5
请输入第5条预约记录的审核结果。
1: 通过
2: 不通过
1
审核通过!
请按任意键继续. . .
```

```
------------------------------------------
请输入你的选择:
2
注：审核中的预约申请可以被操作。
1、申请人:student07 机房名:南区计算机大楼326    日期:周二    开始时间:第7节课    结束时间:第12节课    状态:审核中
2、申请人:student01 机房名:南区计算机大楼327    日期:周四    开始时间:第7节课    结束时间:第8节课    状态:审核中
3、申请人:student04 机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
4、申请人:student02 机房名:南区计算机大楼402    日期:周五    开始时间:第2节课    结束时间:第7节课    状态:审核中
5、申请人:test_student 机房名:南区计算机大楼328 日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核中
请选择你需要审核的预约申请。
5
请输入第5条预约记录的审核结果。
1: 通过
2: 不通过
2
审核不通过!
请按任意键继续. . .
```

test_student 查看已申请预约界面：

```
------------------------------------------
请输入你的选择:
2
1、机房名:南区计算机大楼325    日期:周五    开始时间:第2节课    结束时间:第12节课    状态:审核通过
2、机房名:南区计算机大楼401    日期:周一    开始时间:第5节课    结束时间:第6节课    状态:审核通过
3、机房名:南区计算机大楼327    日期:周三    开始时间:第10节课    结束时间:第10节课    状态:审核通过
4、机房名:南区计算机大楼402    日期:周一    开始时间:第4节课    结束时间:第4节课    状态:审核通过
5、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核失败
6、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:审核通过
7、机房名:南区计算机大楼328    日期:周二    开始时间:第3节课    结束时间:第8节课    状态:审核失败
8、机房名:南区计算机大楼325    日期:周一    开始时间:第6节课    结束时间:第7节课    状态:已取消
9、机房名:南区计算机大楼401    日期:周二    开始时间:第5节课    结束时间:第5节课    状态:审核通过
请按任意键继续. . .
```

如果无需要审核的预约记录信息，会有提示反馈。



六、 总结

本次大作业基于 STL 完成了机房预约系统的初步设计，完成了一些基本的需求。在过程中，进一步的对于各种容器，算法，以及文件操作方式有了很深的了解。当然，该系统仍有一些不足，例如未能实现同一节课机房容量上限的判断，希望在未来可以进一步完善。最后感谢老师半个学期以来的教学，让我了解了 C++的泛型编程思想，拓展了视野。