聊城大学

数据结构课程设计报告

题目：机房预约系统

专业名称：软件工程

班 级： 2021级 5 班

团队成员：于正浩（2021400474）、

王文旭（2021400489）

目录

[一、设计目的 3](#_Toc121389584)

[二、设计内容 3](#_Toc121389585)

[三、概要设计 3](#_Toc121389586)

[1、功能模块图 3](#_Toc121389587)

[2、各个模块详细的功能描述 4](#_Toc121389588)

[四、详细设计 5](#_Toc121389589)

[1、 功能函数的调用关系图 5](#_Toc121389590)

[2、重点设计及编码 5](#_Toc121389591)

[（1）核心数据结构： 5](#_Toc121389592)

[（2）核心编码 6](#_Toc121389593)

[五、测试数据及运行结果 10](#_Toc121389594)

[1、正常的运行结果 10](#_Toc121389595)

[学生登录及操作 10](#_Toc121389596)

[2、教师的操作 13](#_Toc121389597)

[3、管理员的操作 13](#_Toc121389598)

[2、异常数据 16](#_Toc121389599)

[六、测试情况，设计技巧及体会 17](#_Toc121389600)

[1、改进方案 17](#_Toc121389601)

[2、心得体会 17](#_Toc121389602)

[七、参考文献 17](#_Toc121389603)

[八、附录 17](#_Toc121389604)

一、设计目的

1、通过本次课程设计巩固《数据结构》中所学的知识内容

2、提高组员的上级编程以及调试能力

3、提高组员沟通学习能力、团队合作能力

4、设计一个机房预约系统

二、设计内容

1、学校现有几个规格不同的机房，由于学生申请机房步骤复杂并且易出现“撞车”现象,现需求一个可以简化学生申请机房流程，避免“撞车”现象，开发一套机房预约系统，解决这一问题。

2、根据选择的身份进行登录，然后对应身份提供相应操作。

3、为学生提供申请预约、查看我的预约、查看所有预约、取消预约的操作。

4、为教师提供、查看所有预约、审核预约的操作。

5、为管理员提供添加账号、查看账号、查看机房、显示所有预约、清空预约的操作。

三、概要设计

设计思想：

运用了面向对象的设计思想；

使用的数据结构：

采用了线性表（vector）、映射（map）、散列表（unordered\_map）

1、功能模块图

图示

描述已自动生成

2、各个模块详细的功能描述

图示

描述已自动生成

图示

描述已自动生成

图示

描述已自动生成

四、详细设计

1、 功能函数的调用关系图

图示

描述已自动生成

2、重点设计及编码

（1）核心数据结构：

①map<int, unordered\_map<string, string>>M\_AllOrder; //记录所有预约信息的容器

②vector<tuple<string,Date,Time>> V\_DateAndTime; //记录预约状态为"Passed"或"Applying"的预约记录的机房编号和日期与时间段的容器

③vector<ComputerRoom>Vcom;//用来存放机房信息的容器

④vector<Student>Vstu;//用来存放学生账号信息的容器

⑤vector<Teacher>Vtea;//用来存放教师账号信息的容器

⑥vector<ComputerRoom>Vcom;//用来存放机房信息的容器

（2）核心编码

//判断两个时间段是否有相交部分//是返回true 不是返回false

bool Time::Is\_meet(Time& T)

{

//如果this的结束的时间小于T的开始时间 则不重叠

if ((this->End\_Hour < T.Begin\_Hour)|| ((this->End\_Hour == T.Begin\_Hour) && (this->End\_Minute <= T.Begin\_Minute)))

return false;

if ((this->Begin\_Hour > T.End\_Hour) || ((this->Begin\_Hour == T.End\_Hour) && (this->Begin\_Minute >= T.End\_Minute)))

return false;

return true;

}

//OrderLog无参构造函数

OrderLog::OrderLog()

{

//cout << "一构造" << endl;

ifstream ifs;

ifs.open(ORDER\_FILE, ios::in);

if (!ifs.is\_open())//检查文件是否打开

{

cout << "ORDER\_FILE文件打开失败" << endl;

ifs.close();

}

string ComRoomID;//机房编号

string date;//日期

string time;//时间段

string StuName;//学生姓名

string StuID;//学生学号

string status;//状态

this->Map\_size = 0;

while (ifs >> ComRoomID && ifs >> date && ifs >> time && ifs >> StuName && ifs >> StuID && ifs >> status)

{

string key;

string value;

bool V\_flag = true;//用来确定是否插入V\_DateAndTime容器中//为true 的时候插入，为false的时候不插入//

//为了让V\_DateAndTime 里面存的预约记录数据都是没被取消的//防止误判时间相冲的函数出现错误

unordered\_map<string, string > m;

string VcomputerID;// 用于存初始化Vector的string的变量//用来表示机房编号

// 用于存初始化日期的变量

int year=0;

int month=0;

int day=0;

// 用于存初始化具体时间的变量

int beginHour=-1;

int beginMin=-1;

int endHour=-1;

int endMin=-1;

int pos = ComRoomID.find(':');//分离ComRoomID冒号前后的部分

if (pos != -1)

{

key = ComRoomID.substr(0, pos);

value = ComRoomID.substr(pos + 1, ComRoomID.size() - pos - 1);

VcomputerID = value;//给VcomputerID赋值；

m.insert(make\_pair(key, value));

}

pos = date.find(':');//分离Date冒号前后的部分

if (pos != -1)

{

key = date.substr(0, pos);

value = date.substr(pos + 1, date.size() - pos - 1);

m.insert(make\_pair(key, value));

//将Date的每个参数分离出来

int p = value.find('/');//用于存储从头查找出的第一个‘/’的下标；

int q = value.rfind('/');//用于存储从尾查找出的第一个‘/’的下标//即为第二个‘/’的下标

stringstream ss;

ss << value.substr(0, p);

ss >> year;

//cout << year << endl;

ss.clear();

ss << value.substr(p + 1, q - p - 1);

ss >> month;

//cout << month << endl;

ss.clear();

ss << value.substr(q + 1, value.size() - p - 1);

ss >> day;

ss.clear();

//cout << day << endl;

}

pos = time.find(':');//分离Time冒号前后的部分

if (pos != -1)

{

key = time.substr(0, pos);

value = time.substr(pos + 1, time.size() - pos - 1);

m.insert(make\_pair(key, value));

//将Time的每个参数分离出来

int p = value.find(':');//用于存储从头查找出的第一个‘:’的下标；

int i = value.find("--");//用于存储从头查找出的第一个‘--’的下标；

int q = value.rfind(':');//用于存储从尾查找出的第一个‘:’的下标；

stringstream ss;

ss << value.substr(0, p);

ss >> beginHour;

ss.clear();

ss << value.substr(p + 1, i - p - 1);

ss >> beginMin;

ss.clear();

ss << value.substr(i + 2, q - i - 2);

ss >> endHour;

ss.clear();

ss << value.substr(q + 1, value.size() - q - 1);

ss >> endMin;

ss.clear();

}

pos = StuName.find(':');//分离StudentName冒号前后的部分

if (pos != -1)

{

key = StuName.substr(0, pos);

value = StuName.substr(pos + 1, StuName.size() - pos - 1);

m.insert(make\_pair(key, value));

}

pos = StuID.find(':');//分离StudentID冒号前后的部分

if (pos != -1)

{

key = StuID.substr(0, pos);

value = StuID.substr(pos + 1, StuID.size() - pos - 1);

m.insert(make\_pair(key, value));

}

pos = status.find(':');//分离status冒号前后的部分

if (pos != -1)

{

key = status.substr(0, pos);

value = status.substr(pos + 1, status.size() - pos - 1);

if (value == "Canceled"||value == "Failed")//如果该条预约记录的状态是 取消 的或是 预约失败 的

V\_flag = false;//把V\_flag标志置为假//表示不加入V\_DateAndTime容器之中

m.insert(make\_pair(key, value));

}

if(V\_flag==true)

{

Date D(year, month, day);//用参数初始化Date

Time T(beginHour, beginMin, endHour, endMin);//用参数初始化Time

V\_DateAndTime.push\_back(make\_tuple(VcomputerID, D, T));//将以 VcomputerID（机房编号），D（日期）

//T （时间段）组成的元组插入vector容器中

}

//将小的map容器的插入到大的map容器之中//用于表示所有的预约记录

//小的map容器表示一条预约记录

this->Map\_size++;

this->M\_AllOrder.insert(make\_pair(this->Map\_size, m));

}

ifs.close();

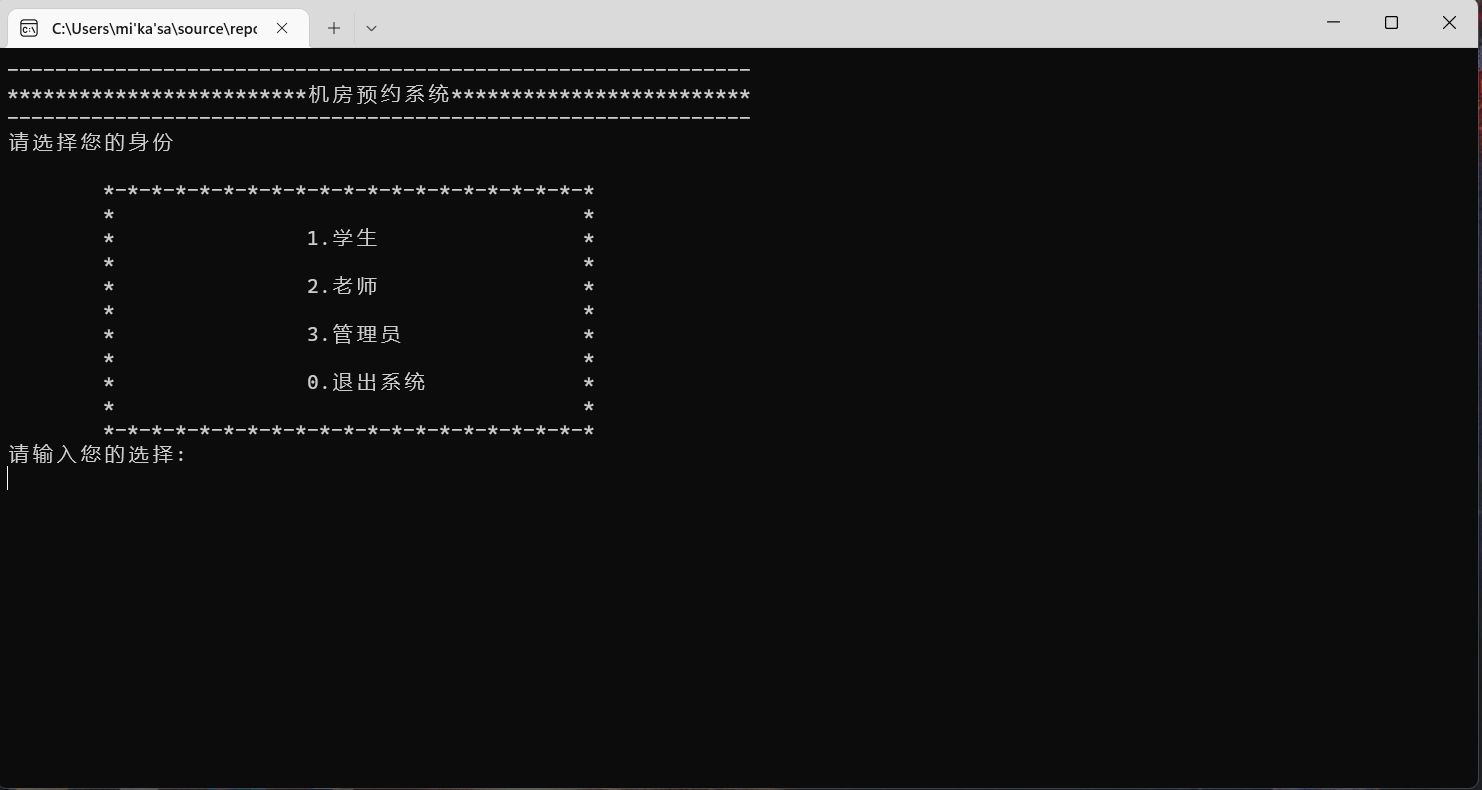
}

五、测试数据及运行结果

1、正常的运行结果

学生登录及操作

主界面：



输入1，然后输入在学生账号的txt文件中有的账号信息，进行登录。

（教师和管理员的登录操作与学生大同小异）

文本

中度可信度描述已自动生成

文本

描述已自动生成

* 进入学生菜单及操作界面

文本

描述已自动生成

在学生操作菜单界面输入1进行申请预约操作

文本

描述已自动生成

开始依次输入

B308 2022 10 23 16 00 18 00

文本

描述已自动生成

按任意键后回到学生操作菜单

文本

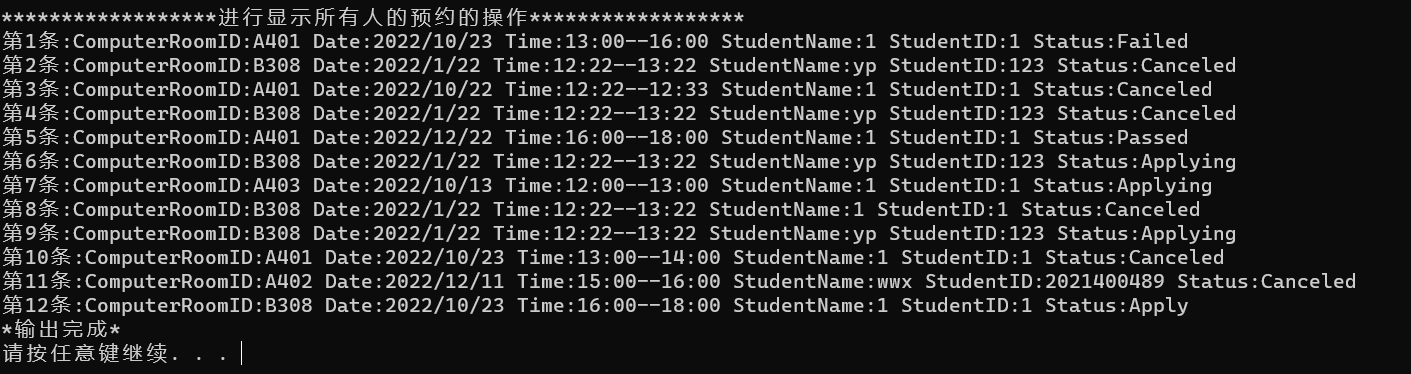
描述已自动生成

在学生操作菜单界面输入2显示我的预约

文本

描述已自动生成

回到学生操作菜单后输入3 进行显示所有预约的操作

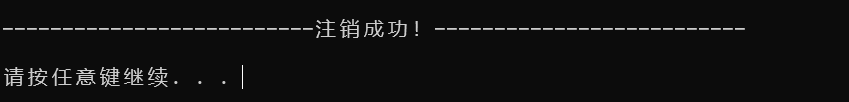


回到学生操作菜单后输入4，进行取消预约的操作，然后输入1

文本

描述已自动生成

回到学生操作菜单后输入\*会回到上一级重新登录

文本

描述已自动生成

回到学生操作菜单后输入0 会直接结束程序

文本

描述已自动生成

2、教师的操作

登录操作与学生的基本一致

登录成功后进入教师菜单及操作页面

文本

描述已自动生成

输入1进行查看所有预约操作

文本

描述已自动生成

输入2进行审核预约操作

文本

描述已自动生成

3、管理员的操作

登录成功后进入管理员菜单及操作界面

文本

描述已自动生成

输入1进行添加账号操作

文本

描述已自动生成

输入1添加学生账号

文本

描述已自动生成

输入2进行添加教师账号操作

文本

描述已自动生成

在管理员操作菜单界面输入2进行查看账号操作

文本

描述已自动生成

输入1进行查看学生账号操作

文本

描述已自动生成

输入2进行查看教师账号操作

文本

描述已自动生成

在管理员操作菜单界面输入3进行查看机房信息操作

文本

描述已自动生成

输入4进行显示所有预约操作

文本

描述已自动生成

输入5进行清空预约操作

文本

描述已自动生成

2、异常数据

在各个界面输入非提示的数据

图片包含 文本

描述已自动生成

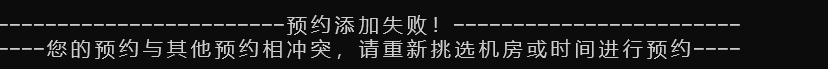
在各个登录界面输入错误的账号密码

文本

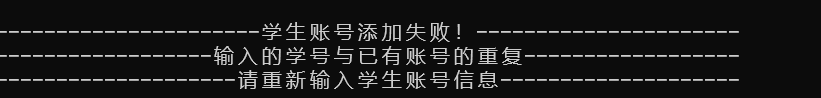
描述已自动生成

在申请预约的时候输入的机房号码、日期、时间段错误

 图片包含 文本

描述已自动生成   

添加账号重复



六、测试情况，设计技巧及体会

1、改进方案

（1）对帐号的密码进行加密。

（2）运用数据库进行账号信息的存储等操作。

2、心得体会

这次课程设计，加强了我们编程、思考、解决问题的能力，巩固和加深了我们对数据结构所学内容的理解，让我们更加了解到数据结构的重要性，让我们对这些知识又有了深刻理解。因为这是一个团队任务，加强了我们的沟通能力，团队合作能力，有利于我们以后的发展。通过这次课程设计使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从理论中得出结论，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。

七、参考文献

《数据结构(C语言版)》严蔚敏\_吴伟民主编 清华大学出版社

八、附录

代码太多不方便粘过来，放在之前的压缩包里。