

课程设计报告

题 目： 职工信息管理系统

课程名称： 面向对象程序设计

学 院： 信息工程学院

专 业： 软 件 工 程

班 级： 18软嵌3班

学生姓名： 顾 震 超

学 号： 20180312305

指导教师： 张 瑜 慧 丁 浩

成 绩：

开课时间： 2018-2019 学年 2 学期

目录

[1、功能介绍](#_Toc7806_WPSOffice_Level1) [1](#_Toc7806_WPSOffice_Level1)

[2、概要设计](#_Toc28701_WPSOffice_Level1) [1](#_Toc28701_WPSOffice_Level1)

[3、详细设计](#_Toc17550_WPSOffice_Level1) [2](#_Toc17550_WPSOffice_Level1)

[3.1声明职工类：](#_Toc9229_WPSOffice_Level2) [2](#_Toc9229_WPSOffice_Level2)

[3.2声明界面类：](#_Toc8358_WPSOffice_Level2) [3](#_Toc8358_WPSOffice_Level2)

[3.3声明数据库类：](#_Toc30733_WPSOffice_Level2) [4](#_Toc30733_WPSOffice_Level2)

[3.4定义主函数](#_Toc28404_WPSOffice_Level2) [4](#_Toc28404_WPSOffice_Level2)

[4、测试结果](#_Toc9434_WPSOffice_Level1) [4](#_Toc9434_WPSOffice_Level1)

[4.1系统登入面](#_Toc15746_WPSOffice_Level2) [4](#_Toc15746_WPSOffice_Level2)

[4.2管理员功能实现](#_Toc11695_WPSOffice_Level2) [6](#_Toc11695_WPSOffice_Level2)

4.3系统功能概括 6

4.[3.1职工信息的录入](#_Toc4305_WPSOffice_Level3) [6](#_Toc4305_WPSOffice_Level3)

4.[3.2职工信息的浏览](#_Toc32675_WPSOffice_Level2) [7](#_Toc32675_WPSOffice_Level2)

4.[3.3职工信息的修改](#_Toc23196_WPSOffice_Level3) [7](#_Toc23196_WPSOffice_Level3)

4.[3.4职工信息的查找](#_Toc6164_WPSOffice_Level3) [8](#_Toc6164_WPSOffice_Level3)

4.[3.5职工信息的删除](#_Toc12399_WPSOffice_Level3) [9](#_Toc12399_WPSOffice_Level3)

4.[3.6职工信息的排序](#_Toc31817_WPSOffice_Level3) [9](#_Toc31817_WPSOffice_Level3)

4.[3.7职工信息的保存](#_Toc14146_WPSOffice_Level3) [10](#_Toc14146_WPSOffice_Level3)

[4.4管理员账号注册](#_Toc11590_WPSOffice_Level3) [11](#_Toc11590_WPSOffice_Level3)

[5、课程设计总结](#_Toc19021_WPSOffice_Level1) [11](#_Toc19021_WPSOffice_Level1)

[6、附录代码](#_Toc29331_WPSOffice_Level1) [14](#_Toc29331_WPSOffice_Level1)

**1、功能介绍**

设计了一个简易的职工信息管理系统，公司的管理者能够通过该系统完成对职工信息的录入，删除，修改，查找，显示，排序等功能。职工的数据信息保存在数据库中，也能够以文件的形式保存在本地。系统具有菜单和提示，界面相对友好。

**2、概要设计**

职工信息管理系统涉及职工、管理员，两个对象，职工和管理员两种角色。为简化系统，本系统只设置了一名管理员，并可在以后任意添加管理员人数，管理员可以管理职工的信息，有对进入该公司的职工进行信息录入，删除，修改，查找，排序，保存等功能。先将职工信息管理系统的功能细化，其功能结构如图1所示。

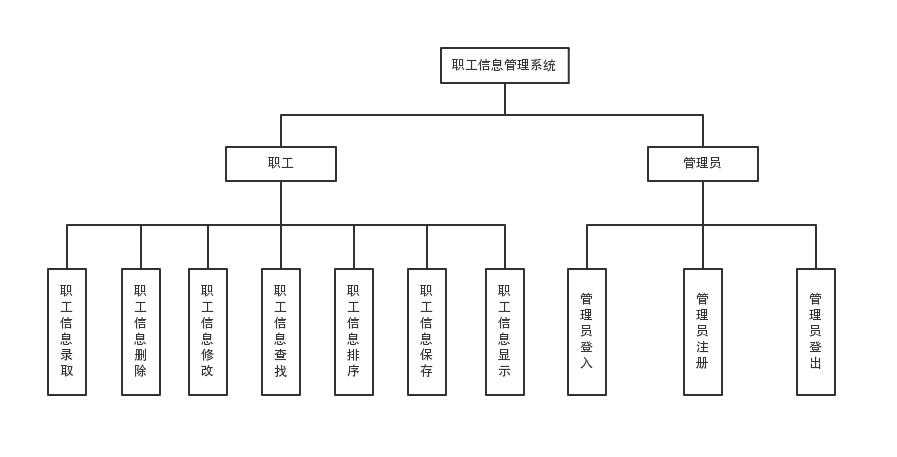
****

图1 职工信息管理系统功能结构图

职工信息的录取：录入职工信息，将职工的信息录入数据库中，按照职工的工号从小到大放入数据库中。

职工信息的删除：删除职工信息，根据职工的姓名删除职工信息。

职工信息的修改：修改职工信息，根据职工的工号进行职工信息的修改。

职工信息的查找：查找职工信息，根据职工的工号，姓名，部门，进行职工信息的查找。

职工信息的排序：排序职工信息，根据职工的工资，部门，年龄，进行职工信息的排序。

职工信息的保存：保存职工信息，将录入数据库中的职工信息保存在本地磁盘之中。

职工信息的显示：显示职工信息，显示录入数据库中所有的职工的信息。

存储数据：可将职工的信息存入数据库，也可以将职工的信息以文件的形式进行存储，放入本地电脑的磁盘之中。

载入数据：将职工的信息从数据库中读出。

登录：只有管理员一种角色。为简化系统，登录只需要输入管理员账号和密码，无需输入其他验证信息，管理员只有一个密码设置也比较简单，同时可以注册多个管理员帐账户。

**3、详细设计**

3.1声明职工类：

class Stuff

{

private:

int id; //工号

int age; //年龄

int salary; //工资

*string* tel; //电话号码

*string* name; //姓名

*string* gender; //性别

*string* department; //部门

*string* degree; //学历

*string* addr; //居住住址

public:

Stuff(); //默认职工信息的构造函数

Stuff(int, int, *string*, *string*, *string*, *string*, *string*, *string*);

//录入数据库的构造函数

void dispInfo(); //显示职工信息

void addStuff(); //录入职工信息

void modifyStuff(); //修改职工信息

void delStuff(); //删除职工信息

void saveStuff(); //保存职工信息

void searchStuff(); //查找职工信息

void sortStuff(); //排序职工信息

int sortSalary(); //根据工资进行排序

int sortDegree(); //根据学历进行排序

int sortAge();

int searchId(); //根据工号进行查找

int searchName(); //根据姓名进行查找

int searchDepartment(); //根据部门进行查找

int searchDegree(int num); //根据学历进行查找

*string* getAge(int num); //获取职工年龄

*string* getTel(int numr); //获取职工电话

*string* getGender(int num); //获取职工性别

*string* getDepartment(int num); //获取职工部门

*string* getDegree(int num); //获取职工学历

};

3.2声明界面类：

class Menu

{

public:

void welcome(); //登入后的欢迎界面

void function\_menu(); //功能菜单界面

void function\_menu\_ctr(); //功能菜单界面控制器

void login\_menu(); //管理员登入界面

void sort\_menu(); //排序菜单界面

void login\_menu\_ctr(); //管理员登入界面控制器

void search\_menu(); //查找菜单界面

void search\_menu\_ctr(); //查找菜单界面控制器

int check\_menu(); //管理员登入信息验证界面

void resign\_up(); //管理员注册界面

};

3.3声明数据库类：

class Mysql

{

public:

*MYSQL* conn;

*MYSQL\_FIELD*\* field;

*MYSQL\_RES*\* result;

*MYSQL\_ROW* row;

*MYSQL\_ROW* mysql\_fetch(const char\* sql);

int mysqConnect();

*string* put\_field(int num);

};

3.4定义主函数

int main(void)

{

Menu me;

me.login\_menu();

me.login\_menu\_ctr();

}

**4、测试结果**

4.1系统登入面

系统运行时的登入界面如图 2 所示，本系统都有默认的管理员账户。在图 2 中如果需要自行注册可以选的“管理员注册”，然后便可创建独有的管理员账户，选择“退出系统”后，程序会被停止，从而结束进程。选择“管理员登入”后进入下一个登入验证界面如图3所示

只有输入相应的字母，如“A或a”，便可进入相应的一步。



图2 系统登入界面

进入图3 登入验证界面后输入管理员账号和密码后便可进入职工信息管理系统主界面。

如图4 管理系统主界面。如果输出的账号或者密码时有误的，则会提示您输出的账户信息有误，并会跳回 如图2系统登入界面。如图5提示账户有误。

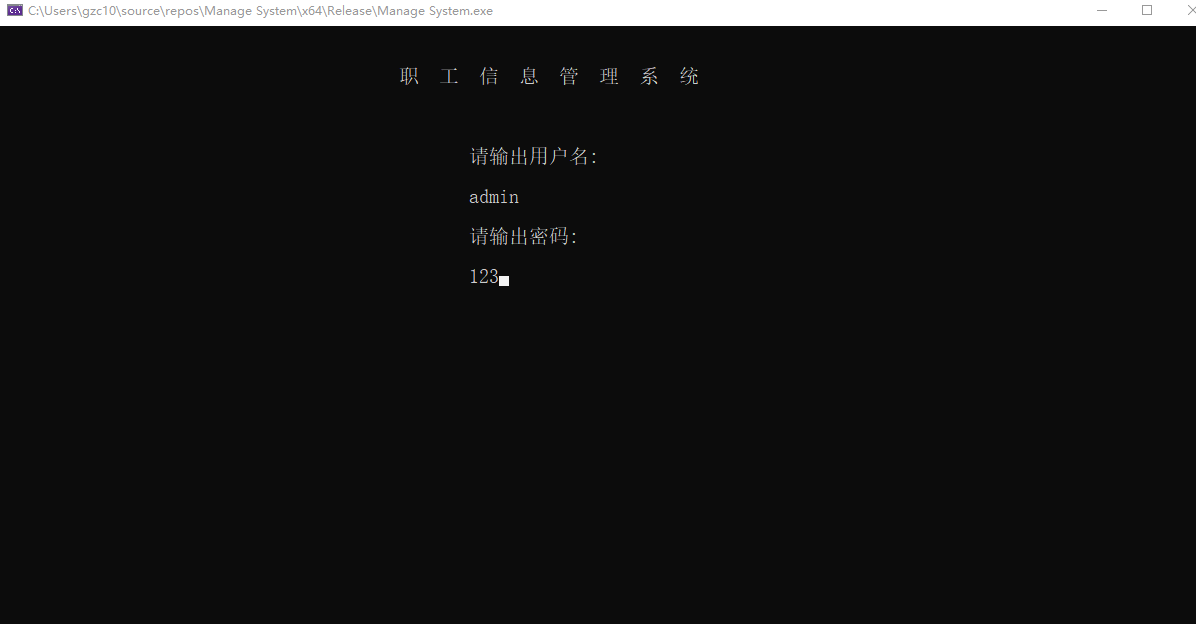


图3 登入验证界面

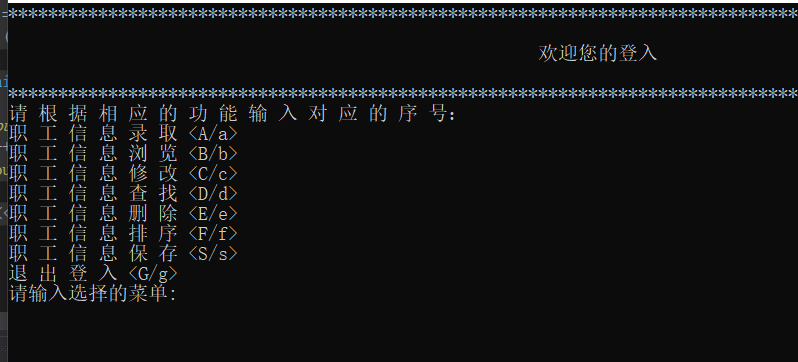


图4 系统主界面

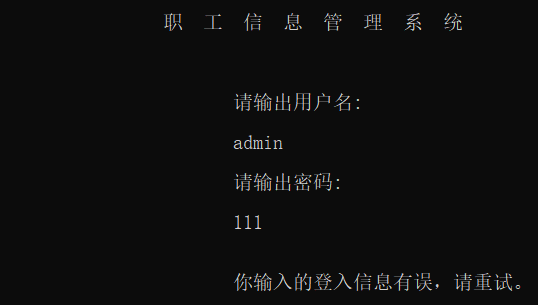


图5 提示账户有误

4.2管理员功能实现

如图5系统主界面,在图 2 中选择管理员登录，即输入“A”或“a”，正确输出管理员账号后，进入如图 4 所示的界面中，管理员可以根据需要实现“职工信息录取、职工信息浏览、职工信息修改、职工信息查找、职工信息删除、职工信息排序，职工信息保存、退出登入”等功能。

4.3系统功能概括

4.3.1职工信息的录入

在图4中按“A或a”选择第一项，即可进入职工信息的录取功能之中，如图6所示，按照提示管理员输入所要录入的信息，知道出现“成功录入”的提示后，则说明，该为职工的信息成功被录入数据库中。

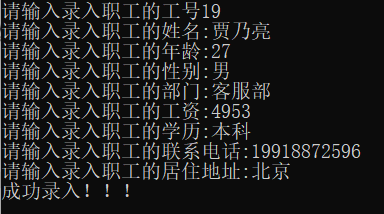


图6职工信息的录取

4.3.2职工信息的浏览

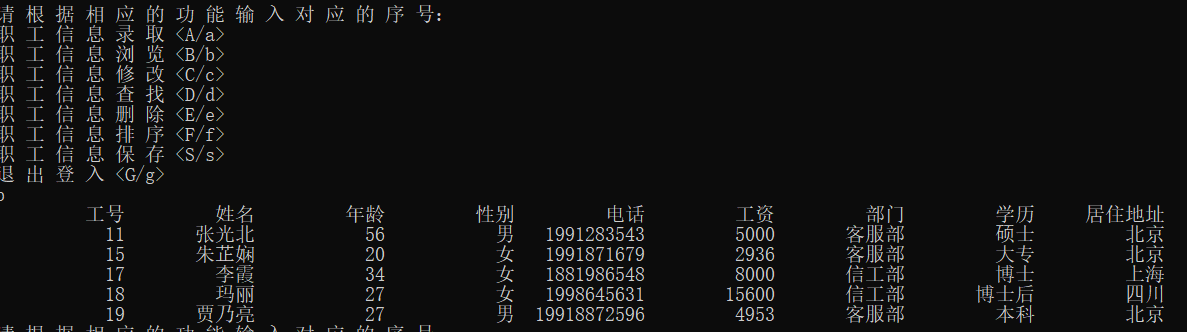
在图4中按“B或b”选择第二项，即可进入职工信息的浏览功能之中，如图7所示，按照提示，管理员输入字符后，则可查看所有的职工信息，工号为19的职工，便是上面职工信息录入功能录入的职工。

图7 职工信息的浏览

4.3.3职工信息的修改

在图4中按“C或c”选择第三项，即可进入职工信息的修改功能之中，在图7中，以工号为11的职工为例，修改其的年龄和居住地址，如图8，确认修改的项输入‘yes‘无需修改的则输入’no‘，最后会显示修改的项数。如图9职工信息修改所示，工号为10的职工的信息被修改了。

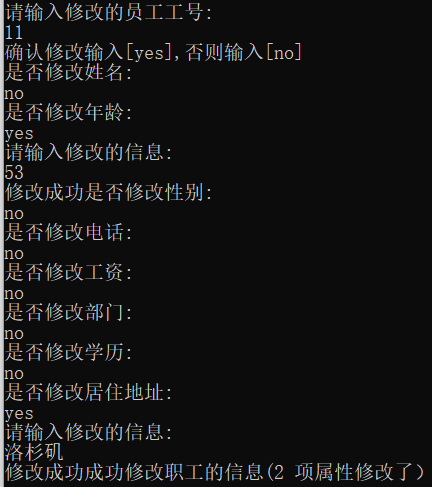


图8职工信息修改

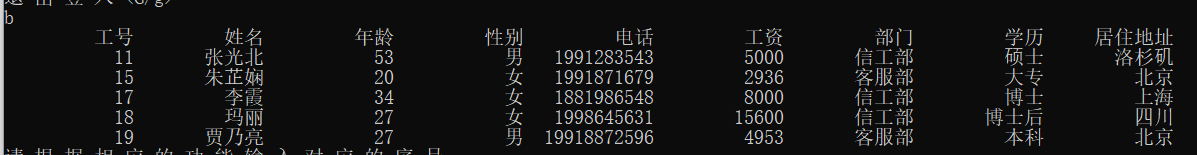


图9职工信息修改

4.3.4职工信息的查找

在图4中按“D或d”选择第四项，即可进入职工信息的查找功能之中，如图10所示，查找功能提供了三种查找方式，分别为工号查找，姓名查找，部门查找，输入4后可以从该菜单回到图4的系统主菜单中。如图11，所示的查找结果。

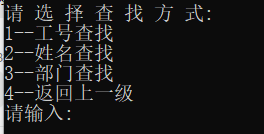
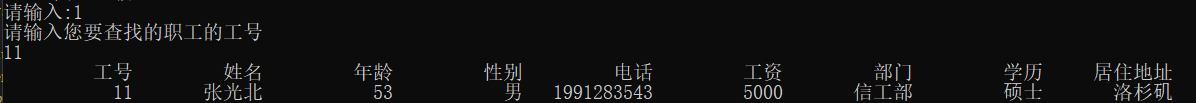
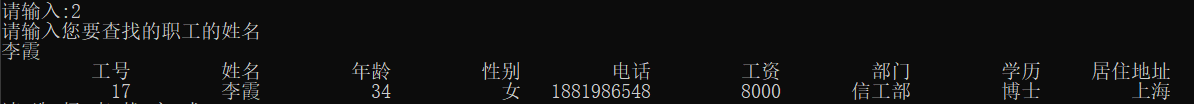


图10职工信息查找





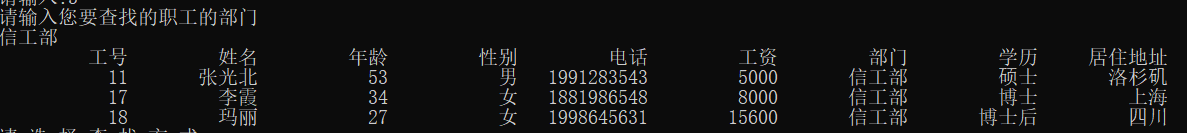


图11查找结果

4.3.5职工信息的删除

在图4中按“E或e”选择第五项，即可进入职工信息的删除功能之中，如图12职工信息删除所示，按照提示输入需要删除职工的姓名，即可删除所想要删除的职工的姓名。如果显示删除成功，则说明该职工的信息被成功删除了。

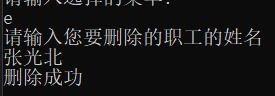


图12职工信息删除

如下图18，浏览职工信息时，则可以发现该职工被删除。

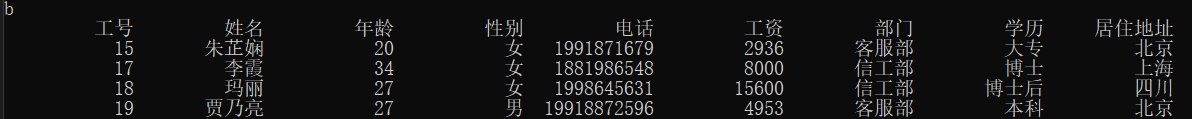


图18浏览结果

4.3.6职工信息的排序

在图4中按“F或f”选择第六项，即可进入职工信息的排序功能之中，如图13所示，进入职工排序功能的选择菜单，该菜单中管理员可以进行对职工信息采取工资，学历，年龄等信息的排序，其中每一项可以进行升序和降序的排序浏览。

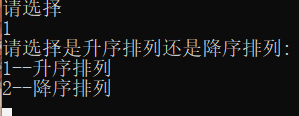
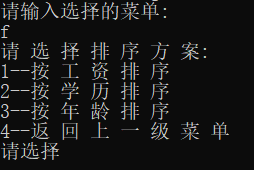
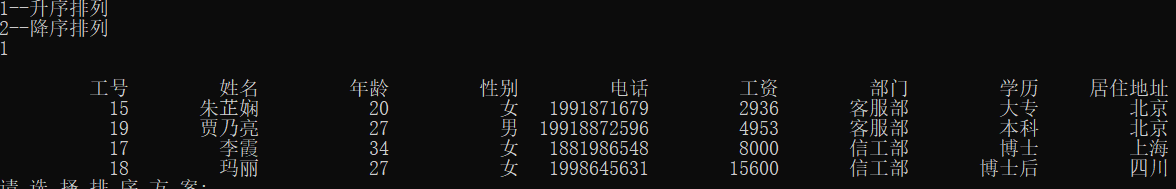


图13职工信息排序

此处为了节省篇幅，只进行按工资的升序和降序演示，效果如图14所示，职工的信息按工资进行输出。



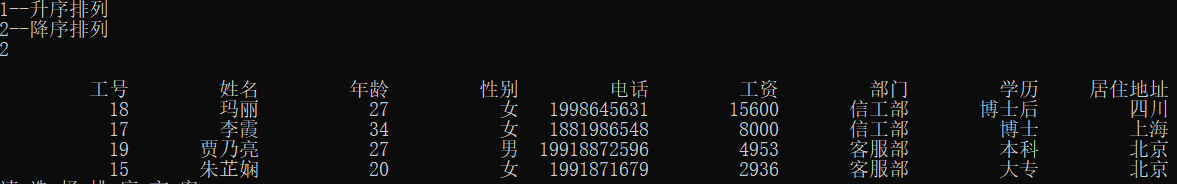


图14职工信息的按工资的升序和降序演示

4.3.7职工信息的保存

在图4中按“S或s”选择第七项，即可进入职工信息的保存功能之中，首先浏览一下数据库中存储的信息，选择“职工信息浏览”功能，如图15所示。

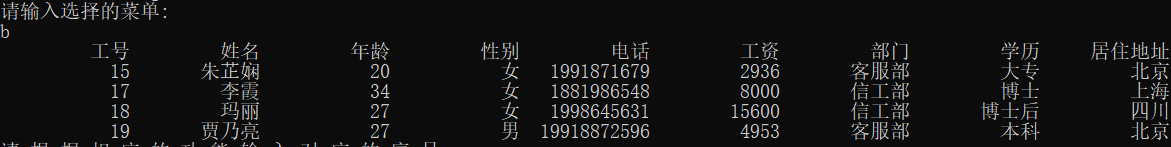


图15

之后选择“职工信息保存”功能，如图提示保存成功，则文件成功被保存在本地磁盘之中，如图16所示，

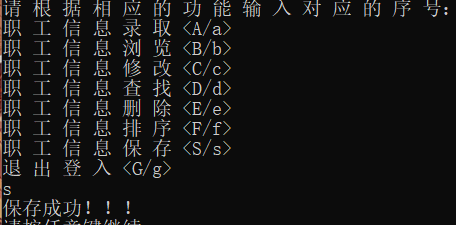
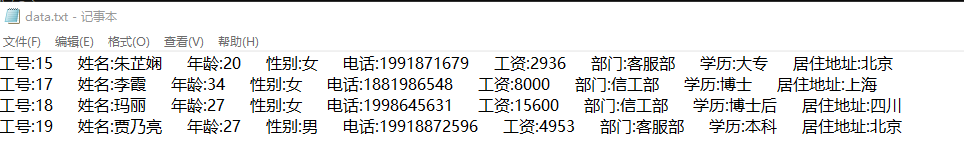


图16职工信息保存

如图17，职工的信息被成功以文本文件的形式存在本地电脑磁盘之中。



4.4管理员账号注册

在图2管理员登入界面，选择“管理员注册”，如图17所示，管理者输入新的管理员账号，后便可以注册新的管理员账户。



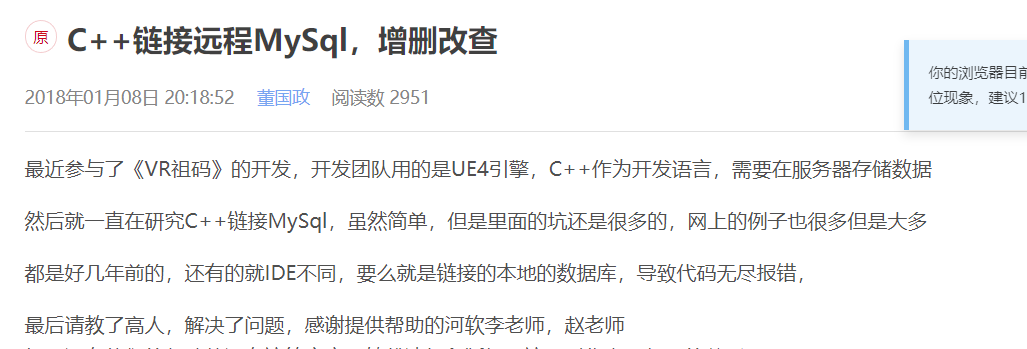
图17管理员账号注册

**5、课程设计总结**

在此系统中运用到了MySql数据库，由于从未在C++中使用过，因此查询了相关信息。在使用MySql数据库是，遇到了如下几个问题。

1. 在VS2019中将数据库连接起来
2. 调用数据库的相关函数
3. 使用数据库是其类的制作

首先，对于第一个问题，首先感谢学长提供的帮助，提供了VS2019连接MySql数据库的资源，同时感谢网友



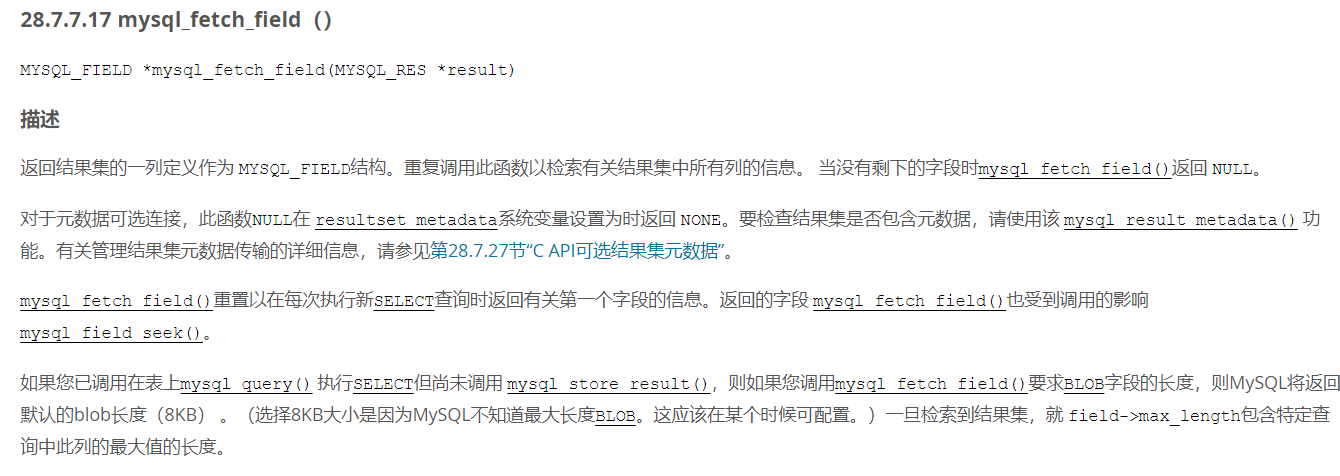
在CSDN上面的经验分享，帮助我解决了，最开始，也是最大的问题。

其次，对于第二个问题，个人有过使用PHP写过网页后端脚本的经验，使用过MySql数据库，但对于如何在C++中操作数据库，并没有经验，一些简单的使用由网友提供了，之后其他的函数如图下



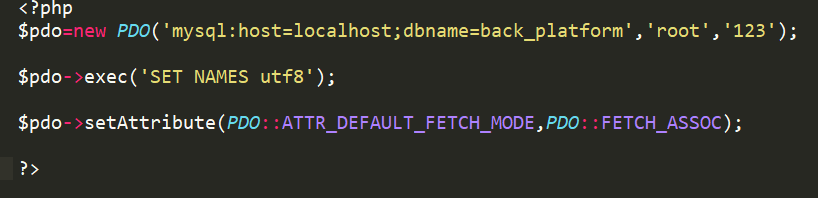
该函数用于获取数据库表中的字段，该函数的使用方法来自于MySql官方的使用手册，下面是使用的函数描述的网页

<https://dev.mysql.com/doc/refman/8.0/en/c-api-functions.html>

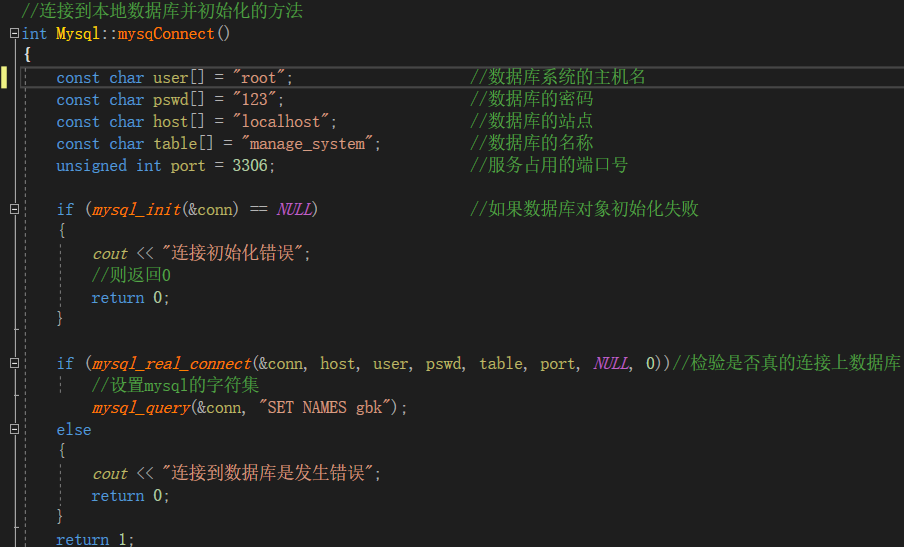


其中还在mysql\_fetch\_row();函数上遇到了问题，一开是把它作为循环条件，在循环体里一次次把数据给取出来，官方手册的说法是每次取出一条记录，之后再将其指向下一条，如果没有，则返回NULL。

最后，对于第三个问题，本课设是C++程序设计，因此进行对MySql数据库功能的包装，将其试着做成一个类，但是一个没有很好的方法，先参照之前过的PHP代码，把MySql数据库功能做成头文件，但是发现不是很好，到处都出现了报错，最后将其做成了类，通过声明该类的对象来进行数据库的调用，发现结果很好。然后就继续使用该方法。实际上，做出的MySql类还是存在问题，主要是个人缺乏相关的经验，这一次便是成了试手。关于参照的PHP代码如下图



不过在C++的代码中还是要进行修改，以下是现在用的



**6、附录代码**

6.1 class.h

#include<string>

#include <mysql.h>

#include<iostream>

using namespace *std*;

#ifndef CLASS\_H\_

#define CLASS\_H\_

class Stuff

{

private:

int id; //工号

int age; //年龄

int salary; //工资

*string* tel; //电话号码

*string* name; //姓名

*string* gender; //性别

*string* department; //部门

*string* degree; //学历

*string* addr; //居住住址

public:

Stuff(); //默认职工信息的构造函数

Stuff(int, int, *string*, *string*, *string*, *string*, *string*, *string*);

//录入数据库的构造函数

void dispInfo(); //显示职工信息

void addStuff(); //录入职工信息

void modifyStuff(); //修改职工信息

void delStuff(); //删除职工信息

void saveStuff(); //保存职工信息

void searchStuff(); //查找职工信息

void sortStuff(); //排序职工信息

int sortSalary(); //根据工资进行排序

int sortDegree(); //根据学历进行排序

int sortAge(); //根据年龄进行排序

int searchId(); //根据工号进行查找

int searchName(); //根据姓名进行查找

int searchDepartment(); //根据部门进行查找

int searchDegree(int num); //根据学历进行查找

*string* getAge(int num); //获取职工年龄

*string* getTel(int numr); //获取职工电话

*string* getGender(int num); //获取职工性别

*string* getDepartment(int num); //获取职工部门

*string* getDegree(int num); //获取职工学历

};

class Menu

{

public:

void welcome(); //登入后的欢迎界面

void function\_menu(); //功能菜单界面

void function\_menu\_ctr(); //功能菜单界面控制器

void login\_menu(); //管理员登入界面

void sort\_menu(); //排序菜单界面

void login\_menu\_ctr(); //管理员登入界面控制器

void search\_menu(); //查找菜单界面

void search\_menu\_ctr(); //查找菜单界面控制器

int check\_menu(); //管理员登入信息验证界面

void resign\_up(); //管理员注册界面

};

class Mysql

{

public:

*MYSQL* conn; //创建数据库连接对象

*MYSQL\_FIELD*\* field; //返回字段的指针

*MYSQL\_RES*\* result; //返回指令执行结果的指针

*MYSQL\_ROW* row; //返回从数据库取出的数据

*MYSQL\_ROW* mysql\_fetch(const char\* sql); //数据输出函数

int mysqConnect(); //数据库连接初始化函数

*string* put\_field(int num); //返回字段的函数

};

void pause\_out(*string*); //动态输出函数

#endif

6.2defination.cpp

#include <iostream>

#include <fstream>

#include<windows.h>

#include<cstring>

#include <mysql.h>

#include <cstdio>

#include <iomanip>

#include "class.h"

using namespace std;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Stuff\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Stuff\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Stuff\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

Stuff::Stuff() //Stuff对象的默认初始化

{

id = 0;

age = 0;

tel = "0";

salary = 0;

name = "Unknown";

gender = "Unknown";

department="no belonging";

degree="Unknown";

addr = "Unknown";

}

//Stuff对象的初始化，获得工号，姓名，电话，性别，学位，地址和年龄

Stuff::Stuff(int id\_, int age\_, string tel\_, string name\_, string gender\_, string degree\_, string addr\_,string department\_)

{

id = id\_;

age = age\_;

tel = tel\_;

salary = 0;

name = name\_;

gender = gender\_;

department = department\_;

degree = degree\_;

addr = addr\_;

}

//显示职工的信息

//将从数据库中获取的数据输出

void Stuff::dispInfo()

{

Menu me;

Mysql mys;

mys.mysqConnect(); //调用连接到mysql数据库的连接初始化方法

mys.mysql\_fetch("select \* from info");

//调用Mysql的数据信息获取方法

cin.get();

me.function\_menu();

}

//录入职工的信息

void Stuff::addStuff()

{

Mysql mys;

Stuff worker;

Menu me;

cout << "请输入录入职工的工号";

string id\_;

cin >> id\_;

if (id\_ == "q") //输入q时退出功能菜单界面

{

system("cls");

me.welcome();

me.function\_menu();

me.function\_menu\_ctr();

}

else

worker.id = stoi(id\_); //将字符串转为int类型

cout << "请输入录入职工的姓名:";

cin >> worker.name;

cout << "请输入录入职工的年龄:";

cin >> worker.age;

cout << "请输入录入职工的性别:";

cin >> worker.gender;

cout << "请输入录入职工的部门:";

cin >> worker.department;

cout << "请输入录入职工的工资:";

cin >> worker.salary;

cout << "请输入录入职工的学历:";

cin >> worker.degree;

cout << "请输入录入职工的联系电话:";

cin >> worker.tel;

cout << "请输入录入职工的居住地址:";

cin >> worker.addr;

//组合调用数据库的指令，将变量中的值，写入sql语句之中

char sql[300];

sprintf\_s(sql,300,"insert into info(name,age,gender,tel,salary,department,degree,addr) values('%s','%d','%s','%s','%d','%s','%s','%s')",

(worker.name).data(), worker.age, (worker.gender).data(), (worker.tel).data(), worker.salary, (worker.department).data(), (worker.degree).data(), (worker.addr).data());

//data()函数将string类型转换为char类型的字符串类型

mys.mysqConnect();

mysql\_query(&(mys.conn), sql);

pause\_out("成功录入！！！");

cout << endl;

pause\_out("请继续您的操作:");

me.function\_menu();

}

//修改职工的信息

void Stuff::modifyStuff()

{

int id;

Menu me;

string name;

string key;

Mysql mys;

mys.mysqConnect();

char sql[] = "select \* from info";

mysql\_query(&(mys.conn), sql); //执行所输出的sql语句的命令

mys.result = mysql\_store\_result(&(mys.conn)); //将执行的结果返回给mys.result

mys.field = mysql\_fetch\_field(mys.result); //获取的存放在数据库表中的字段

pause\_out("请输入修改的员工工号:");

cin >> id;

pause\_out("确认修改输入[yes],否则输入[no]");

string head = "是否修改";

int q = 0;

for (int i=1;i<9;)

{

string tail = (mys.put\_field(i)); //put\_field里面存储这字段

string whole = head + tail+":"; //通过循环将不同字段组合成同一句话

pause\_out(whole); //然后输出，从而减少代码量

cin >> key;

if (key == "yes")

{

pause\_out("请输入修改的信息:");

cin >> name;

mys.mysqConnect();

char sql[50];

sprintf\_s(sql, 50, "update info set %s='%s' where id=%d",mys.field[i].name ,(name).data(), id);

mysql\_query(&(mys.conn), sql);

cout << "修改成功";

q++; //如果修改的话则可以进入下一次循环

i++; //从而修改下一个字段，否则持续同一个循环

}

else

{

if (key != "no") //此为非yes和非no的异常输出的对应

pause\_out("\*\*\*\*\*请输入[yes]或者[no]\*\*\*\*\*");

else

i++; //key为no时，不对该项字段进行修改

}

}

cout << "成功修改职工的信息" << "(" << q << " 项属性修改了）"<<endl<<endl;//如果修改了一项则q+1，可以显示修改的项数

me.function\_menu();

}

//删除职工的信息

void Stuff::delStuff()

{

Menu me;

Mysql mys;

string name;

pause\_out("请输入您要删除的职工的姓名");

cin >> name;

char sql[50];

sprintf\_s(sql, 50, "delete from info where name='%s'", name.data());

if (name != "q")

{

mys.mysqConnect();

if (!(mysql\_query(&(mys.conn), sql)))//成功归零。如果发生错误，则为非零

{ //摘自Mysql手册

cin.get();

cout << "删除成功" << endl;

cin.get();

system("cls");

me.welcome();

me.function\_menu();

}

}

else

{

cin.get();cin.get();

system("cls");

me.welcome();

me.function\_menu();

}

}

//保存职工信息

void Stuff::saveStuff()

{

Mysql mys;

Menu me;

mys.mysqConnect();

mysql\_query(&mys.conn, "select \* from info");

mys.result = mysql\_store\_result(&mys.conn);

ofstream fout;

fout.open("C:\\Users\\gzc10\\Desktop\\data.txt"); //注意‘\\’的规则，不然路径会出现问题

while (mys.row = mysql\_fetch\_row(mys.result))

{

string fields;

for (int i = 0; i < 9; i++)

{

fields += mys.put\_field(i)+":"; //将多个字段拼起来组合成一条记录

fields += (mys.row[i]); //通过string类的重载实现

fields += " ";

}

fields += "\n"; //加上\n可以让一条记录独占一行

fout << fields; //写入创建的文本文档之中

}

fout.close();

pause\_out("保存成功！！！");

system("pause");

system("cls");

me.welcome();

me.function\_menu();

}

//分为id和姓名和部门查找

void Stuff::searchStuff()

{

Menu me;

me.search\_menu(); //调用查找菜单的界面和选择器

me.search\_menu\_ctr(); //选择器中包含了查找的方法

}

//根据职工工号进行查找

int Stuff::searchId()

{

int id;

pause\_out("请输入您要查找的职工的工号");

cin >> id;

//输出要查找的信息的姓名

Mysql mys;

mys.mysqConnect();

char sql[80];

sprintf\_s(sql, 80, "select \* from info where id=%d", id);

//通过sprintf\_s函数的功能将sql指令组合起来

mysql\_query(&mys.conn, sql); //执行mysql命令

mys.result = mysql\_store\_result(&mys.conn);

if (mysql\_fetch\_row(mys.result)) //如果获取到内容，则结果为非空

{ //否则返回NULL

for (int i = 0; i < 9; i++)

cout << setw(13) << mys.put\_field(i);

cout << endl;

mysql\_query(&mys.conn, sql);

mys.result = mysql\_store\_result(&mys.conn);

//判断语句的条件用掉了第一次存储的数据

while (mys.row = mysql\_fetch\_row(mys.result))

//此处的目的时重新获取一次成功取得的内容

{

for (int i = 0; i < 9; i++)

{

cout << setw(13) << mys.row[i];

}

cout << endl;

}

return 1;

}

else

pause\_out("抱歉，没有该员工的信息。");

return 0;

}

//根据职工姓名进行查找

int Stuff::searchName()

{

string name;

pause\_out("请输入您要查找的职工的姓名");

cin >> name;

//输出要查找的信息的姓名

Mysql mys;

mys.mysqConnect();

char sql[80];

sprintf\_s(sql, 80, "select \* from info where name='%s'", (name).data());

//通过sprintf\_s函数的功能将sql指令组合起来

mysql\_query(&mys.conn, sql); //执行mysql命令

mys.result = mysql\_store\_result(&mys.conn);

if (mysql\_fetch\_row(mys.result)) //如果获取到内容，则结果为非空

{ //否则返回NULL

for (int i = 0; i < 9; i++)

cout << setw(13) << mys.put\_field(i);

cout << endl;

mysql\_query(&mys.conn, sql);

mys.result = mysql\_store\_result(&mys.conn);

//判断语句的条件用掉了第一次存储的数据

while (mys.row = mysql\_fetch\_row(mys.result))

//此处的目的时重新获取一次成功取得的内容

{

for (int i = 0; i < 9; i++)

{

cout << setw(13) << mys.row[i];

}

cout << endl;

}

return 1;

}

//如果没有获取则为NULL执行此处的语句

else

pause\_out("抱歉，没有该员工的信息。");

return 0;

}

//根据职工部门进行查找

int Stuff::searchDepartment()

{

string department;

pause\_out("请输入您要查找的职工的部门");

cin >> department;

//输出要查找的信息的部门

Mysql mys;

mys.mysqConnect();

char sql[80];

sprintf\_s(sql, 80, "select \* from info where department='%s'", (department).data());

//通过sprintf\_s函数的功能将sql指令组合起来

mysql\_query(&mys.conn, sql);

mys.result = mysql\_store\_result(&mys.conn); //此处的语句和上一个函数的用法相同

if (mysql\_fetch\_row(mys.result))

{

for (int i = 0; i < 9; i++)

cout << setw(13) << mys.put\_field(i);

cout << endl;

mysql\_query(&mys.conn, sql);

mys.result = mysql\_store\_result(&mys.conn);

while (mys.row = mysql\_fetch\_row(mys.result))

{

for (int i = 0; i < 9; i++)

{

cout << setw(13) << mys.row[i];

}

cout << endl;

}

return 1;

}

else

pause\_out("抱歉，没有该员工的信息。");

return 0;

}

//排序职工信息

void Stuff::sortStuff()

{

Menu me;

me.sort\_menu();

}

//根据工资进行排序

int Stuff::sortSalary()

{

Mysql mys;

Menu me;

int option;

if(!mys.mysqConnect())

return 0 ;

pause\_out("请选择是升序排列还是降序排列:");

pause\_out("1--升序排列");

pause\_out("2--降序排列");

cin >> option;

cout << endl;

switch (option)

{

case 1:

mys.mysql\_fetch("select \* from info order by salary asc");

//通过mysql的指令实现升序排序

break;

case 2:

mys.mysql\_fetch("select \* from info order by salary desc");

//通过mysql的指令实现降序排序

break;

default:

break;

}

me.sort\_menu();

pause\_out("请继续操作:");

return 1;

}

//根据学历进行排序

int Stuff::sortDegree()

{

Mysql mys;

Menu me;

int option;

pause\_out("请选择是升序排列还是降序排列:");

pause\_out("1--升序排列");

pause\_out("2--降序排列");

cin >> option; //输入选择

cout << endl;

for (int i = 0; i < 9; i++)

cout << setw(10) << mys.put\_field(i);

cout << endl;

switch (option)

{

case 1:

for (int k = 0; k < 9; k++)

searchDegree(k); //searchDegree函数中保存这从第到高的学历顺序

//此处时从低到高调用

break;

case 2:

for (int j = 8; j >=0; j--) //重复调用该函数，实现一种选择性的排序

searchDegree(j); //从而解决问题，去升序的方法类似

//此处时从高到底调用

break;

default:

sortDegree();

break;

}

me.sort\_menu();

return 1;

}

//根据年龄进行排序

int Stuff::sortAge()

{

Mysql mys;

Menu me;

int option;

if (!mys.mysqConnect())

return 0;

pause\_out("请选择是升序排列还是降序排列:");

pause\_out("1--升序排列");

pause\_out("2--降序排列");

cin >> option;

if (option == 1)

{

const char\* sql = "select \* from info order by age asc";

//使用mysql指令实现升序排序

mys.mysql\_fetch(sql);

me.sort\_menu();

}

else

{

if (option == 2)

{

const char\* sql = "select \* from info order by age desc";

//使用mysql指令实现降序排序

mys.mysql\_fetch(sql);

me.sort\_menu();

}

else

{

pause\_out("请输入正确的序号:");

me.sort\_menu();

}

}

pause\_out("请继续操作:");

return 1;

}

int Stuff::searchDegree(int num)

{

Mysql mys;

mys.mysqConnect();

const char\* sql = "select \* from info";

mysql\_query(&(mys.conn), sql);

mys.result = mysql\_store\_result(&(mys.conn));

while (mys.row = mysql\_fetch\_row(mys.result))

{

if (mys.row[7] ==getDegree(num)) //row[7]中存储的是学历

//getDegree()中存储的是不同的学历

{

for (int i = 0; i < 9; i++)

//如果判断成立则则将该条字段输出

cout <<setw(10)<< mys.row[i];

cout << endl;

}

}

return 1;

}

//用于输出学历

string Stuff::getDegree(int num)

{

char degree[9][10] = { "小学","初中","中专","高中","大专","本科","硕士","博士","博士后" };

return degree[num];

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Stuff\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Stuff\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Stuff\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

//显示欢迎界面

void Menu::welcome()

{

for (int i = 0; i < 120; i++)

//通过循环达到一种连续输出

cout << "\*";

cout << endl << endl;

cout <<setw(65)<< "欢迎您的登入"<<endl<<endl;

for (int i = 0; i < 120; i++)

cout << "\*";

cout << endl;

}

//显示功能菜单

void Menu::function\_menu()

{

pause\_out("请 根 据 相 应 的 功 能 输 入 对 应 的 序 号：");

pause\_out("职 工 信 息 录 取 <A/a>");

pause\_out("职 工 信 息 浏 览 <B/b>");

pause\_out("职 工 信 息 修 改 <C/c>");

pause\_out("职 工 信 息 查 找 <D/d>");

pause\_out("职 工 信 息 删 除 <E/e>");

pause\_out("职 工 信 息 排 序 <F/f>");

pause\_out("职 工 信 息 保 存 <S/s>");

pause\_out("退 出 登 入 <G/g>");

}

//功能界面选择控制器

void Menu::function\_menu\_ctr()

{

pause\_out("请输入选择的菜单:");

Stuff worker;

char key;

cin >> key;

while (key != 'q') //通过一个key值判断的循环，来实现一个功能的连续

{ //输入q时强制中断循环退出界面

key = tolower(key);

switch (key)

{

case 'a':

worker.addStuff(); //进入信息录入的函数

break;

case 'b':

worker.dispInfo(); //输出所有信息的函数

break;

case 'c':

worker.modifyStuff(); //修改信息的函数

break;

case 'd':

worker.searchStuff(); //查找信息的函数

break;

case 'e':

worker.delStuff(); //删除信息的函数

break;

case 'f':

worker.sortStuff(); //排序的函数

break;

case 's':

worker.saveStuff(); //保存信息的函数

break;

case 'g': //退出登入

system("cls"); //清屏后显示登入界面

login\_menu();

login\_menu\_ctr();

break;

default:

pause\_out("请输入正确的菜单号"); //此处时处理不规范输入的选择

function\_menu();

}

cin >> key;

}

}

//显示登入界面

void Menu::login\_menu()

{

cout << endl << endl;

cout << setw(70)<<"职 工 信 息 管 理 系 统";

cout << endl<< endl << endl << endl;

cout << setw(62) << "管理员登入<A/a>" << endl << endl;

cout << setw(62) << "管理员注册<B/b>" << endl << endl;

cout << setw(60) << "退出系统<C/c>"<<endl;

cout << endl << endl << endl << endl << endl;

cout << setw(100) << "18 软嵌3班 顾震超制作" << endl << endl << endl << endl;

cout << endl << endl << endl << endl << endl << endl << endl << endl;

cout << "注：当您想要停止所选定的功能时，请按‘q’便会放弃该次操作环节"<<endl;

}

//登入界面选择控制器

void Menu::login\_menu\_ctr()

{

char key;

cin >> key;

//将大小写都转化为小写，从而解决大小的统一性问题

key = tolower(key);

while (key!='q')

{

switch (key)

{

case 'a':

system("cls");

//登入成功的话返回1，失败则返回0从而决定是否进入系统

if (check\_menu())

{

system("cls");

welcome();

function\_menu();

function\_menu\_ctr();

}

else //失败的情况执行下列语句，重新回答登入界面

system("cls");

login\_menu();

break;

case 'b':

system("cls");

//进行用户注册

resign\_up();

break;

case 'c':

//退出登入系统

pause\_out("感谢您的使用！！！");

exit(0);

//exit的功能是终止程序的运行，参数0便是正常结束

break;

default:

//输出异常的输入

cout << "请输入正确的序号";

break;

}

cin >> key;

}

}

//显示排序菜单

void Menu::sort\_menu()

{

char key;

Stuff worker;

pause\_out("请 选 择 排 序 方 案:");

pause\_out("1--按 工 资 排 序");

pause\_out("2--按 学 历 排 序");

pause\_out("3--按 年 龄 排 序");

pause\_out("4--返 回 上 一 级 菜 单");

pause\_out("请选择");

cin >> key;

while (key != 'q') //通过key实现留在该级的逗留

{

switch (key)

{

case '1':

worker.sortSalary(); //根据工资排序的函数

break;

case '2':

worker.sortDegree(); //根据学历排序的函数

break;

case '3':

worker.sortAge(); //根据年龄排序的函数

break;

case '4': //返回到上一级

system("cls");

welcome();

function\_menu();

function\_menu\_ctr();

break;

default:

cout << "请输入正确的序号";

}

cin >> key;

}

}

//检验登入界面

//该函数返回一个int类型的值，来反映是否找到数据库中的该用户

int Menu::check\_menu()

{

string username; //登入的用户名

string password; //登入的密码

cout << endl << endl;

cout << setw(70) << "职 工 信 息 管 理 系 统";

cout << endl << endl << endl << endl;

cout << setw(60) << "请输出用户名:" << endl << endl;

cout << " " ;

cin >> username;

cout << endl;

cout << setw(58) << "请输出密码:" << endl << endl;

cout << " ";

cin >> password;

Mysql mys;

mys.mysqConnect();

const char\* sql = "select \* from user";

mysql\_query(&mys.conn, sql);

//从数据库中获取用户名和密码

mys.result = mysql\_store\_result(&mys.conn);

//对每一条用户名和密码进行比对，如果找到则返回1

//否侧全部比对饭后没有的话则返回0

while (mys.row = mysql\_fetch\_row(mys.result))

{

if (mys.row[1] == username)

if (mys.row[2] == password)

return 1;

}

//结束全部比对后，仍然没有匹配的，则执行下列语句

cout << setw(48)<< endl << endl;

pause\_out("你输入的登入信息有误，请重试。");

system("pause");

return 0;

}

//显示查找菜单界面

void Menu::search\_menu()

{

pause\_out("请 选 择 查 找 方 式:");

pause\_out("1--工号查找");

pause\_out("2--姓名查找");

pause\_out("3--部门查找");

pause\_out("4--返回上一级");

cout << "请输入:";

}

//查找菜单界面控制器

void Menu::search\_menu\_ctr()

{

Menu me;

Stuff worker;

char option;

cin >> option;

while (option!='q') //通过option实现逗留在该级

{

switch (option)

{

case '1':

worker.searchId(); //根据工号查找的函数

break;

case '2':

worker.searchName(); //根据姓名查找的函数

break;

case '3':

worker.searchDepartment(); //根据部门查找的函数

break;

case '4': //返回上一级菜单

system("cls");

me.welcome();

me.function\_menu();

me.function\_menu\_ctr();

break;

default:

pause\_out("请输入正确的序号:");

break;

}

me.search\_menu();

cin >> option;

}

}

//用户注册界面

void Menu::resign\_up()

{

Mysql mys;

string username; //注册的用户名

string password; //注册的密码

string repassword; //注册的密码验证

cout << endl << endl;

cout << setw(76) << "职 工 信 息 管 理 系 统 注 册";

cout << endl << endl << endl << endl;

cout << setw(60) << "请输出用户名:" << endl << endl;

cout << " ";

cin >> username;

cout << endl;

cout << setw(58) << "请输出密码:" << endl << endl;

cout << " ";

cin >> password;

cout << endl;

cout << setw(62) << "请再次输出密码:" << endl << endl;

cout << " ";

cin >> repassword;

if (password==repassword)

//如果密码验证成功的话执行将用户信息存入数据库的指令

{

char sql[100];

sprintf\_s(sql, 100,"insert into user(username,password) values('%s','%s')", (username).data(), (password).data());

//通过sprintf\_s把变量的值和sql语句组合在一起

//初始化数据库

mys.mysqConnect();

//执行sql指令

mysql\_query(&mys.conn, sql);

//指令执行成功返回零，该处运用的是该函数的返回值的相反数

if (!mysql\_query(&mys.conn, sql)) {

pause\_out("注册成功");

system("pause");

}

else

{

pause\_out("注册失败");

system("pause");

}

system("cls");

login\_menu();

login\_menu\_ctr();

}

else

pause\_out("输出的信息有误");

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Mysql\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Mysql\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Mysql\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

//连接到本地数据库并初始化的方法

int Mysql::mysqConnect()

{

const char user[] = "root"; //数据库系统的用户名

const char pswd[] = "123"; //数据库的密码

const char host[] = "localhost"; //数据库的站点

const char table[] = "manage\_system"; //数据库的名称

unsigned int port = 3306; //服务占用的端口号

if (mysql\_init(&conn) == NULL) //如果数据库对象初始化失败

{

cout << "连接初始化错误";

//则返回0

return 0;

}

if (mysql\_real\_connect(&conn, host, user, pswd, table, port, NULL, 0))//检验是否真的连接上数据库

//设置mysql的字符集

mysql\_query(&conn, "SET NAMES gbk");

else

{

cout << "连接到数据库是发生错误";

return 0;

}

return 1;

}

//获取数据库中的职工们的信息

//该函数接受一个const char\* 类型的参数，返回 MYSQL\_ROW的对象

MYSQL\_ROW Mysql::mysql\_fetch(const char\* sql)

{

//MYSQL\_ROW row;

mysql\_query(&conn, sql);

result = mysql\_store\_result(&conn);

for (int i = 0; i < 9; i++)

cout << setw(13) << put\_field(i);

cout << endl;

while (row = mysql\_fetch\_row(result))

{

int control = 0;

while (control < 9)

{

cout << setw(13) << row[control];

control++;

}

cout << endl;

}

return mysql\_fetch\_row(result);

}

//从低到高一次存放不同的学历

//该函数接受一个int类型的值，返回一个字符串

string Mysql::put\_field(int num)

{

char field[9][10] = {"工号", "姓名" , "年龄" , "性别" , "电话" , "工资" , "部门" , "学历" , "居住地址" };

return field[num];

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Mysql\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Mysql\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Mysql\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

//动态输出函数

//该函数接受一个字符串

void pause\_out(string str)

{

long count = 10000000;

//通过让一个让为0的数自增到一个很大的值来延迟输出一个字符的时间

//达到一种动态输出的效果

int i = 0;

while (str[i] != '\0')

{

while (count)

count--;

cout << str[i];

i++;

count = 10000000;

}

cout << endl;

}

6.3 main.cpp

#include "class.h"

#include <iostream>

#include <cstring>

using namespace *std*;

int main(void)

{

Menu me;

me.login\_menu();

me.login\_menu\_ctr();

*cout* << *endl*;

}