课 程 设 计 报 告

课程名称 计算机程序设计基础 **2**

班 级 无22

学 号 2022010542

姓 名 王炜致

**2023** 年 **7** 月 **3** 日

一、设计内容与设计要求

1.课程设计目的

面向对象程序设计课程设计是集中实践性环节之一， 是学习完《计算机程序设计基 础 2》C++面向对象程序设计课程后进行的一次全面的综合练习。 要求学生达到熟练掌握 C++语言的基本知识和技能；基本掌握面向对象程序设计的思想和方法；能够利用所学 的基本知识和技能， 解决简单的面向对象程序设计问题，从而提高动手编程解决实际问 题的能力。 尤其重视创新思维培养。

2.课题题目

1) 学生成绩管理系统 (或公司人事管理系统)

3．文档设计要求

3.1 设计课题题目：每个同学都单独完成 1 道课题。后面有范题，仅供同学们参

考，不列入本次课程设计的课题。

3.2 对于程设题目，按照范题的格式。自行虚构软件需求。并按照第 4 点要求， 编写设计文档。 基本要求系统中设计的类的数目不少于 4 个，每个类中要有各自的属 性 (多于 3 个) 和方法 (多于 3 个)；需要定义一个抽象类，采用继承方式派生这些 类。并设计一个多重继承的派生类。在程序设计中，引入虚函数的多态性、运算符重 载等机制。

4. 程序设计的基本要求：

(1)要求利用面向对象的方法以及 C++的编程思想来完成系统的设计；

(2)要求在设计的过程中，建立清晰的类层次；

(3) 根据课题完成以下主要工作： ①完成系统需求分析： 包括系统设计目的与 意义； 系统功能需求(系统流程图)；输入输出的要求。 ②完成系统总体设计： 包括 系统功能分析； 系统功能模块划分与设计(系统功能模块图)。③完成系统详细设计： 数据文件； 类层次图； 界面设计与各功能模块实现。 ④系统调试： 调试出现的主要问 题，编译语法错误及修改， 重点是运行逻辑问题修改和调整。 ⑤使用说明书及编程体 会：说明如何使用你编写的程序， 详细列出每一步的操作步骤。⑥关键源程序(带注 释)。

(4) 自己设计测试数据， 将测试数据存在文件中， 通过文件进行数据读写来获

得测试结果。

(5)按规定格式完成课程设计报告，并在网络学堂上按时提交。

(6) 不得抄袭他人程序、课程设计报告，每个人应独立完成，在程序和设计报 告中体现自己的个性设计。

5、进度安排

|  |  |  |  |
| --- | --- | --- | --- |
| 小学期  第 **2** 周 | 完成类的属性与方法的初步设计 |  |  |
| 编写各个类的属性、功能 |  |  |
| 编写各个类的功能 |  |  |
| 编写各个类的功能并分别测试 |  |  |
| 编写各个类的功能并集中测试 |  |  |

注： 1、一定要保留自己那个课题的完整任务书在课程设计报告里面。

2、“评分表”放在“附录：源程序清单”的后面。

# 目 录

1、系统需求分析 1

2、总体设计 2

3、详细设计 4

4、系统调试 8

5、使用说明书及编程体会 10

6、关键源程序 15

附录：评分表

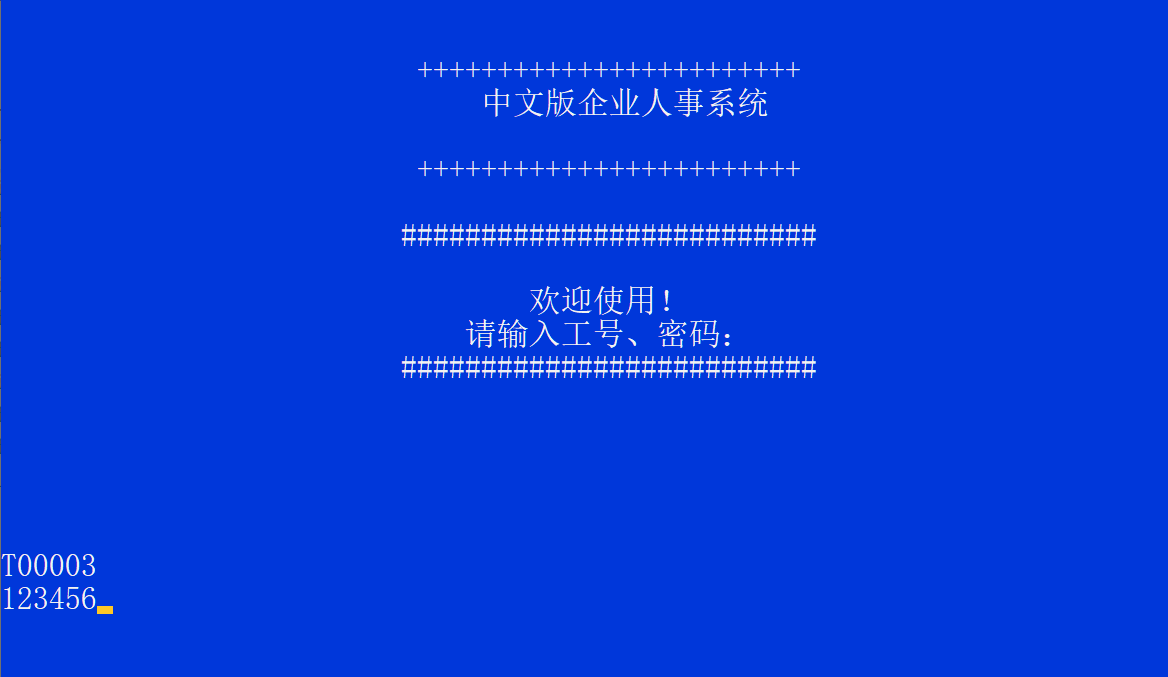
1. 系统需求分析

开发者选择了课题题目“公司人事管理系统”。

在开发者的设想中，本公司人事管理系统是为一家计算机科技开发公司设计的，旨在对公司成员的基本信息进行保存和维护，促进公司内部层次分明的有效管理。作为**本系统特色**，公司成员被简要划分为三类或三个层次：普通程序员（coder类，C）、技术负责人（techmanager类，T）和董事会成员（board类，B），三层次依次增高。技术负责人和董事会成员属于管理层，前者可充当后者的管理助手。

本人事管理系统是面向这三者开发的。他们在人事管理系统中的权限各不相同，层次越高权限越大。在我的设计中，公司人事管理系统需要具备以下基本功能：

1. 人员登录：公司人员凭以C/T/B开头的6位工号及密码登录，根据登录者身份提供不同种类功能。



1. 功能实现：通过菜单方式进行工作，用户通过输入给定的选项进入不同功能页面。系统为不同类用户提供不同的功能。功能可以分为查询、修改、新增、整理统计四个类型。如程序员功能应具备基本的个人信息查询与修改功能；对于管理者而言，还需要具备更多权限如调出、更改他人信息、新增成员等功能；为了衡量下级人员的工作成效，还应当设置绩效属性并配置绩效排行功能**（后注：本程序仅实现了绩效排行功能，而其它排行如年龄、工资排行原理完全一致，为使程序精简，不再添加这些功能）**。操作完成后应当退回菜单页面。
2. 重新登录与退出机制：用户可以退出登录回到登录界面，也可以直接结束程序运行。

由此可见，本管理系统为普通程序员提供了一定的自主管理服务，有助于减轻管理层的负担与保障程序员的基本信息隐私（如密码）；同时管理层仍具有相当大的权限。值得注意的是，使用该系统模板，通过改变称谓可以在大体上适合不同类型公司的使用需求，具有较强的适配性。

2、总体设计

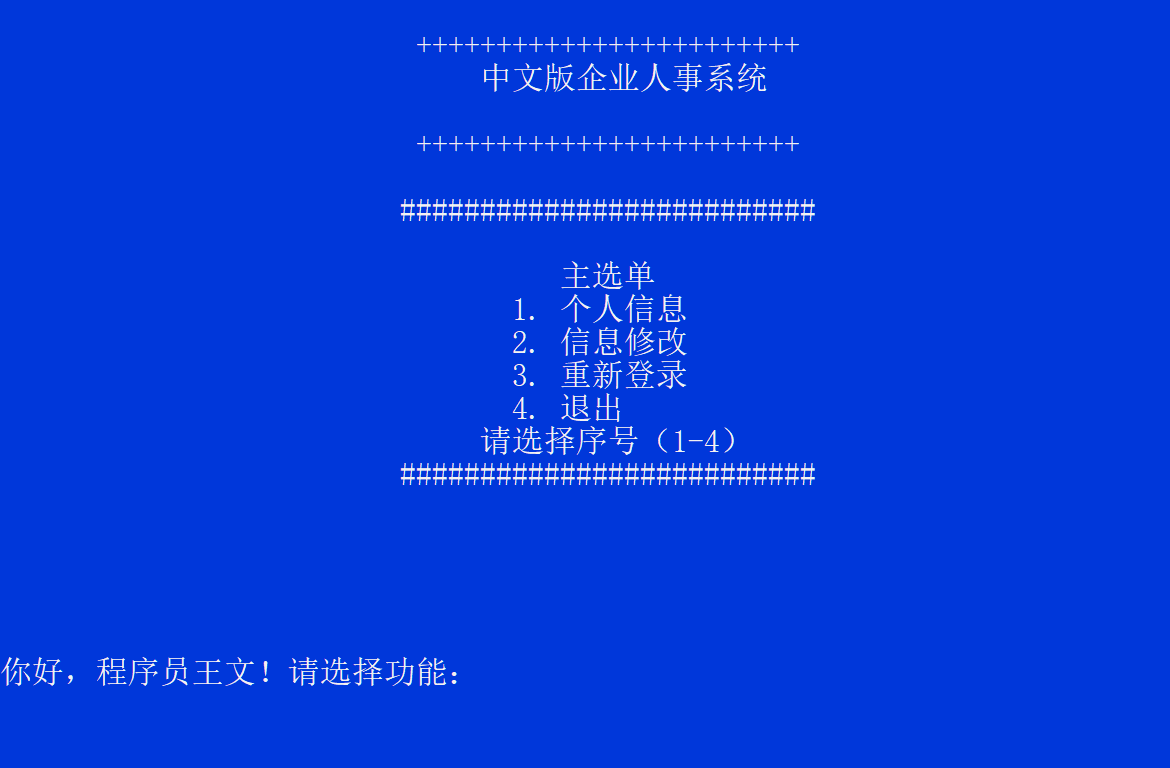
0）通过自定义login函数实现登录功能：login函数读取文件信息。提供返回值，根据返回值判断程序下一步运行；如登录成功，login函数以文件中储存的信息对相应对象进行初始化。

1）为程序员C提供的功能：

①个人信息查询

②个人信息修改（只能修改一部分，工号、工资等无法自行修改）

③重新登录与退出机制



2）为技术负责人T提供的功能：

①个人信息查询

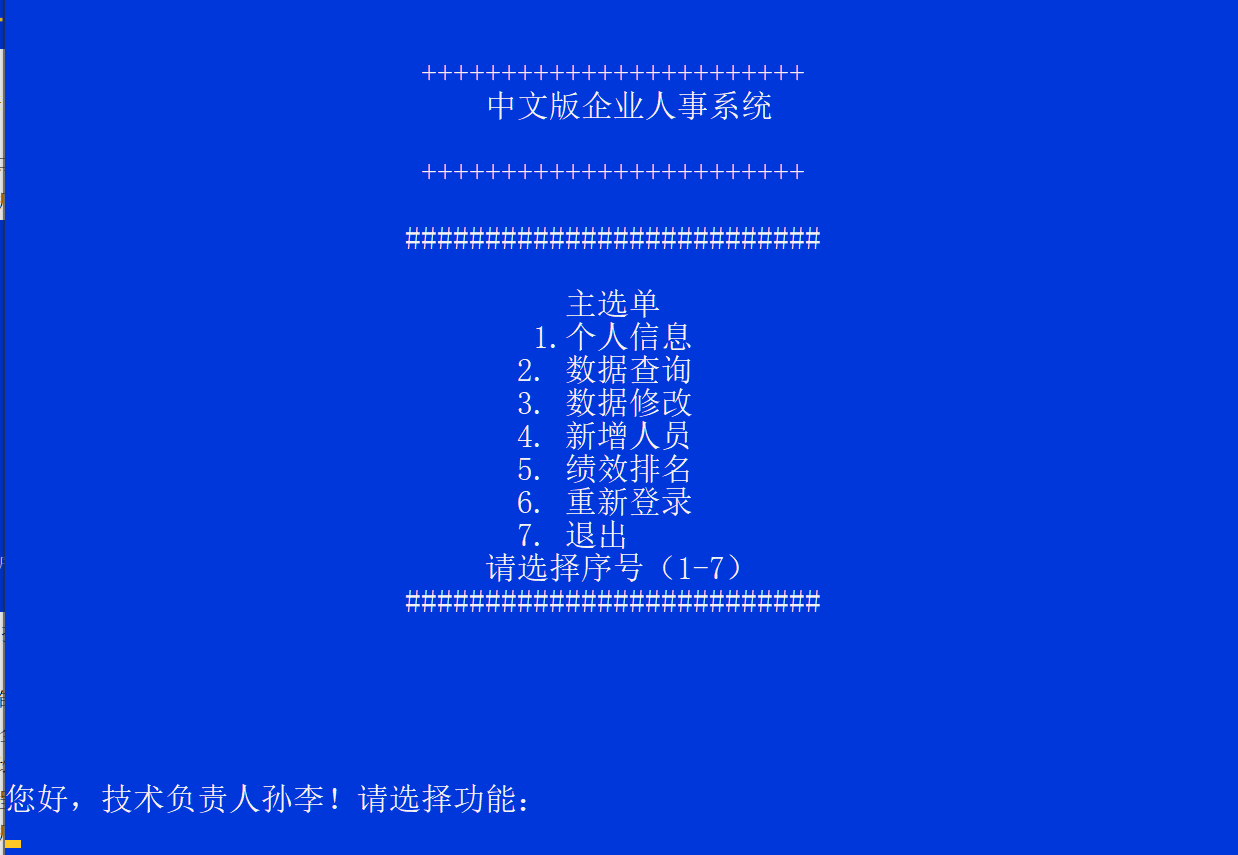
②公司人员信息查询（只能查询一部分，密码等无法查询）

③数据修改（自己/全体程序员，只能修改一部分）

④人员新增（C,T,B均可新增）

⑤绩效排名（仅对程序员进行排名）

⑥重新登录与退出机制



3）为董事会成员B提供的功能：

①个人信息查询

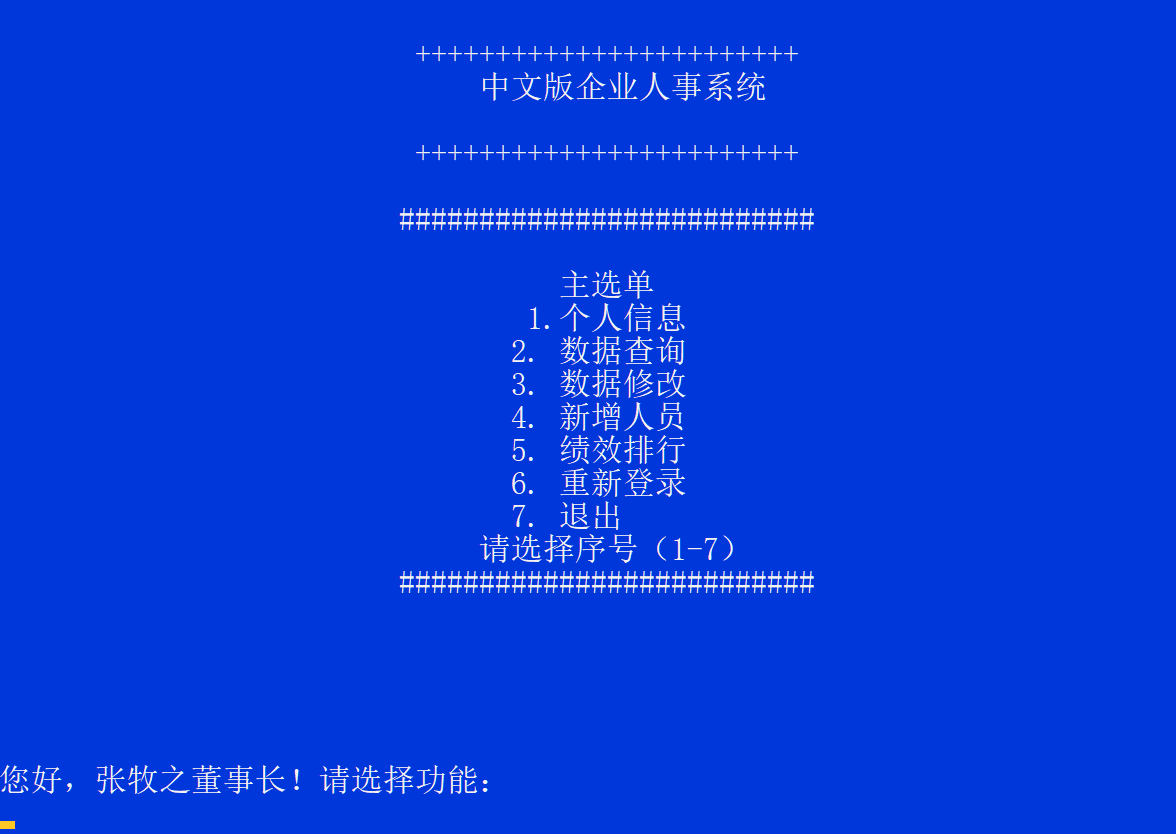
②公司人员信息查询（只能查询一部分，密码等无法查询）

③数据修改（自己/全体程序员/全体技术负责人，只能修改一部分）

④人员新增（C,T,B均可新增）

⑤绩效排名（对程序员、技术负责人分别进行排名）

⑥重新登录与退出机制



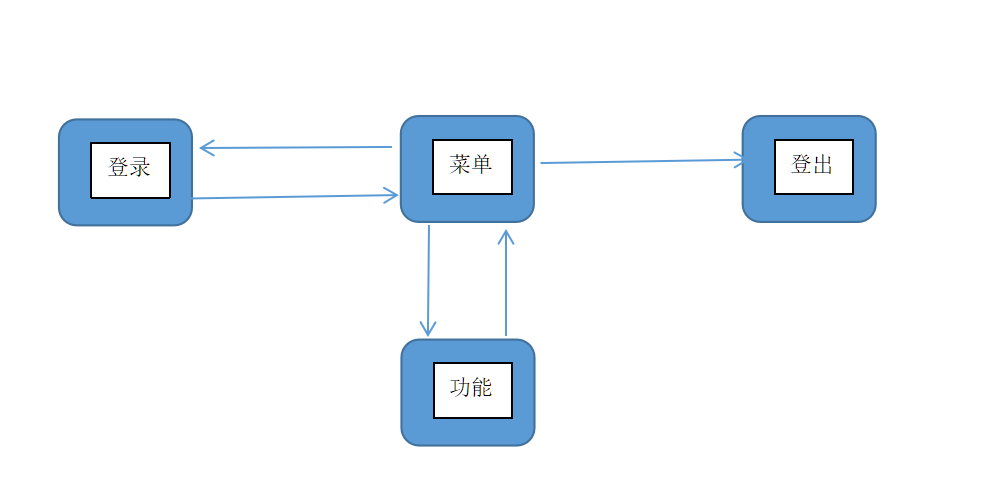
4）输入要求与防错机制：

①登录失败（工号、密码不匹配）反馈-重输机制

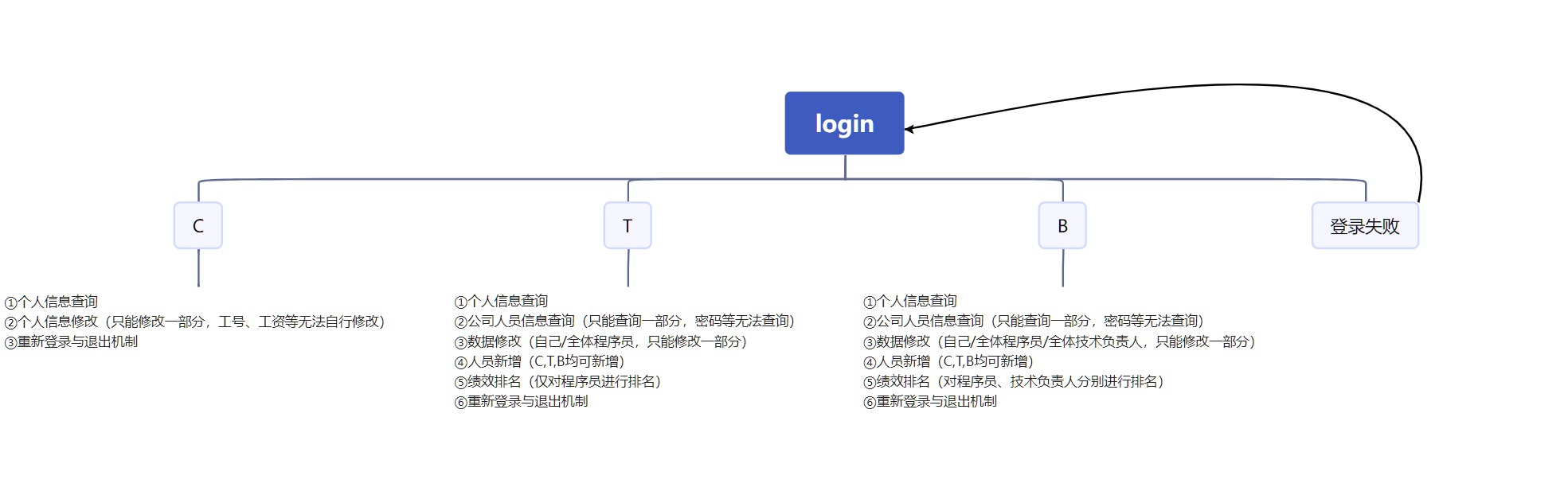
②主选单功能选项输入错误（输入其它数字或字符）反馈-重输机制

③查询失败（查无该工号或无权限访问）反馈-重输机制

系统基本流程图如下：

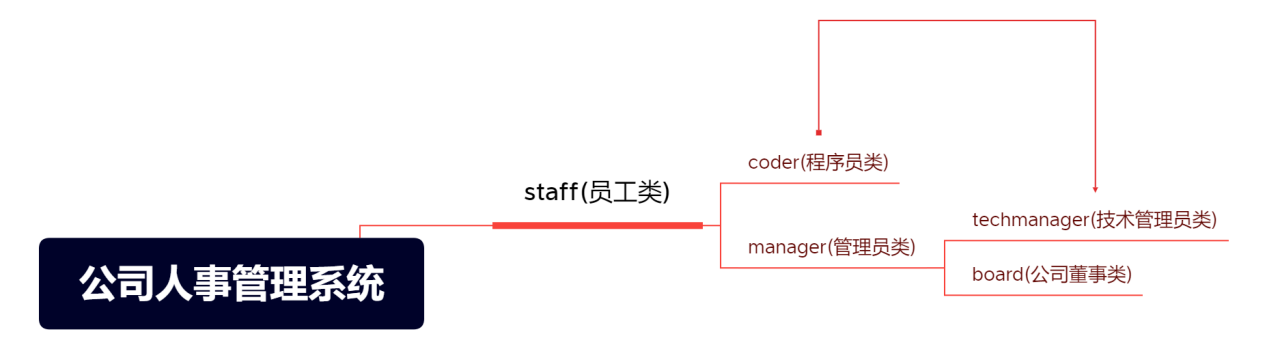


具体功能如下：

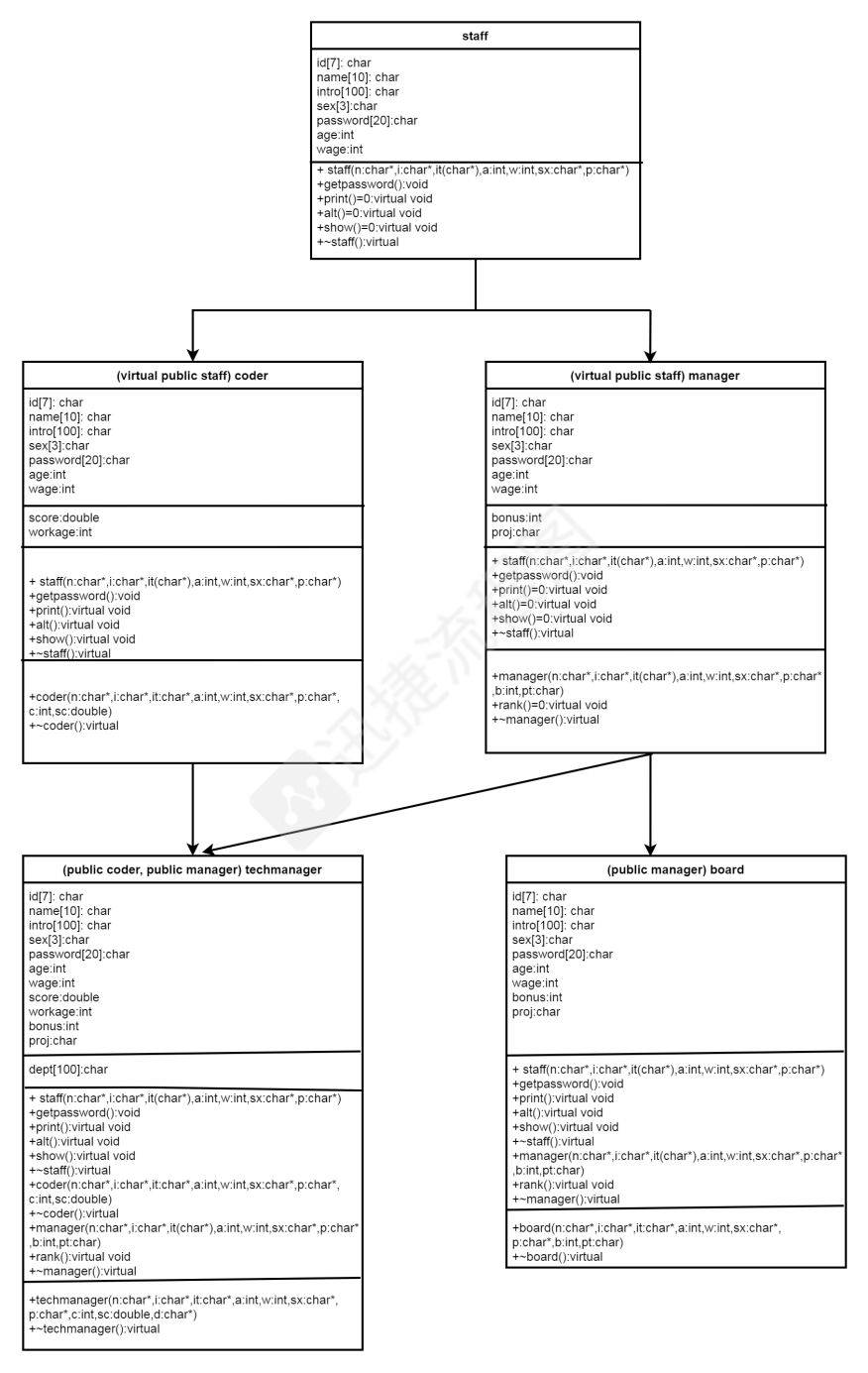


3、详细设计

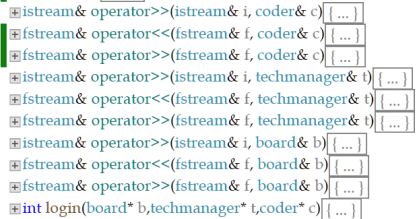
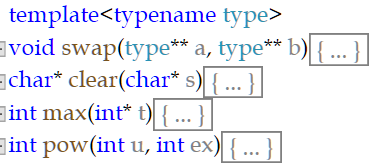
设置**抽象类staff类**，由staff类派生出coder类和manager类。techmanager类由虚基类coder类与虚基类manager类共同派生，为**多重继承派生类**。board类由manager类派生。**共有5个类**。类层次图如下：



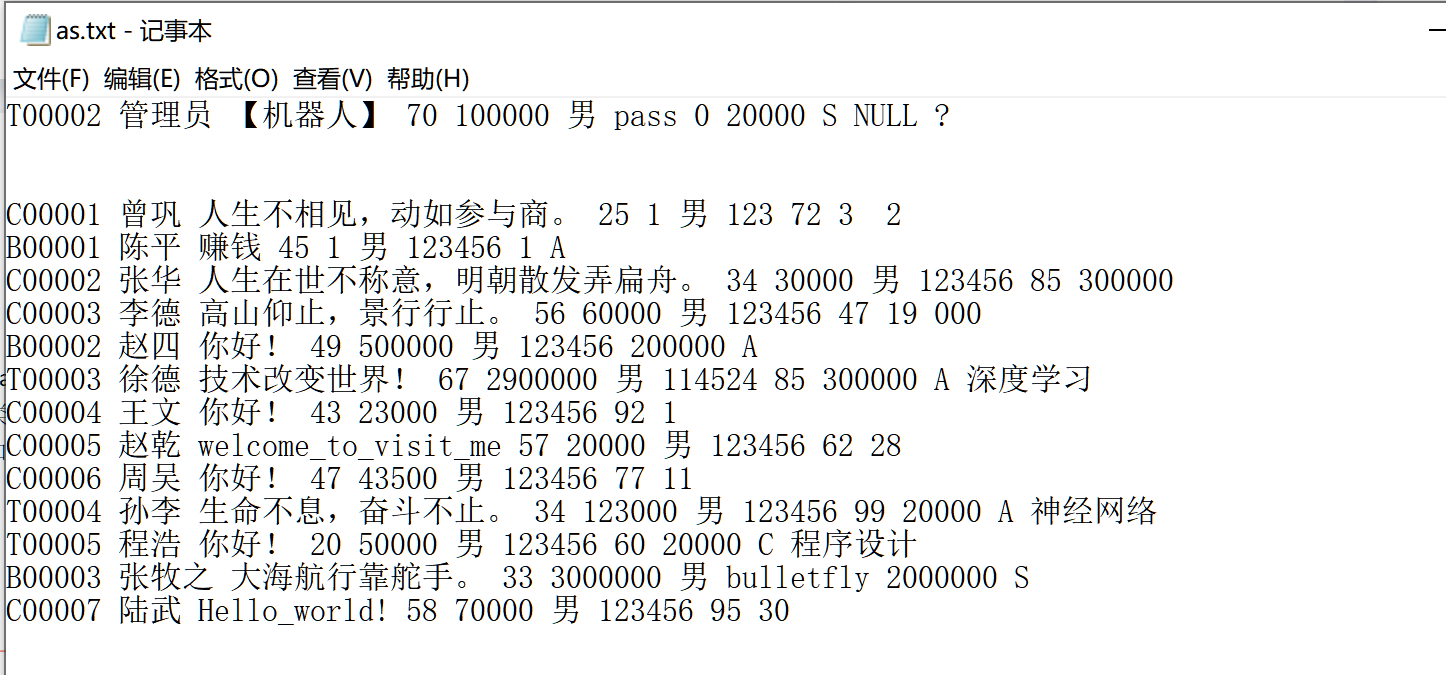
进一步地，五个类的UML图如下，其中体现了**虚函数多态性**（**由继承性，可见每个类的属性和方法均超过3个**）：



此外，还有如下非成员函数（详见源代码及注释）：

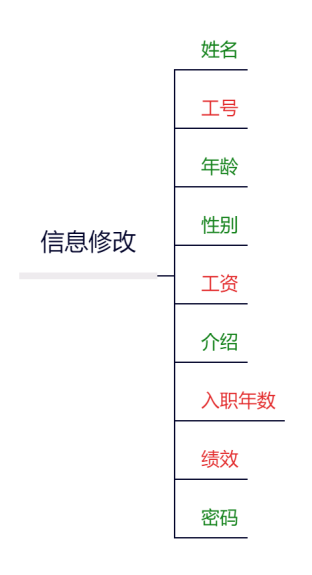
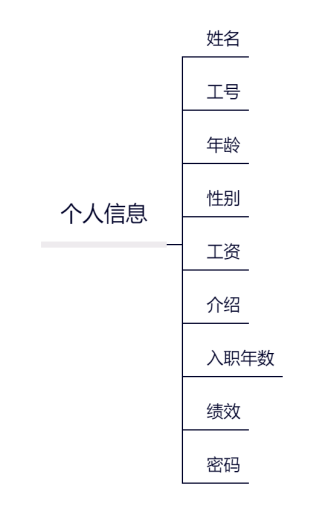


文件比较清晰地记载了成员的数据。为保证文件读、写、删（实际上是覆盖）的可靠性，开发者做了如下设计（具体体现在源代码中）：根据每行首字母判定人员类型；通过空格分隔人员属性（**需要注意的是这一局限性——键入char[]类型数据时不能键入空格，需要用\_代替**）；由于同类型人员属性数目是固定的，读取时总是可以一一对应；每一行覆写时最后补一个空格，可以防止新覆写数据与未被覆盖的旧数据相连而影响读取；每行尾部有大量空格，可以有效防止覆写数据过长而对后面几行信息造成破坏，影响数据读取。文件具体格式如下。

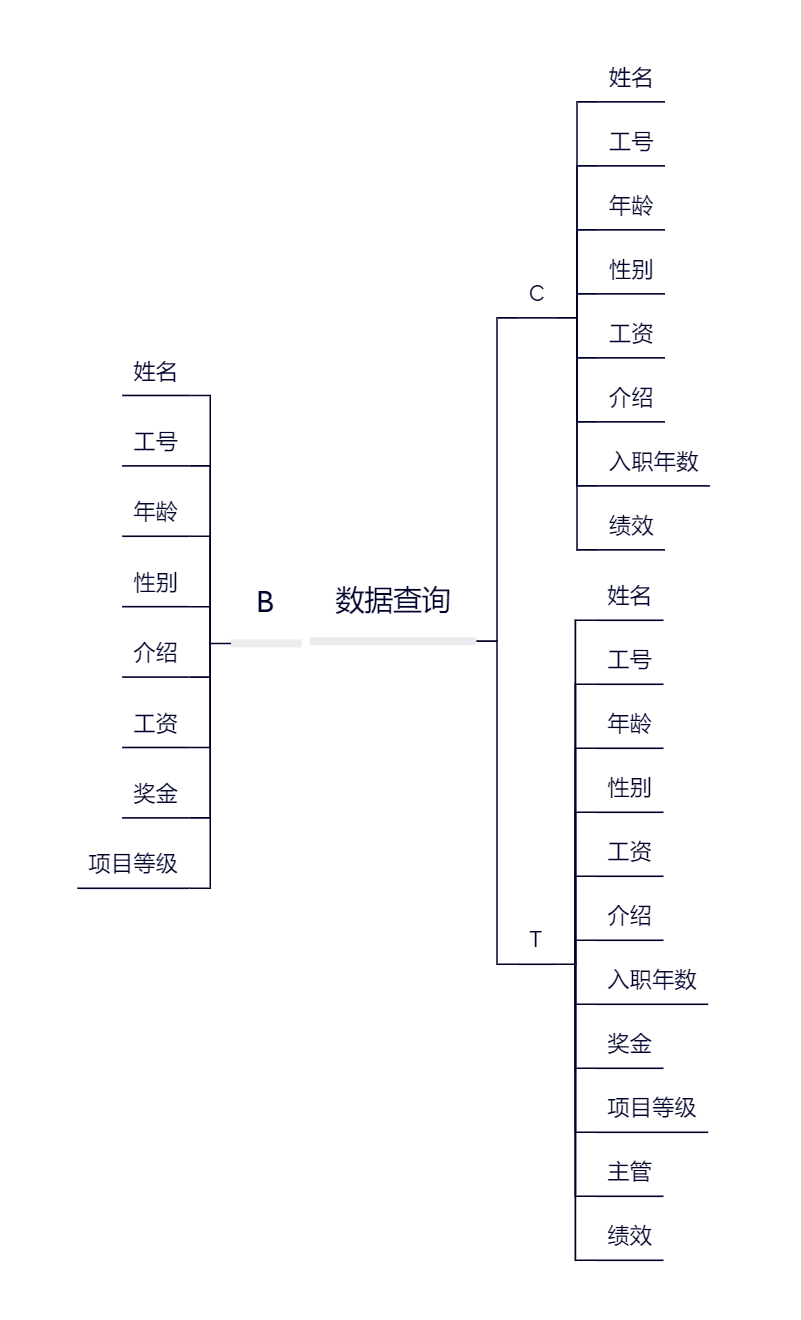
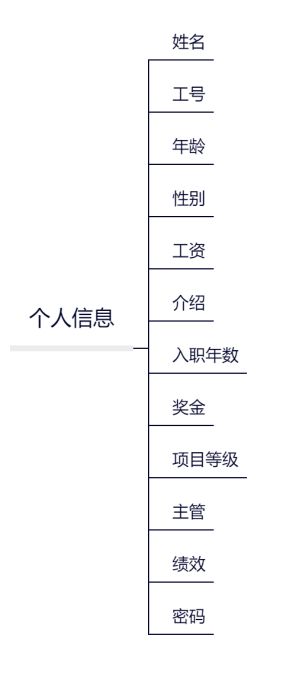


接下来对功能进行详细介绍（重新登录与退出功能是显然的，从略）。

1. 程序员C功能规划（红色代表无法更改，绿色代表可以更改，下同）

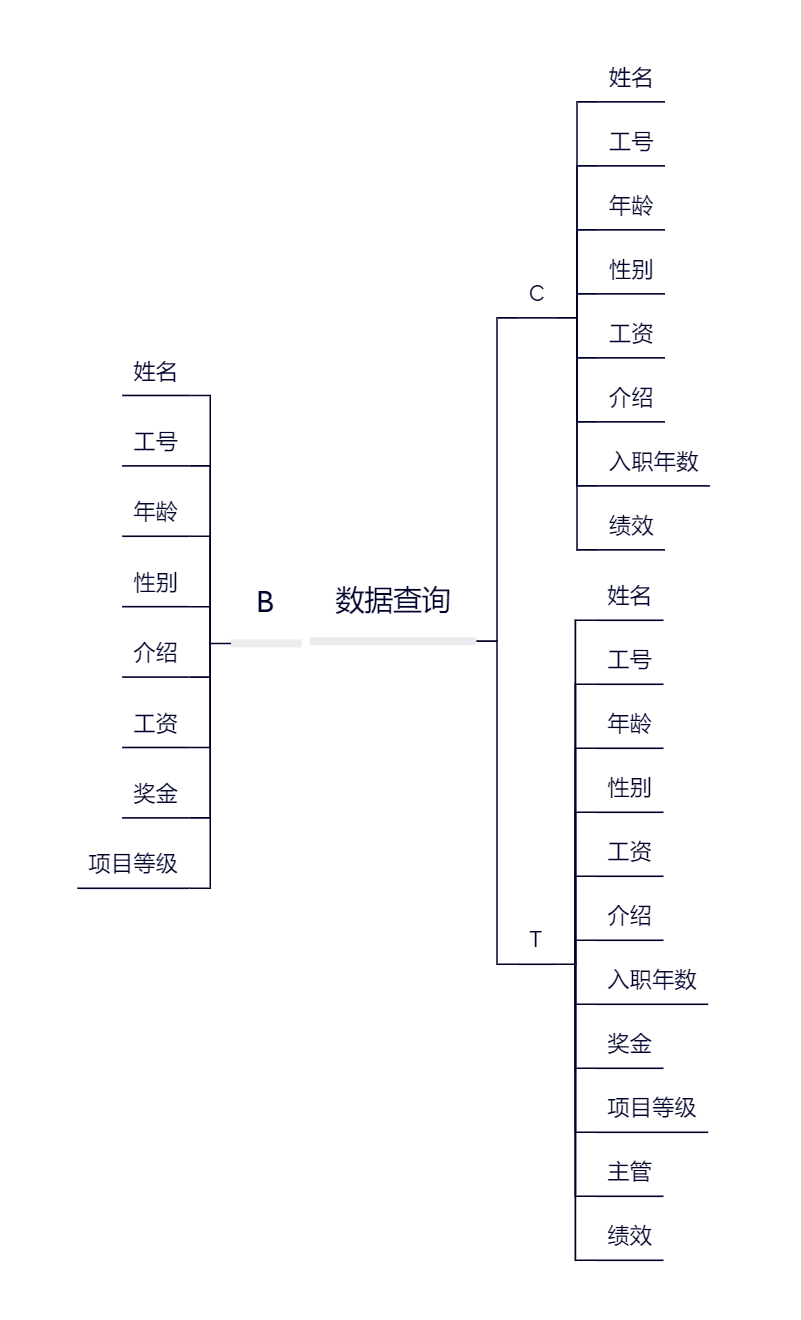


1. 技术负责人T功能规划





1. 董事会成员B功能规划





1. 系统调试

在编写程序的同时，我不断对程序进行调试，发现了以下问题，并针对性地给出了解决方案。解决了这些问题，程序能够比较稳定地实现它的目标功能。

1. 有时cout, cin会提示“不明确”。

方案：注释掉using namespace std;，再去掉注释即可。

1. 在调试修改函数alt()的过程中，发现在文件写入时，新信息对旧文本是逐字节覆盖的。如果新信息过长，则会将\n覆盖从而侵吞下一行的文本，破坏了文本的正常结构；如果新信息过短，则旧文本无法被完全覆盖，在该行留下残留。

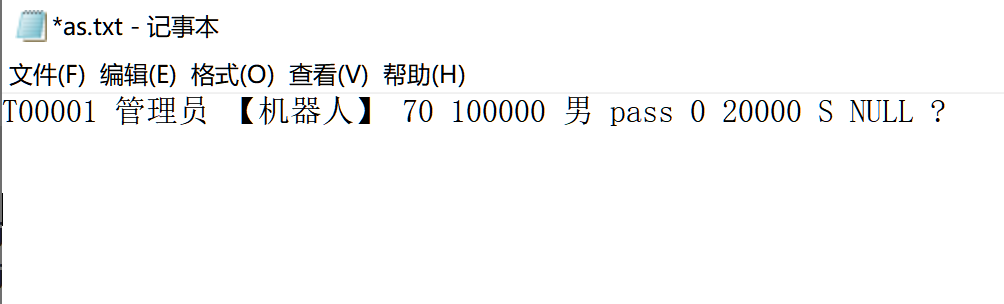
方案：一开始考虑引入行删除功能（没有现成的函数，需要通过创建新文件来实现），但是过于麻烦且使得代码冗长，于是考虑到更优方案：①在创建新人员（新行）时，在人员信息后补入长空格，以防止侵吞下一行，利用getline切到下一行；②在写入人员信息后再补一个空格，分开新旧文本（读取的时候不会读到旧文本）。

1. 有时会出现无法读入的情况，例如：文本中有C00001，但程序似乎把它忽略了。创建新程序员时，系统仍然自动分配C00001。这可能是文件结构有问题。

方案：在第一行后面多空几行即可，尽管这降低了文本文档的美观度。

以下是新建一位程序员的调试案例：

1↓



2↓



3↓



4↓



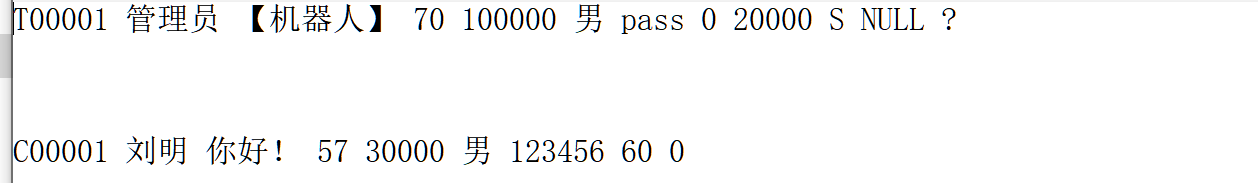
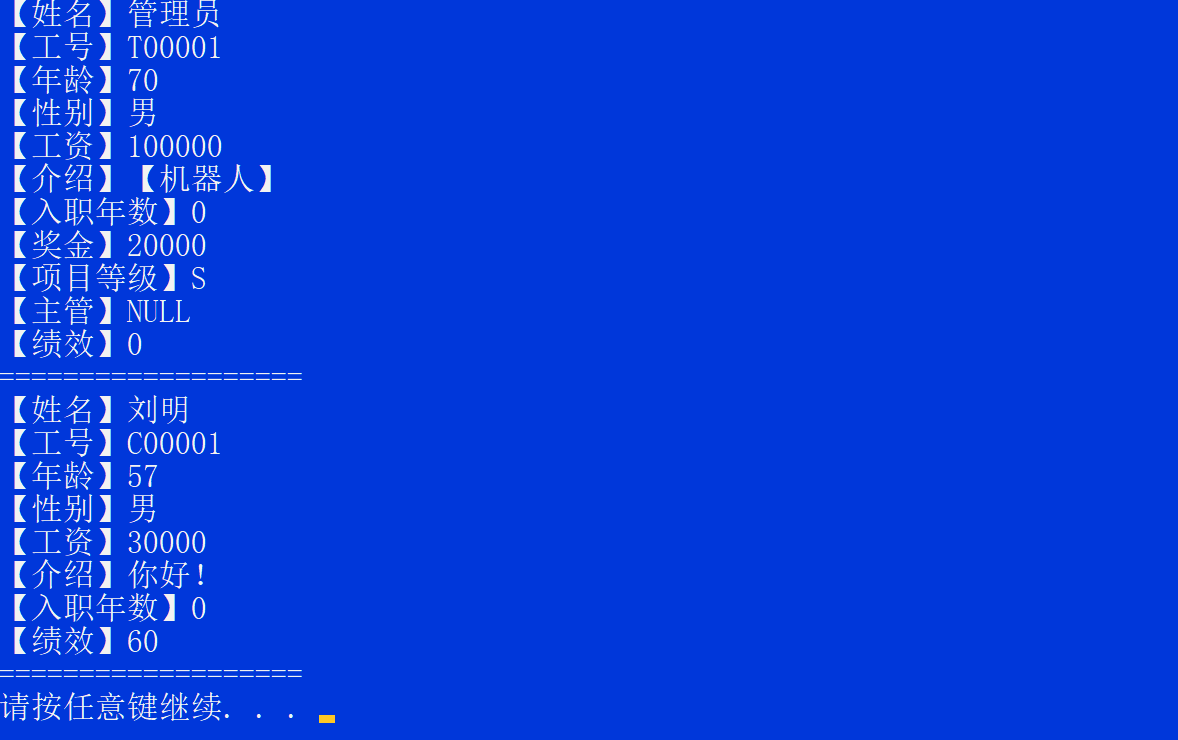
5↓

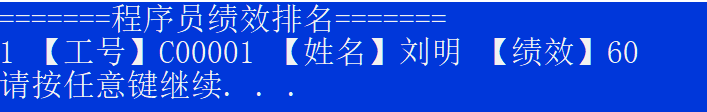


6↓



7↓



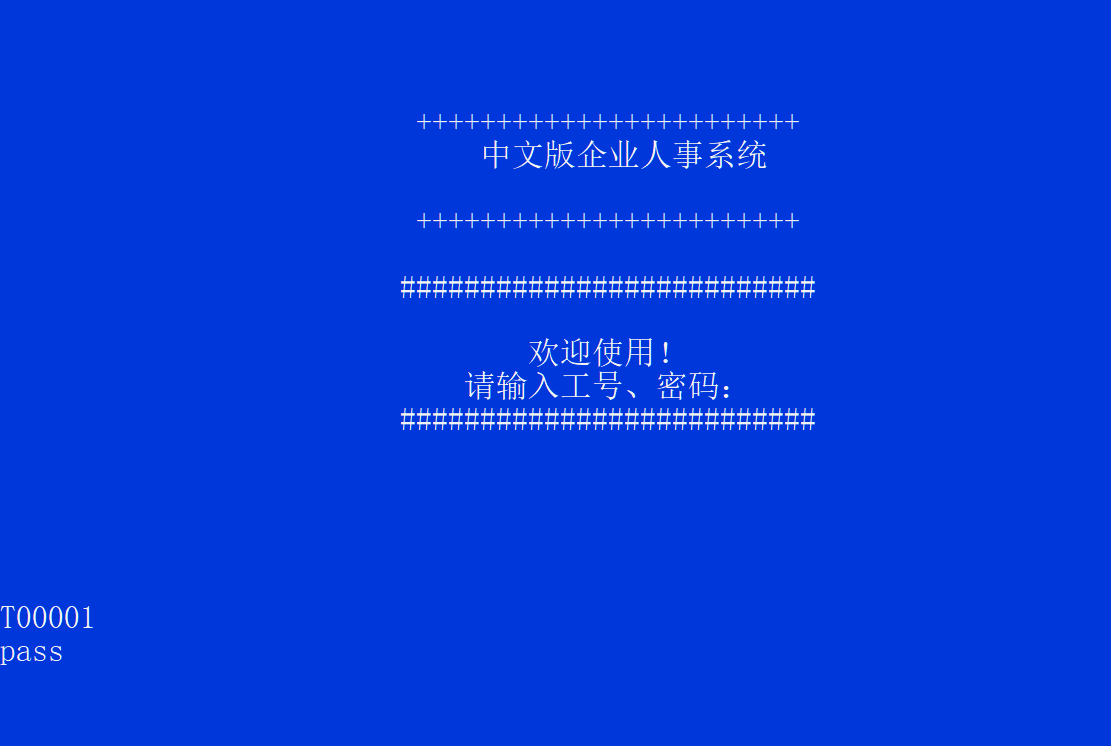


1. 使用说明书及编程体会
2. **登录说明书**

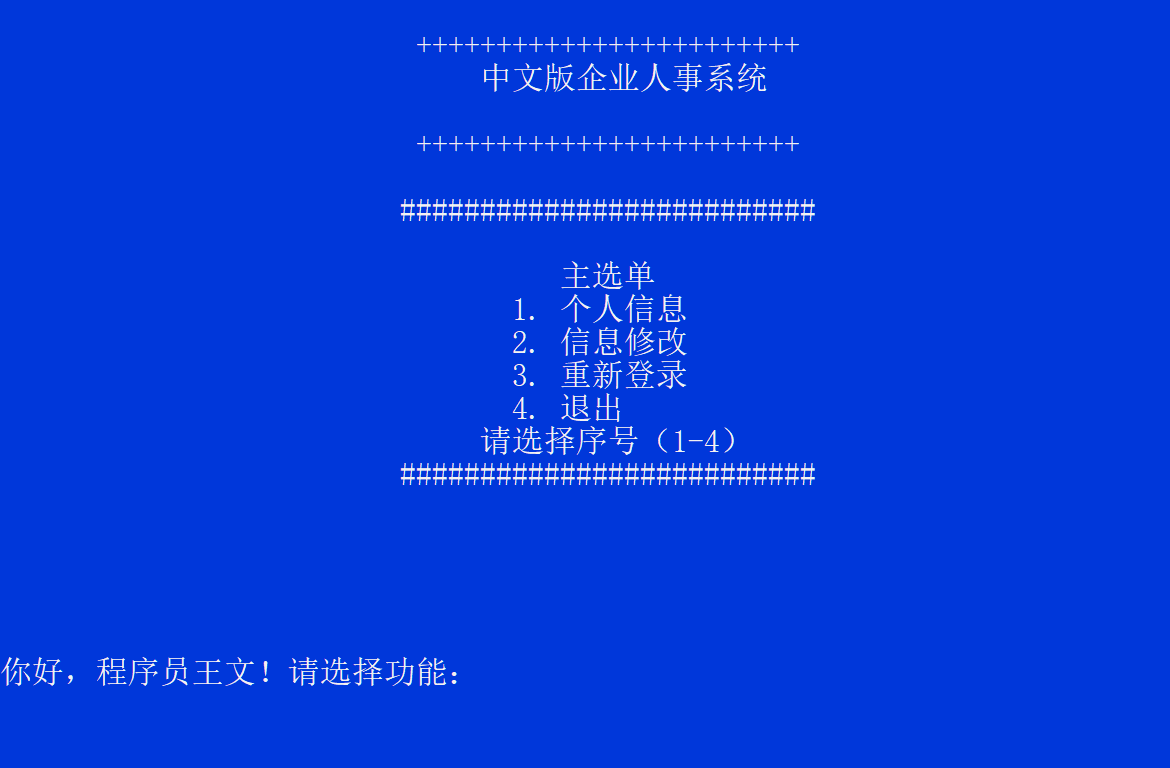
最初使用时，信息文件as.txt中仅有



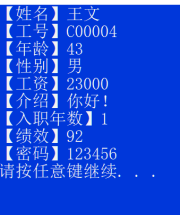
需要先用工号T00001, 密码pass登录，然后利用【新增人员】功能录入人员信息。



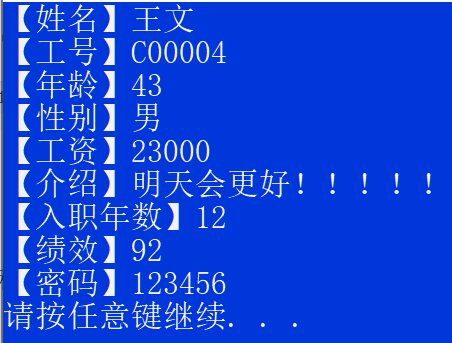
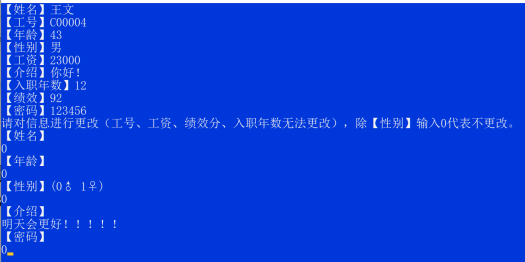
**1）程序员C功能说明书**



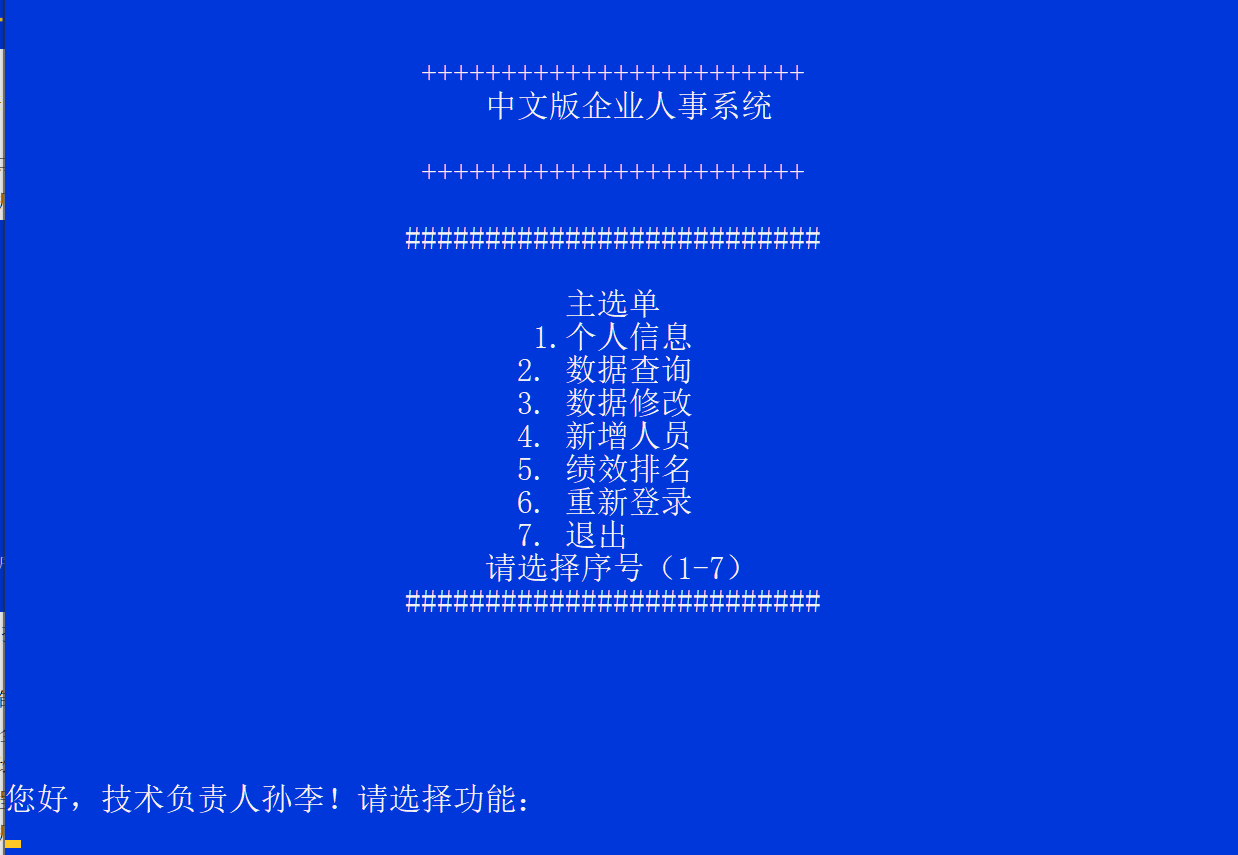
①输入1选择个人信息：展示该程序员全部信息↓



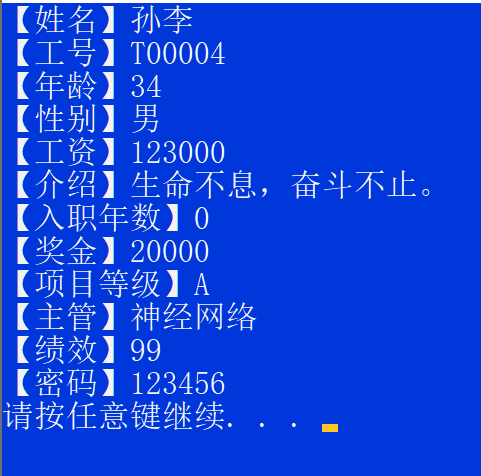
②输入2选择信息修改：修改该程序员部分信息↓



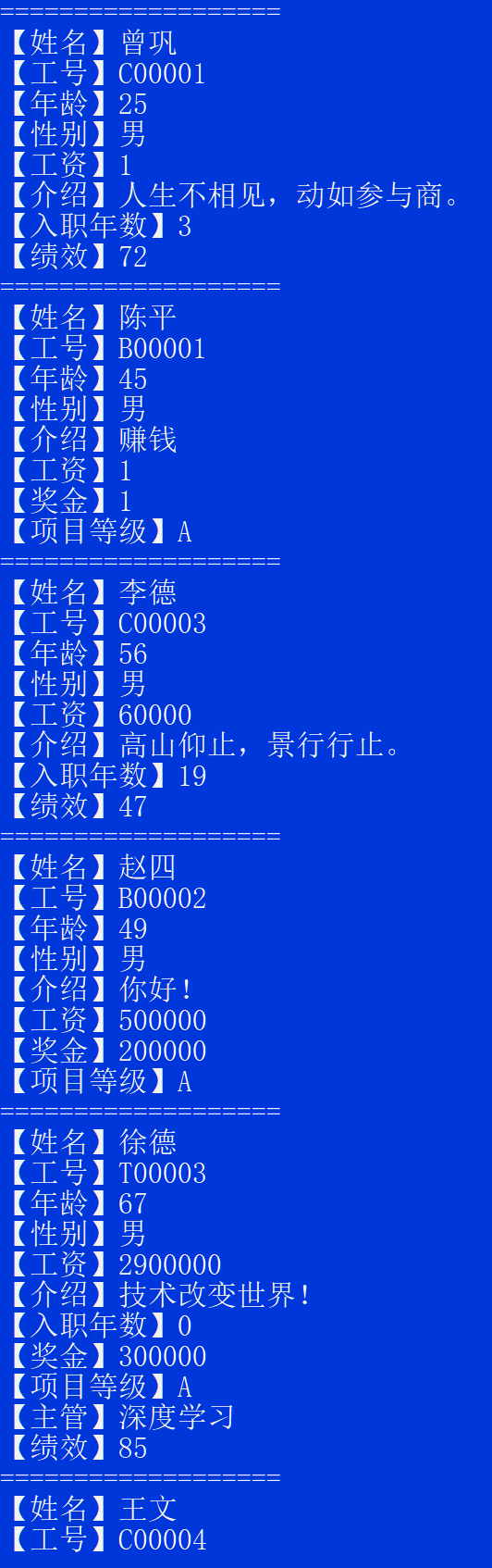
#### 技术负责人T功能说明书



①输入1选择个人信息：展示该技术负责人全部信息↓

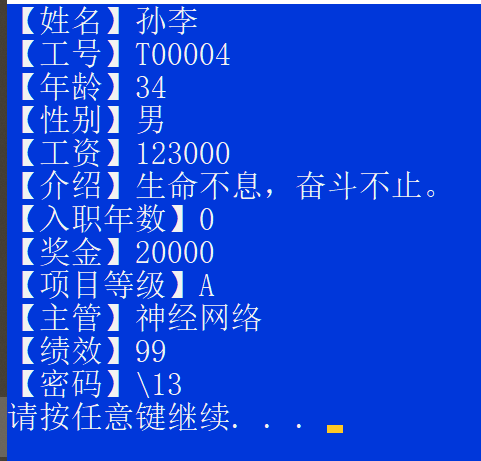
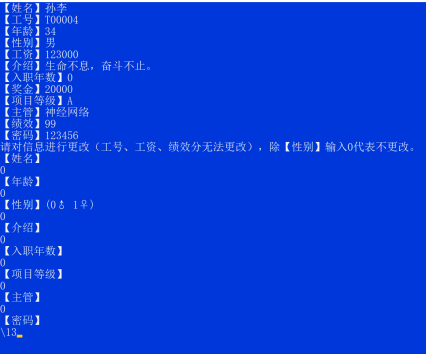


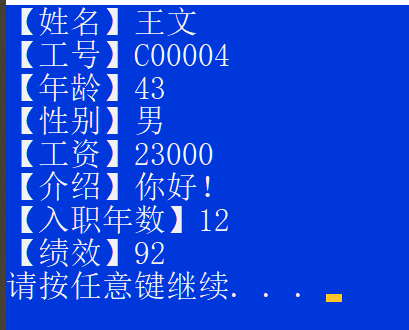
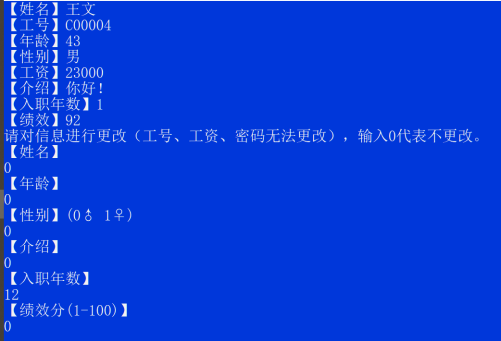
②输入2选择数据查询：展示所有人除密码外的全部信息↓



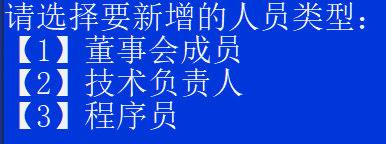
③输入3选择数据修改：可以修改自己的部分信息↓也可以修改任意程序员的部分信息↓↓

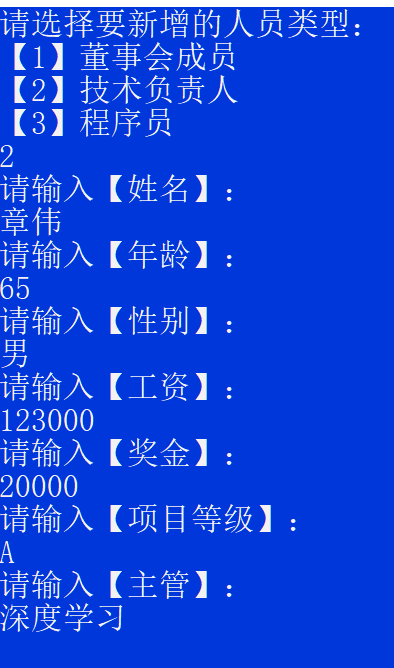


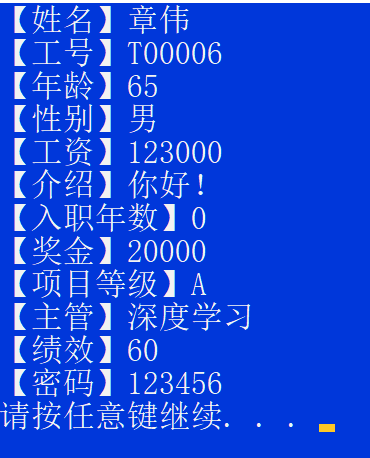




④输入4选择新增人员：可以新增C/T/B类人员。程序提供自动编号机制，新编号为同类最大编号+1↓



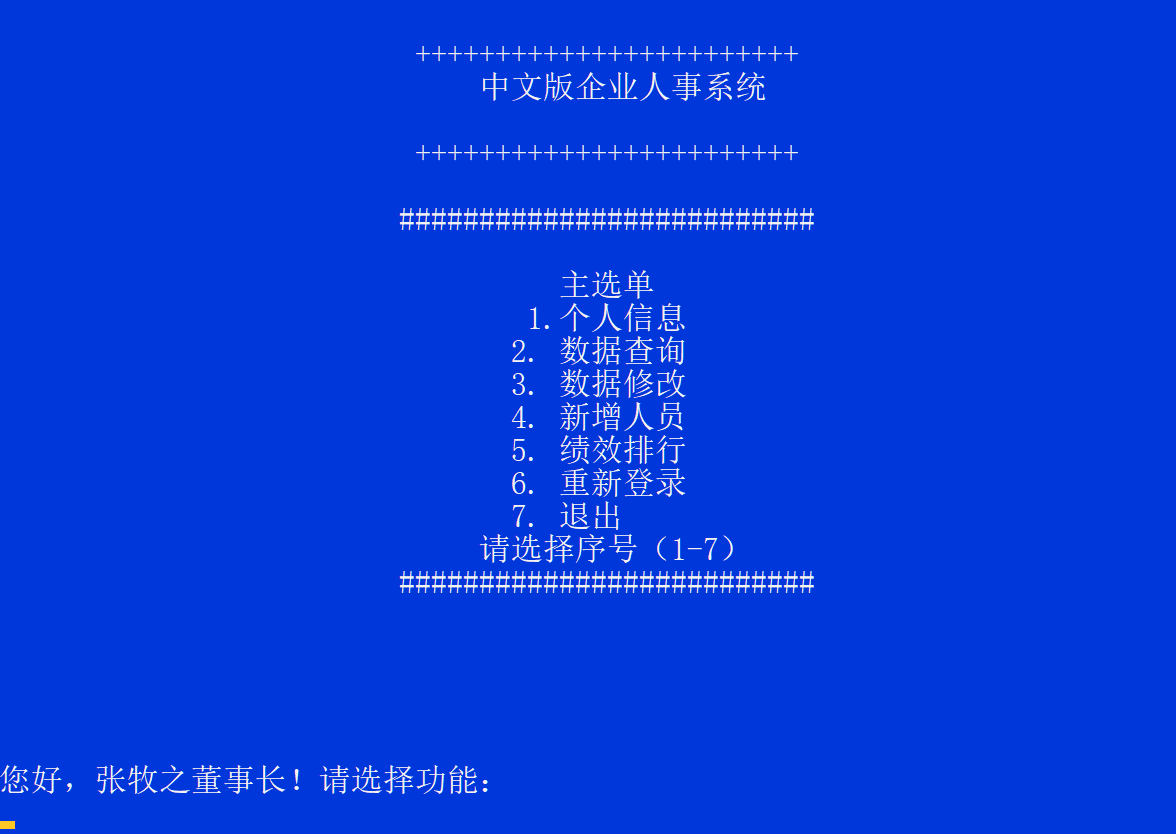




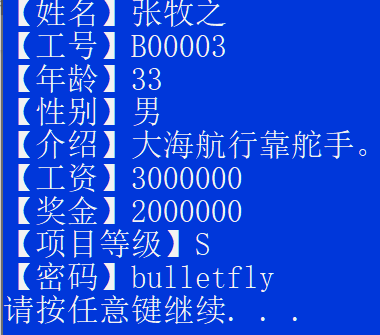
⑤输入5选择绩效排名：对全体程序员进行绩效排名↓



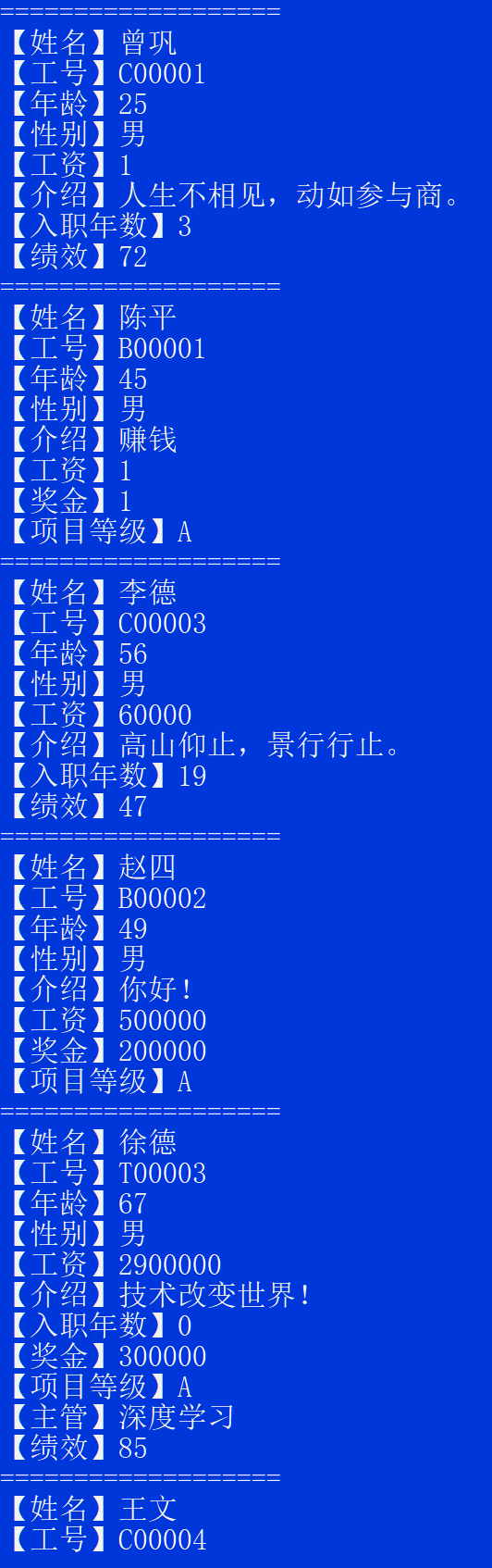
#### 董事会成员B功能说明书



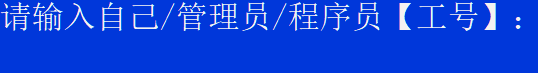
①输入1选择个人信息：展示该董事会成员全部信息↓



②输入2选择数据查询：展示所有人除密码外的全部信息↓

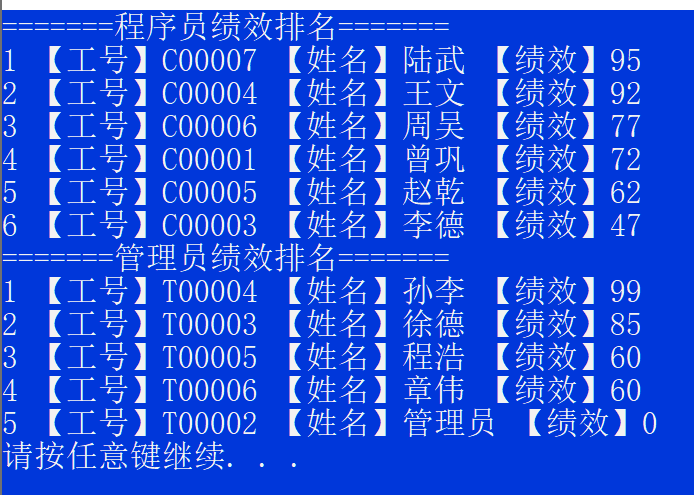


③输入3选择修改信息：可以修改自己的部分信息↓也可以修改任意程序员、负责人部分信息，过程与技术负责人类似，从略。



④输入4选择新增人员：可以新增C/T/B类人员。程序提供自动编号机制，新编号为同类最大编号+1，过程与技术负责人类似，从略。

⑤输入5选择绩效排名：对全体程序员、技术负责人进行绩效排名↓



【注意：年龄输入限制[20,70], 工资奖金限制C[0,1e6], T[0,1e7], B[0,1e8]】

**编程体会**

在本次编程中，我初次感受到了写一个较大项目的诸多困难。我意识到，编写代码前的宏观设计是非常有必要的，它有助于我们保持清醒的头脑，在面对越来越长的代码时不会因为忘记代码块对应功能而失去头绪。若不做好对属性、方法、目标功能等的预先规划，则后期的任何更改将牵一发而动全身，耗费我大量的时间精力进行调试修改。另一方面，代码的精简度也是我一向的追求，然而在本次程序设计中我做得还不够好。有部分代码实际上是高度重复的，但我并没有对这些代码做好精简工作。也许，通过更精妙的宏观设计，代码行数还可以大大减省。然而仍然值得欣喜的是，我通过有效的处理方案顺利、较快地解决了目前所发现的程序硬伤，让程序能够比较正常地工作；我在这样一次大作业综合训练中，收获了许多程序设计的经验，获得了长足的进步。

再补充谈谈本程序的实用性问题。我选择了“公司人事管理系统”这个课题，但我对现实公司人事的具体情况毕竟不甚了解。经过一些了解，我虚构了公司背景并针对性地设计了这样一个管理平台。面对现实问题，它必然存在许多设计上、考虑上的缺陷，但这应当是与程序设计能力无关的；相信只要对于公司人事情况有了切身的了解后，我能够设计出更贴合实际情况的人事管理系统。

6、源程序

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

char bin[1000];//供getline跳转到下一行使用

template<typename type>

void swap(type\*\* a, type\*\* b) {//传地址以进行值交换，用于绩效排名

type\* temp; temp = \*a; \*a = \*b; \*b = temp;}

char\* clear(char\* s) {//将字符数组每项置为\0，以便strcmp

for (int i = 0; i < strlen(s); i++) s[i] = '\0';

return s;}

int max(int\* t) { int max = 0;//求数组最大值，用于创建新编号

for (int i = 0; i <= 100; i++) {

if (t[i] > max) max = t[i];} return max + 1;}

int pow(int u, int ex) { int temp = 1;//乘方函数，用于编号大小比较

for (int i = 0; i < ex; i++) temp \*= u;

return temp;}

class coder; class manager;

class techmanager; class board;//提前声明以便login函数调用

class staff {//抽象类staff

protected:

char id[7]; char name[10]; char intro[100];//工号、姓名、介绍

int age; int wage; char sex[3]; char password[20];//年龄、工资、性别、密码

public:

staff(char\* n=(char\*)".", char\* i = (char\*)".", char\* it = (char\*)"你好！",

int a = 45, int w=5000,char\*sx=(char\*)"男", char\* p = (char\*)"123456") :

age(a), wage(w) {

clear(name); clear(id); clear(intro); clear(sex);

clear(password); strcpy\_s(name, n);

strcpy\_s(id, i); strcpy\_s(intro, it);

strcpy\_s(password, p); strcpy\_s(sex, sx); }

void getpassword() { cout << "【密码】" << password << endl; }//独立于print()，照顾到密码的隐私性

virtual void print() = 0; virtual void alt() = 0; virtual void show() = 0; //定义纯虚函数以便派生类内重载

friend int login(board\*, techmanager\*, coder\*);//将login函数设为友元，便于其访问对象的id和password

virtual ~staff() {}};

class coder :virtual public staff {//程序员类

protected: double score; int workage;//绩效、入职年数

public:

coder(char\* n = (char\*)".", char\* i = (char\*)".", char\* it = (char\*)"你好！",

int a = 45, int w = 5000, char\* sx = (char\*)"男", char\* p = (char\*)"123456", int c=0, double sc=60)

:staff(n,i,it,a,w,sx,p),workage(c),score(sc) {}

virtual void print() {

cout << "【姓名】" << name << endl << "【工号】" << id << endl

<< "【年龄】" << age << endl << "【性别】" << sex << endl

<< "【工资】" << wage << endl << "【介绍】" << intro << endl

<< "【入职年数】" << workage << endl <<"【绩效】"<<score<<endl; }

virtual void alt() {

char n[100]; int i; system("cls"); print(); getpassword();

cout << "请对信息进行更改（工号、工资、绩效分、入职年数无法更改），除【性别】输入0代表不更改。" << endl;

cout << "【姓名】"<<endl; cin>>n; if((n[0]-48)||n[1]) { clear(name); strcpy\_s(name,n); }//更改姓名，判断输入是否为0，下同

cout << "【年龄】"<<endl; cin >> i; while (cin.fail() || i<20&&i|| i>70)//更改年龄，若输入字符或不符合区间则重新输入，下同

{ cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; }

if (i) { age = i; }//判断输入是否为0，下同

cout << "【性别】(0♂ 1♀)" << endl; cin >> i; while (cin.fail() || i !=0 && i!=1) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; } clear(sex);

switch (i) {

case 0:strcpy\_s(sex, "男"); break;

case 1: strcpy\_s(sex, "女"); }

cout << "【介绍】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(intro); strcpy\_s(intro, n); }

cout << "【密码】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(password); strcpy\_s(password, n); }

system("cls"); print(); getpassword();

fstream iof("as.txt", ios::in | ios::out); char ids[10] = { '\0' };//将更改后的信息覆写入文件

try { int err = 1; if (!iof) throw err; }//检查文件是否打开成功，下同

catch (int) { cerr << "文件出错，请检查。" << endl; exit(0); }

while(!iof.eof()) { iof >> ids;//匹配工号与文件中保存的工号

if (!strcmp(ids, id)) {

iof.seekg(-6, ios::cur); iof << \*this<<" "; iof.close(); break;}//覆写，关闭文件

else iof.getline(n, 2023); }system("pause");}

virtual void show() {

system("cls");

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

cout.fill(' '); cout.setf(ios::right);

cout << setw(50) << "++++++++++++++++++++++++" << endl

<< setw(48) << "中文版企业人事系统" << endl << endl

<< setw(50) << "++++++++++++++++++++++++" << endl << endl

<< setw(51) << "##########################" << endl << endl

<< setw(41) << "主选单" << endl

<< setw(43) << "1. 个人信息" << endl

<< setw(43) << "2. 信息修改" << endl

<< setw(43) << "3. 重新登录" << endl

<< setw(39) << "4. 退出" << endl

<< setw(47) << "请选择序号（1-4）" << endl

<< setw(51) << "##########################" << endl;

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

cout << "你好，程序员"<<name<<"！请选择功能：" << endl; }

friend void anew();

friend board; friend techmanager;

friend istream& operator>>(istream&, coder&);//运算符重载，下同

friend fstream& operator<<(fstream&, coder&);

friend fstream& operator>>(fstream&, coder&);

virtual ~coder() {}};

class manager :virtual public staff {//管理员类

protected: int bonus; char proj;//奖金、项目等级

public:

manager(char\* n = (char\*)".", char\* i = (char\*)".", char\* it = (char\*)"你好！",

int a = 45, int w = 5000, char\* sx = (char\*)"男", char\* p = (char\*)"123456", int b=1000,char pt='A') :

staff(n, i, it, a, w, sx, p), bonus(b), proj(pt){}

friend void anew(); virtual void rank() = 0; virtual ~manager() {};};

class techmanager : public coder, public manager {//技术负责人类

protected: char dept[100];//主管

public:

techmanager(char\* n = (char\*)".", char\* i = (char\*)".", char\* it = (char\*)"你好！",

int a = 45, int w = 5000, char\* sx = (char\*)"男", char\* p = (char\*)"123456", int c = 0, int b = 1000, char pt = 'A', char\* d = (char\*)".",double sc=60)

:staff(n, i, it, a, w, sx, p), coder(n, i, it, a, w, sx, p, c, sc), manager(n, i, it, a, w, sx, p, b, pt) { clear(dept); strcpy\_s(dept, d); }

virtual void print() {

cout << "【姓名】" << name << endl << "【工号】" << id << endl

<< "【年龄】" << age << endl << "【性别】" << sex << endl

<< "【工资】" << wage << endl << "【介绍】" << intro << endl << "【入职年数】" << workage << endl

<< "【奖金】" << bonus << endl << "【项目等级】" << proj << endl << "【主管】" << dept << endl<<"【绩效】"<<score<<endl ; }

virtual void alt() {

LT: coder C; char ID[100]; char IDc[100];

cout << "请输入自己或程序员【工号】：" << endl; cin >> ID;

fstream iof("as.txt", ios::in | ios::out); system("cls"); char n[1000]; int i;

try { int err = 1; if (!iof) throw err; }

catch (int) { cerr << "文件出错，请检查。" << endl; exit(0); }

if (!strcmp(ID, id)) {//判断是否是自己的工号，对自己信息进行修改

print(); getpassword();

cout << "请对信息进行更改（工号、工资、绩效分无法更改），除【性别】输入0代表不更改。" << endl;

cout << "【姓名】" << endl; cin >> n; if ((n[0] - 48) || n[1]) { clear(name); strcpy\_s(name, n); }

cout << "【年龄】" << endl; cin >> i; while (cin.fail() || i < 20&&i || i>70){

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; }

if (i) { age = i; }

cout << "【性别】(0♂ 1♀)" << endl; cin >> i; while (cin.fail() || i != 0 && i != 1) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;

} clear(sex);

switch (i) {

case 0:strcpy\_s(sex, "男"); break; case 1: strcpy\_s(sex, "女");}

cout << "【介绍】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(intro); strcpy\_s(intro, n); }

cout << "【入职年数】" << endl; cin >> i; while (cin.fail() || i < 0) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;}

if (i) { workage = i; }

cout << "【项目等级】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { proj = n[0]; cin.clear(); }

cout << "【主管】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(dept); strcpy\_s(dept, n); }

cout << "【密码】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(password); strcpy\_s(password, n); }

system("cls"); print(); getpassword();

while (!iof.eof()) {

iof >> ID;

if (!strcmp(ID, id)) {

iof.seekg(-6, ios::cur); iof << \*this <<" "; iof.close(); break; }

else iof.getline(bin, 1000);}}

else {bool flag = 0;//若不是自己的工号，则考虑是不是程序员工号

while (!iof.eof()) {

iof >> IDc;

if (!strcmp(IDc, ID)&&ID[0]=='C') {

iof.seekg(-6, ios::cur);

iof >> C; iof.close(); flag = 1; break;}

else iof.getline(bin, 1000); }

if (flag) {//若有程序员工号成功匹配，则修改程序员信息

system("cls"); C.print();

cout << "请对信息进行更改（工号、工资、密码无法更改），输入0代表不更改。" << endl;

cout << "【姓名】" << endl; cin >> n; if ((n[0] - 48) || n[1]) { clear(C.name); strcpy\_s(C.name, n); }

cout << "【年龄】" << endl; cin >> i; while (cin.fail() || i < 0 || i>70)

{cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; }

if (i) { C.age = i; }

cout << "【性别】(0♂ 1♀)" << endl; cin >> i; while (cin.fail() || i != 0 && i != 1) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; } clear(C.sex);

switch (i) {case 0:strcpy\_s(C.sex, "男"); break; case 1: strcpy\_s(C.sex, "女");}

cout << "【介绍】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(C.intro); strcpy\_s(C.intro, n); }

cout << "【入职年数】" << endl; cin >> i; while (cin.fail() || i < 0) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { C.workage = i; }

cout << "【绩效分(1-100)】" << endl; cin >> i; while (cin.fail() || i < 0 || i>100) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { C.score = i; }

system("cls"); C.print(); iof.open("as.txt", ios::in | ios::out);

while (!iof.eof()) {

iof >> IDc;

if (!strcmp(IDc, ID)) {

iof.seekg(-6, ios::cur);

iof << C<<" "; iof.close(); break; }

else iof.getline(bin, 1000, '\n');}}

else {cout << "查无此人或无权更改！" << endl; goto LT; }}

iof.close(); }

virtual void show() {

system("cls");

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

cout.fill(' '); cout.setf(ios::right);

cout << setw(50) << "++++++++++++++++++++++++" << endl

<< setw(48) << "中文版企业人事系统" << endl

<< endl

<< setw(50) << "++++++++++++++++++++++++" << endl

<< endl

<< setw(51) << "##########################" << endl

<< endl

<< setw(41) << "主选单" << endl

<< setw(43) << "1.个人信息" << endl

<< setw(43) << "2. 数据查询" << endl

<< setw(43) << "3. 数据修改" << endl

<< setw(43) << "4. 新增人员" << endl

<< setw(43) << "5. 绩效排名" << endl

<< setw(43) << "6. 重新登录" << endl

<< setw(39) << "7. 退出" << endl

<< setw(47) << "请选择序号（1-7）" << endl

<< setw(51) << "##########################" << endl;

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

cout << "您好，技术负责人" << name << "！请选择功能：" << endl; }

virtual void rank() {

fstream iof("as.txt", ios::in | ios::out); int n = 0;

try { int err = 1; if (!iof) throw err; }

catch (int) { cerr << "文件出错，请检查。" << endl; exit(0); }

while (!iof.eof()) {

if (iof.peek() == 'C') n++; //这一次打开文件是为了统计程序员个数

iof.getline(bin, 1000);}

if (!n) { cout << "暂无程序员！" << endl; return; }

coder\* C = new coder[n]; coder\*\* pC = new coder \* [n];

for (int i = 0; i < n; i++) pC[i] = C + i;

n = 0; iof.close(); iof.open("as.txt", ios::in | ios::out);//这一次打开文件是为了把程序员数据读到程序员数组里

while (!iof.eof()) {

if (iof.peek() == 'C') iof >> C[n++];

iof.getline(bin, 1000);} iof.close();

for (int i = 0; i < n-1 ; i++) {

for (int j = 0; j < n - 1 - i; j++) {

if (pC[j]->score < pC[j + 1]->score) {

swap(pC+j, pC+j+1);}}}//冒泡排序

cout << "=======程序员绩效排名=======" << endl;

for (int i = 0; i < n; i++) cout << i+1 <<" 【工号】"<<pC[i]->id <<" 【姓名】" <<pC[i]->name <<" 【绩效】"<<pC[i]->score<< endl;

delete[] C, pC;}

friend board; virtual ~techmanager() {}

friend istream & operator>>(istream&, techmanager&);

friend fstream& operator<<(fstream&, techmanager&);

friend fstream& operator>>(fstream&, techmanager&);

};

class board : public manager {//B

public:

board(char\* n = (char\*)".", char\* i = (char\*)".", char\* it = (char\*)"你好！",

int a = 45, int w = 5000, char\* sx = (char\*)"男", char\* p = (char\*)"123456", int b = 1000, char pt = 'A') :

staff(n, i, it, a, w, sx, p), manager(n, i, it, a, w, sx, p, b, pt) {}

virtual void print() {

cout << "【姓名】" << name << endl << "【工号】" << id << endl

<< "【年龄】" << age << endl << "【性别】" << sex << endl

<< "【介绍】" << intro << endl << "【工资】" << wage << endl << "【奖金】" << bonus << endl << "【项目等级】" << proj << endl;}

virtual void alt() {

LB:coder C; techmanager T; char ID[10]; char IDc[10];

cout << "请输入自己/管理员/程序员【工号】：" << endl; cin >> ID;

fstream iof("as.txt", ios::in | ios::out);

try { int err = 1; if (!iof) throw err; }

catch (int) { cerr << "文件出错，请检查。" << endl; exit(0); }

system("cls"); char n[100]; int i;

if (!strcmp(ID, id)) {//是自己

print(); getpassword();

cout << "请对信息进行更改（工号无法更改），除【性别】输入0代表不更改。" << endl;

cout << "【姓名】" << endl; cin >> n; if ((n[0] - 48) || n[1]) { clear(name); strcpy\_s(name, n); }

cout << "【年龄】" << endl; cin >> i; while (cin.fail() || i < 20&&i || i>70){

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; } if (i) { age = i; }

cout << "【性别】(0♂ 1♀)" << endl; cin >> i; while (cin.fail() || i != 0 && i != 1) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; } clear(sex);

switch (i) {

case 0:strcpy\_s(sex, "男"); break; case 1: strcpy\_s(sex, "女");}

cout << "【介绍】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(intro); strcpy\_s(intro, n); }

cout << "【工资】" << endl; cin >> i; while (cin.fail() || i < 0 ||i>1e8) {//只有董事会成员有修改工资、奖金的权限，下同

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { wage = i; }

cout << "【奖金】" << endl; cin >> i; while (cin.fail() || i < 0||i>1e8) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { bonus = i; }

cout << "【项目等级】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { proj = n[0]; cin.clear(); }

cout << "【密码】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(password); strcpy\_s(password, n); }

system("cls"); print(); getpassword();

while (!iof.eof()) {//定位、覆写

iof >> ID;

if (!strcmp(ID, id)) {

iof.seekg(-6, ios::cur); iof << \*this <<" "; iof.close(); break; }

else iof.getline(bin, 1000);}}

else {bool flag = 0;

if (ID[0] == 'C') {//可能是程序员类

while (!iof.eof()) {

iof >> IDc;

if (!strcmp(IDc, ID)) {

iof.seekg(-6, ios::cur);

iof >> C; iof.close(); flag = 1; break;}

else iof.getline(bin, 1000); }

if (flag) {

system("cls"); C.print();

cout << "请对信息进行更改（工号、密码无法更改），除【性别】输入0代表不更改。" << endl;

cout << "【姓名】" << endl; cin >> n; if ((n[0] - 48) || n[1]) { clear(C.name); strcpy\_s(C.name, n); }

cout << "【年龄】" << endl; cin >> i; while (cin.fail() || i < 0 || i>70)

{cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; }

if (i) { C.age = i; }

cout << "【性别】(0♂ 1♀)" << endl; cin >> i; while (cin.fail() || i != 0 && i != 1) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; } clear(C.sex);

switch (i) {case 0:strcpy\_s(C.sex, "男"); break; case 1: strcpy\_s(C.sex, "女");}

cout << "【介绍】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(C.intro); strcpy\_s(C.intro, n); }

cout << "【入职年数】" << endl; cin >> i; while (cin.fail() || i < 0) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { C.workage = i; }

cout << "【工资】" << endl; cin >> i; while (cin.fail() || i < 0 ||i>1e6) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { C.wage = i; }

cout << "【绩效分(1-100)】" << endl; cin >> i; while (cin.fail() || i < 0 || i>100) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { C.score = i; }

system("cls"); C.print(); iof.open("as.txt", ios::in | ios::out);

while (!iof.eof()) {//定位、覆写

iof >> IDc;

if (!strcmp(IDc, ID)) {

iof.seekg(-6, ios::cur);

iof << C<<" "; iof.close(); break; }

else iof.getline(bin, 1000, '\n');}}

}

else if (ID[0] == 'T') {//可能是负责人类

while (!iof.eof()) {

iof >> IDc;

if (!strcmp(IDc, ID)) {

iof.seekg(-6, ios::cur);

iof >> T; iof.close(); flag = 1; break;}

else iof.getline(bin, 1000); }

if (flag) {

system("cls"); T.print();

cout << "请对信息进行更改（工号、工资、密码无法更改），输入0代表不更改。" << endl;

cout << "【姓名】" << endl; cin >> n; if ((n[0] - 48) || n[1]) { clear(T.name); strcpy\_s(T.name, n); }

cout << "【年龄】" << endl; cin >> i; while (cin.fail() || i < 0 || i>70)

{cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; }

if (i) { T.age = i; }

cout << "【性别】(0♂ 1♀)" << endl; cin >> i; while (cin.fail() || i != 0 && i != 1) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i; } clear(T.sex);

switch (i) {case 0:strcpy\_s(T.sex, "男"); break; case 1: strcpy\_s(T.sex, "女");}

cout << "【介绍】" << endl; clear(n); cin >> n; if ((n[0] - 48) || n[1]) { clear(T.intro); strcpy\_s(T.intro, n); }

cout << "【入职年数】" << endl; cin >> i; while (cin.fail() || i < 0) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { T.workage = i; }

cout << "【工资】" << endl; cin >> i; while (cin.fail() || i < 0 ||i>1e7) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { T.wage = i; }

cout << "【奖金】" << endl; cin >> i; while (cin.fail() || i < 0 || i>1e7) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { T.bonus = i; }

cout << "【绩效分(1-100)】" << endl; cin >> i; while (cin.fail() || i < 0 || i>100) {

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; cin >> i;} if (i) { T.score = i; }

system("cls"); T.print(); iof.open("as.txt", ios::in | ios::out);

while (!iof.eof()) {//定位、覆写

iof >> IDc;

if (!strcmp(IDc, ID)) {

iof.seekg(-6, ios::cur);

iof << T<<" "; iof.close(); break; }

else iof.getline(bin, 1000, '\n');}}}

else {cout << "查无此人或无权更改！" << endl;goto LB;}}

iof.close();}

virtual void show() {

system("cls");

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

cout.fill(' '); cout.setf(ios::right);

cout << setw(50) << "++++++++++++++++++++++++" << endl

<< setw(48) << "中文版企业人事系统" << endl << endl

<< setw(50) << "++++++++++++++++++++++++" << endl << endl

<< setw(51) << "##########################" << endl << endl

<< setw(41) << "主选单" << endl

<< setw(43) << "1.个人信息" << endl

<< setw(43) << "2. 数据查询" << endl

<< setw(43) << "3. 数据修改" << endl

<< setw(43) << "4. 新增人员" << endl

<< setw(43) << "5. 绩效排行" << endl

<< setw(43) << "6. 重新登录" << endl

<< setw(39) << "7. 退出" << endl

<< setw(47) << "请选择序号（1-7）" << endl

<< setw(51) << "##########################" << endl;

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

cout << "您好，" << name << "董事长！请选择功能：" << endl; }

virtual void rank() {

fstream iof("as.txt", ios::in | ios::out); int n = 0;

try { int err = 1; if (!iof) throw err; }

catch (int) { cerr << "文件出错，请检查。" << endl; exit(0); }

while (!iof.eof()) {

if (iof.peek() == 'C') n++;

iof.getline(bin, 1000);}

if (!n) { cout << "暂无程序员！" << endl; iof.close(); return; }

coder\* C = new coder[n]; coder\*\* pC = new coder \* [n];

for (int i = 0; i < n; i++) pC[i] = C + i;

n = 0; iof.close(); iof.open("as.txt", ios::in | ios::out);

while (!iof.eof()) {

if (iof.peek() == 'C') iof >> C[n++];

iof.getline(bin, 1000);} iof.close();

for (int i = 0; i < n-1 ; i++) {

for (int j = 0; j < n - 1 - i; j++) {

if (pC[j]->score < pC[j + 1]->score) {

swap(pC + j, pC + j + 1);}}}

cout << "=======程序员绩效排名=======" << endl;

for (int i = 0; i < n; i++) cout << i + 1 << " 【工号】" << pC[i]->id << " 【姓名】" << pC[i]->name << " 【绩效】" << pC[i]->score << endl;

iof.close(); delete[] C, pC; iof.open("as.txt", ios::in | ios::out); n = 0;

while (!iof.eof()) {

if (iof.peek() == 'T') n++;

iof.getline(bin, 1000);}

if (!n) { cout << "暂无管理员！" << endl; iof.close(); return; }

techmanager\* T = new techmanager[n]; techmanager\*\* pT = new techmanager \* [n];

for (int i = 0; i < n; i++) pT[i] = T + i;

n = 0; iof.close(); iof.open("as.txt", ios::in | ios::out);

while (!iof.eof()) {

if (iof.peek() == 'T') iof >> T[n++];

iof.getline(bin, 1000);} iof.close();

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - 1 - i; j++) {

if (pT[j]->score < pT[j + 1]->score) {

swap(pT + j, pT + j + 1);}}}

cout << "=======管理员绩效排名=======" << endl;

for (int i = 0; i < n; i++) cout << i + 1 << " 【工号】" << pT[i]->id << " 【姓名】" << pT[i]->name << " 【绩效】" << pT[i]->score << endl;

delete[] T, pT;}

friend istream& operator>>(istream&, board&);

friend fstream& operator<<(fstream&, board&);

friend fstream& operator>>(fstream&, board&);

virtual ~board() {}};

istream& operator>>(istream& i, coder& c) {//工号、简介、密码、绩效、入职年数均默认

cout << "请输入【姓名】：" << endl; i >> c.name;

cout << "请输入【年龄】：" << endl; i >> c.age; while (i.fail() || c.age < 20 || c.age>70){

cin.clear(); cin.ignore(); cout << "请重新输入!" << endl; i >> c.age;}

cout << "请输入【性别】：" << endl; i >> c.sex;

cout << "请输入【工资】：" << endl; i >> c.wage; while (i.fail() || c.wage < 0 || c.wage>1e6) {//设置工资（奖金）上限，下同

i.clear(); i.ignore(); cout << "请重新输入!" << endl; i >> c.age; }

return i; }

fstream& operator<<(fstream& f, coder& c) {

f << c.id << " " << c.name << " " << c.intro << " " << c.age << " " << c.wage << " " << c.sex << " " << c.password << " " <<c.score<<" " << c.workage;

return f; }//设置空格以区分不同属性，下同

fstream& operator>>(fstream& f, coder& c) {

f >> c.id; f.ignore(); f >> c.name; f.ignore();

f >> c.intro; f.ignore(); f >> c.age; f.ignore();

f >> c.wage; f.ignore(); f >> c.sex; f.ignore();

f >> c.password; f.ignore(); f >> c.score; f.ignore(); //忽略空格，下同

f >> c.workage; return f; }

istream& operator>>(istream& i, techmanager& t) {

cout << "请输入【姓名】：" << endl; i >> t.name;

cout << "请输入【年龄】：" << endl; i >> t.age; while (i.fail() || t.age < 20 || t.age>70) {

i.clear(); i.ignore(); cout << "请重新输入!" << endl; i >> t.age; }

cout << "请输入【性别】：" << endl; i >> t.sex;

cout << "请输入【工资】：" << endl; i >> t.wage; while (i.fail() || t.wage < 0 || t.wage>1e7) {

i.clear(); i.ignore(); cout << "请重新输入!" << endl; i >> t.wage; }

cout << "请输入【奖金】：" << endl; i >> t.bonus; while (i.fail() || t.bonus < 0 || t.bonus>1e7) {

i.clear(); i.ignore(); cout << "请重新输入!" << endl; i >> t.bonus;}

cout << "请输入【项目等级】：" << endl; i >> t.proj; cout << "请输入【主管】：" << endl; i >> t.dept; return i;}

fstream& operator<<(fstream& f, techmanager& t) {

f << t.id << " " << t.name << " " << t.intro << " " << t.age << " " << t.wage << " " << t.sex << " " << t.password << " " <<t.score<<" " << t.bonus << " " << t.proj<<" "<<t.dept;

return f; }

fstream& operator>>(fstream& f, techmanager& t) {

f >> t.id; f.ignore(); f >> t.name; f.ignore();

f >> t.intro; f.ignore(); f >> t.age; f.ignore();

f >> t.wage; f.ignore(); f >> t.sex; f.ignore();

f >> t.password; f.ignore(); f >> t.score; f.ignore();

f >> t.bonus; f.ignore(); f >> t.proj; f.ignore(); f >> t.dept; return f;}

istream& operator>>(istream& i, board& b) {

cout << "请输入【姓名】：" << endl; i >> b.name;

cout << "请输入【年龄】：" << endl; i >> b.age; while (i.fail() || b.age < 0 || b.age>70) {

i.clear(); i.ignore(); cout << "请重新输入!" << endl; i >> b.age; }

cout << "请输入【性别】：" << endl; i >> b.sex;

cout << "请输入【工资】：" << endl; i >> b.wage; while (i.fail() || b.wage < 0 || b.wage>1e8) {

i.clear(); i.ignore(); cout << "请重新输入!" << endl; i >> b.wage; }

cout << "请输入【奖金】：" << endl; i >> b.bonus; while (i.fail() || b.bonus < 0 || b.bonus >1e8) {

i.clear(); i.ignore(); cout << "请重新输入!" << endl; i >> b.bonus; }

cout << "请输入【项目等级】：" << endl; i >> b.proj;

return i; }

fstream& operator<<(fstream& f, board& b) {

f << b.id << " " << b.name << " " << b.intro << " " << b.age << " " << b.wage << " " << b.sex << " " << b.password<< " " << b.bonus<< " " << b.proj;

return f; }

fstream& operator>>(fstream& f, board& b) {

f >> b.id; f.ignore(); f >> b.name; f.ignore();

f >> b.intro; f.ignore(); f >> b.age; f.ignore();

f >> b.wage; f.ignore(); f >> b.sex; f.ignore();

f >> b.password; f.ignore(); f >> b.bonus; f.ignore();

f >> b.proj; return f; }

int login(board\* b,techmanager\* t,coder\* c) {//登录函数

fstream lfile("as.txt");

try {int err = 1; if (!lfile) throw err;}

catch (int) { cerr << "文件出错，请检查。" << endl; exit(0);}

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

cout.fill(' '); cout.setf(ios::right);

cout << setw(50) << "++++++++++++++++++++++++" << endl

<< setw(48) << "中文版企业人事系统" << endl << endl

<< setw(50) << "++++++++++++++++++++++++" << endl << endl

<< setw(51) << "##########################" << endl << endl

<< setw(43) << "欢迎使用！" << endl << setw(47) << "请输入工号、密码：" << endl

<< setw(51) << "##########################" << endl;

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

char identity[10] = { '\0' }; char code[10] = { '\0' };

cin >> identity; cin >> code;

while (!lfile.eof()) {

switch (lfile.peek()) {//根据每行首字母判断人员类型

case 66: lfile >> (\*b); //传地址，用登录者对应信息赋值，下同

if (!strcmp(identity,(\*b).id)){//比较工号，下同

if (!strcmp(code, (\*b).password)) {//比较密码，下同

lfile.close();return 1;}}break;

case 84: lfile >>(\*t);

if (!strcmp(identity, (\*t).id)) {

if (!strcmp(code, (\*t).password)) {

lfile.close();return 2;}}break;

case 67: lfile >>(\*c);

if (!strcmp(identity, (\*c).id)) {

if (!strcmp(code, (\*c).password)) {

lfile.close();return 3;}} }

lfile.getline(bin, 1000);}//跳转到下一行

cout << "账号或密码错误！" << endl; system("cls");

lfile.close(); return 0;}

board B; techmanager T; coder C; int choose;

void showall() {

fstream iof("as.txt", ios::in | ios::out); coder C; techmanager T; board B;

while (!iof.eof()) {

switch (iof.peek()) {

case 67: iof >> C; C.print(); cout << "===================" << endl; break;

case 84: iof >> T; T.print(); cout << "===================" << endl; break;

case 66: iof >> B; B.print(); cout << "===================" << endl;}

iof.getline(bin, 1000);} iof.close(); }

void anew() {

coder C; techmanager T; board B;

cout << "请选择要新增的人员类型：" << endl << "【1】董事会成员" << endl << "【2】技术负责人" << endl << "【3】程序员" << endl;

cin >> choose; while (cin.fail() || (choose !=1 && choose != 2 && choose !=3))

{cin.clear();cin.ignore();cout << "请重新输入!" << endl;cin >> choose;}

fstream iof("as.txt", ios::in|ios::out); try { int err = 1; if (!iof) throw err;}

catch (int) { cerr << "文件出错，请检查。" << endl; exit(0); }

int n[100] = { 0 }; char IDc[10] = { 'C' }; char IDt[10] = { 'T' }; char IDb[10] = { 'B' }; int mx; char t[10] = { '\0' };

switch (choose) {

case 3: cin >> C; for (int j = 0; !iof.eof(); j++) {

if (iof.peek() == 67) {

for (int k = 4; k >= 0; k--) { iof.seekg(1, ios::cur); n[j] += pow(10, k) \* (iof.peek() - 48); }}

iof.getline(bin, 1000);} iof.close(); mx = max(n);

for (int j = 4; j >= 0; j--) {t[ j] = mx % 10 + 48; mx/=10; }

strcat\_s(IDc, t); strcpy\_s(C.id, IDc);

system("cls"); C.print(); C.getpassword(); iof.open("as.txt", ios::app); iof << C

<< " "<< endl; iof.close(); break;//预留修改空间，下同

case 2: cin >> T; for (int j = 0; !iof.eof(); j++) {

if (iof.peek() == 84) {

for (int k = 4; k >= 0; k--) { iof.seekg(1, ios::cur); n[j] += pow(10, k) \* (iof.peek() - 48); }}

iof.getline(bin, 1000);} iof.close(); mx = max(n);

for (int j = 4; j >= 0; j--) { t[j] = mx % 10 + 48; mx /= 10; }

strcat\_s(IDt, t); strcpy\_s(T.id, IDt);

system("cls"); T.print(); T.getpassword(); iof.open("as.txt", ios::app); iof << T

<< " "<< endl; iof.close(); break;

case 1: cin >> B; for (int j = 0; !iof.eof(); j++) {

if (iof.peek() == 66) {

for (int k = 4; k >= 0; k--) { iof.seekg(1, ios::cur); n[j] += pow(10, k) \* (iof.peek() - 48); }}

iof.getline(bin, 1000); } iof.close(); mx = max(n);

for (int j = 4; j >= 0; j--) { t[j] = mx % 10 + 48; mx /= 10; }

strcat\_s(IDb, t); strcpy\_s(B.id, IDb);

system("cls"); B.print(); B.getpassword(); iof.open("as.txt", ios::app); iof << B

<< " " << endl; iof.close(); }}

**int main() {**

**system("color 1F");//设置背景颜色，美观**

**L:switch (login(&B, &T, &C)) {**

**case 0: goto L;//工号或密码错误**

**case 1://B类**

**L1: B.show(); cin >> choose;**

**while (cin.fail() || choose > 7 || choose < 1)**

**{cin.clear();cin.ignore();cout << "请重新输入!" << endl;cin >> choose;}**

**system("cls"); switch (choose) {**

**case 1: B.print(); B.getpassword(); system("pause"); goto L1;**

**case 2: showall(); system("pause"); goto L1;**

**case 3: B.alt(); system("pause"); goto L1;**

**case 4: anew(); system("pause"); goto L1;**

**case 5: B.rank(); system("pause"); goto L1;**

**case 6: goto L;**

**case 7: cout << "欢迎您再次使用，再见！" << endl; exit(0); }**

**case 2://T类**

**L2: T.show(); cin >> choose;**

**while (cin.fail() || choose > 7 || choose < 1){**

**cin.clear(); cin.ignore();**

**cout << "请重新输入!" << endl; cin >> choose; }**

**system("cls"); switch (choose) {**

**case 1: T.print(); T.getpassword(); system("pause"); goto L2;**

**case 2: showall(); system("pause"); goto L2;**

**case 3: T.alt(); system("pause"); goto L2;**

**case 4: anew(); system("pause"); goto L2;**

**case 5: T.rank(); system("pause"); goto L2;**

**case 6: goto L;**

**case 7: cout << "欢迎您再次使用，再见！" << endl; exit(0);}**

**case 3://C类**

**L3: C.show(); cin >> choose;**

**while (cin.fail() || choose > 4 || choose < 1)**

**{ cin.clear(); cin.ignore();**

**cout << "请重新输入!" << endl; cin >> choose; }**

**system("cls"); switch (choose) {**

**case 1: C.print(); C.getpassword(); system("pause"); goto L3;**

**case 2: C.alt(); goto L3;**

**case 3: goto L;**

**case 4: cout << "欢迎您再次使用，再见！" << endl; exit(0);}}}**

附录 评分表

第 1 题评分标准

|  |  |  |
| --- | --- | --- |
| 项 目 | 评 价 | |
| 设计方案的合理性与创新性 | **6** |  |
| 设计与调试结果 | **8** |  |
| 设计说明书的质量 | **2** |  |
| 程序基本要求涵盖情况 | **8** |  |
| 程序代码编写素养情况 | **4** |  |
| 课程设计周表现情况 | **2** |  |
| 综合成绩 | **30** |  |