# 程序设计基础2课程设计任务书

## 一 题目

公司人员管理系统

## 二 目的要求

本课程设计在最后应该包括两部分内容：一部分是计算机上实现的系统源程序（可运行的），另一部分是课程设计报告。

计算机上所实现的系统应能基本实现所选题目的功能要求，能正常演示。提倡所设计的系统具有一定的拓展和创新。学生应对所实现的程序熟练掌握。

总结报告包括需求分析、总体设计、详细设计、编码(重要功能部分)、测试的步骤和内容、课程设计总结、参考资料等。

## 三 主要内容及技术要求

主要内容：

创建了两个类来实现程序要求，1为People类，定义了一个人员的各种信息，2为Main类，定义了各种操作函数，main函数里用switch来选择操作函数。实现了以下功能

1）人员编号在生成人员信息时同时生成，每输入一个人员信息编号顺序加1；

2）程序对所有人员有提升级别的功能；

3）月薪的计算方法是：经理拿固定月薪，兼职技术人员按工作小时数领取月薪，兼职推销员的报酬按该推销员当月销售额提成 ,销售经理既拿固定月薪也领取销售提成；

4）能显示全部人员信息。

5）能按姓名或者编号显示、查找、增加、删除或者删除全部人员和保存各类人员的信息。

6）显示负责人联系方式。

7）信息整合保存成a.txt文件。

8）读取a.txt文件。

所采用的技术：

编译环境：CodeBlocks

面向对象的相关技术：

1封装

2重载

3用vector容器装People类成员，二进制文本文件的存储与读取。

4 system("mode con cols=150 lines=30");

system("color 3f");

来改变黑框的构造

5 system("cls") 来清空画面

6 switch()来选择操作的函数

7 system("pause"); 按任意键继续

8 getchar(); 吸收键盘输入

## 四 主要参考

传智播客高教产品研发部. C++程序设计教程[M]. 北京:人民邮电出版社,2015.

程 序 设 计 报 告 单

# 课程设计报告

# **功能描述：**

公司人员管理系统

人员信息修改管理

人员信息查询管理

人员文件存储管理

负责人联系方式

月底工资账单总览

查找人员及其个人信息

增加人员及其个人信息

删除人员及其个人信息

删除全部人员及其个人信息

升职操作

信息整合保存成a.txt文件

读取a.txt文件

**（2）类的设计**：依据功能描述进行类的设计，即给出类的定义（介绍类中所包含的数据成员和成员函数的具体意义）。

# 设计了两个类，1为People类 数据成员：

# string job; //职位

# string name; //姓名

# int number; //编号

# int money; //工资

static int sum; //公司人数

成员函数：

1 People() {} //重载构造函数

2 People(string job1, string name1,int money1,int number1)

{

job=job1;

name=name1;

number=number1;

money=money1;

} //构造一个People类

3 ~People(); //析构一个People类

# 2为Main类 数据成员：

vector<People> Peo; //用容器来存放People成员 易于操作，比类数组要好

成员函数：

程 序 设 计 报 告 单

1 Main() {} //构造函数

2 void Promote(); //升职函数 查找人员名字来操作升职

3 void Show(); //显示总信息

4 void Look(); //根据名字来查找 成员

5 void Increase(); // 增加一个成员

6 void Delete(); //根据名字来删除一个成员

7 void all(); //删除全部人员

8 void write(); //将数据存入txt文件

9 void read(); //从txt文件中读取内容

10 int Sum() // //显示公司人数

1. **代码实现**

#include <iostream>

#include <iomanip>

#include <fstream>

#include <cstdio>

#include <cstdlib>

#include <vector>

#include <cstring>

#define four 20000;

#define three 15000;

#define two 8000;

#define one 4000;

using namespace std;

class People //定义一个人员的类

程 序 设 计 报 告 单

{

public:

string job; //职位

string name; //名字

int number; //编号

int money; //工资

People() {}

People(string job1, string name1,int money1,int number1) //构造函数初始化一个人员

{

job=job1;

name=name1;

number=number1;

money=money1;

}

~People();

static int sum; //这是static类型的sum 控制编号数字

};

People::~People()

{

}

class Main //总的控制台 控制人员；人员存放在vector的容器中 通过内部函数进行各种操作

{

protected:

vector<People> Peo;

public:

Main() {}

void Promote(); //升职函数

void Show(); //显示总信息

void Look(); //查找

void Increase(); //增加一个成员

void Delete(); //删除一个成员

void all();

void write(); //讲数据存入txt文件

void read();

int Sum() //显示公司人数

{

return Peo.size();

程 序 设 计 报 告 单

}

};

void Main::read() //读取

{

// char www[200];

cout<<"\* 融资股份有限公司人员管理系统 \*"<<endl;

ifstream ifs("C:\\Users\\Administrator\\Desktop\\a.txt",ios::in|ios::binary);

if(ifs.peek()==EOF)

cout<<"文件为空"<<endl;

if(ifs)

{

People Peo1;

// ifs.getline(www,200,'\*');

// cout<<www<<endl;

for(int i=0; i<Peo.size(); i++)

{

ifs.read(reinterpret\_cast<char\*>(&Peo1),sizeof(Peo1));

cout<<"名字："<<Peo1.name<<setw(15)<<"编号："<<Peo1.number<<setw(15)<<"职务："

<<Peo1.job<<setw(15)<<"工资："<<Peo1.money<<endl;

}

}

}

void Main::all() //删除全部

{

vector <People>::iterator w;

if(Peo.size()==0)

cout<<"删除失败\n";

else

{

cout<<"总人数是"<<Peo.size()<<endl;

for(w=Peo.begin(); w!=Peo.end(); w++)

{

w=Peo.erase(w);

w--;

}

程 序 设 计 报 告 单

cout<<"删除成功"<<endl;

}

}

void Main::write() //存入txt文件模块

{

// string str="月底工资账单总览\*";

ofstream ofs;

ofs.open("C:\\Users\\Administrator\\Desktop\\a.txt",ios::out|ios::binary); //打开txt文件，操作是每次打开删除之前的数据

if(ofs)

{

if(Peo.size()==0)

cout<<"操作失败"<<endl;

else

{

// ofs.write(str.c\_str(), strlen(str.c\_str()));

// ofs.flush();

for(int i=0; i<Peo.size(); i++)

{

ofs.write(reinterpret\_cast<char\*>(&Peo[i]),sizeof(Peo[i]));

ofs.flush();

}

cout<<"copy成功"<<endl;

ofs.close(); //每次打开文件后最后都要关掉，不然存不进去

}

}

else

cout<<"文件打开失败";

}

void Main::Promote() //升职模块

{

string name1; //升职人员的名字 偷懒了 name后面加了个1就算新的名字

string zhiwu; //所升的职务

int money1,numb,j; //升职后的工资 升职人数 和for循环所用的一个变量j

cout<<"请输入要升职的人数："<<endl;

cin>>numb;

if(numb==0||Peo.size()==0||numb>Peo.size())

程 序 设 计 报 告 单

{

cout<<"操作失败\n";

return;

}

for(int i=1; i<=numb; i++)

{

cout<<"请输入第"<<i<<"升职人的名字"<<endl;

cin>>name1;

for( j=0; j<Peo.size(); j++)

if(Peo[j].name==name1)

{

cout<<"请输入要升职的职位"<<endl;

cin>>zhiwu;

cout<<"请输入职位的工资"<<endl;

cin>>money1;

Peo[j].job=zhiwu;

Peo[j].money=money1;

}

if(j==Peo.size()) //没有此人的判断条件有三个 1.输入的人数为0 2.输入的人数超过总人数 3.输入的名字没有存在

cout<<"没有此人"<<endl;

}

}

void Main::Show() // 显示数据模块

{

cout<<" 月底工资账单总览 "<<endl;

for(int i=0; i<Peo.size(); i++)

{

cout<<"名字："<<Peo[i].name<<setw(15)<<"编号："<<Peo[i].number<<setw(15)<<"职务："

<<Peo[i].job<<setw(15)<<"工资："<<Peo[i].money<<endl;

}

if(Peo.size()==0)

cout<<"无人员"<<endl;

}

void Main::Look() //查找模块

{

string name1;

cout<<"请输入要查找人的姓名:"<<endl;

程 序 设 计 报 告 单

cin>>name1;

for(int i=0; i<Peo.size(); i++)

if(Peo[i].name==name1)

{

cout<<"名字："<<Peo[i].name<<setw(15)<<"编号："<<Peo[i].number<<setw(15)<<"职务："

<<Peo[i].job<<setw(15)<<"工资："<<Peo[i].money;

}

cout<<"没有此人"<<endl;

}

void Main::Increase() //增加人员模块

{

string name1;

string job1;

int money1;

int numb;

cout<<"请输入要增加的人数："<<endl;

cin>>numb;

for(int i=1; i<=numb; i++)

{

cout<<"请输入第"<<i<<"个人的名字："<<endl;

cin>>name1;

cout<<"请输入职务："<<endl;

cin>>job1;

cout<<"请输入工资："<<endl;

cin>>money1;

People::sum++;

People ch(job1,name1,money1,People::sum);

Peo.push\_back(ch);

}

}

void Main::Delete() //删除模块

{

string name1;

int numb,j,sum;

cout<<"请输入要删除的人数："<<endl;

cin>>numb;

if(numb==0||Peo.size()==0||numb>Peo.size())

{

cout<<"操作失败\n";

return;

程 序 设 计 报 告 单

}

for(int i=1; i<=numb; i++)

{

cout<<"请输入要删除的第"<<i<<"个人的姓名："<<endl;

cin>>name1;

for(j=0; j<Peo.size(); j++)

{

sum=Peo.size();

if(Peo[j].name==name1)

{

Peo.erase(Peo.begin()+j);

cout<<"删除成功"<<endl;

break;

}

}

if(j==sum)

cout<<"没有此人"<<endl;

}

}

int People::sum=0;

int main()

{

system("mode con cols=150 lines=30");

system("color 3f");

Main M;

while(1)

{

system("cls");

cout<<"\*-----------------------------------------------------------------------------------------------------------------------\*\n";

cout<<"\*注：经理固定月薪：20000￥"<<setw(15)<<"销售经理:15000￥"<<setw(15)<<"兼职技术人员:8000￥"<<setw(15)<<"兼职推销员:4000￥ \*"<<endl;

cout<<"\* 融资股份有限公司人员管理系统 \*"<<endl;

cout<<"\* 月底工资账单总览 \*"<<endl;

cout<<"\* 1/增加人员及其个人信息 \*"<<endl;

cout<<"\* 2/删除人员及其个人信息

程 序 设 计 报 告 单

\*"<<endl;

cout<<"\* 3/删除全部人员及其个人信息 \*"<<endl;

cout<<"\* 4/查找人员及其个人信息 \*"<<endl;

cout<<"\* 5/升职操作 \*"<<endl;

cout<<"\* 6/公司人员总数 \*\n";

cout<<"\* 7/月底工资账单总览 \*"<<endl;

cout<<"\* 8/负责人联系方式 \*"<<endl;

cout<<"\* 9/信息整合保存成a.txt文件 \*"<<endl;

cout<<"\* A/读取a.txt文件 \*"<<endl;

cout<<"------------------------------------------------------------------------------------------------------------------------\*"<<endl;

char num=getchar(); //getchar（）吸收键盘输入的字符数字 便可实现人机交互，可选择操作

switch(num)

{

case '1':

M.Increase();

break;

case '2':

M.Delete();

break;

case '3':

M.all();

break;

case '4':

M.Look();

break;

case '7':

M.Show();

break;

case '8':

cout<<"微信：15968628321；手机号：15968628321"<<endl;

break;

程 序 设 计 报 告 单

case '9':

M.write();

break;

case '5':

M.Promote();

break;

case '6':

cout<<"公司总人数:"<<M.Sum()<<endl;

break;

case 'A':

M.read();

default:

break;

}

system("pause"); //按任意键继续，实现了画面冻结

getchar(); //这个getchar（）是为了吸收删一个输入的回车键，防止下次循环中num吸收了这个回车键

}

}

**效果展示**

**1显示系统界面：**

**系统运行后，显示运行界面，如图：**

**选择1/增加人员及其个人信息**

**2/删除人员及其个人信息**

**3/删除全部人员及其个人信息**

**4/查找人员及其个人信息**

**5/升职操作**

**6/公司人员总数**

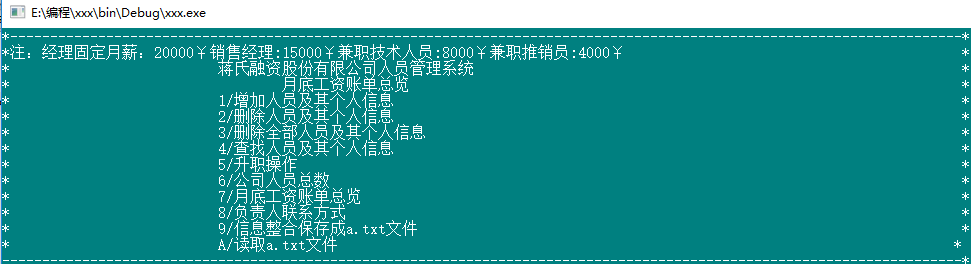
**7/月底工资账单总览**

程 序 设 计 报 告 单

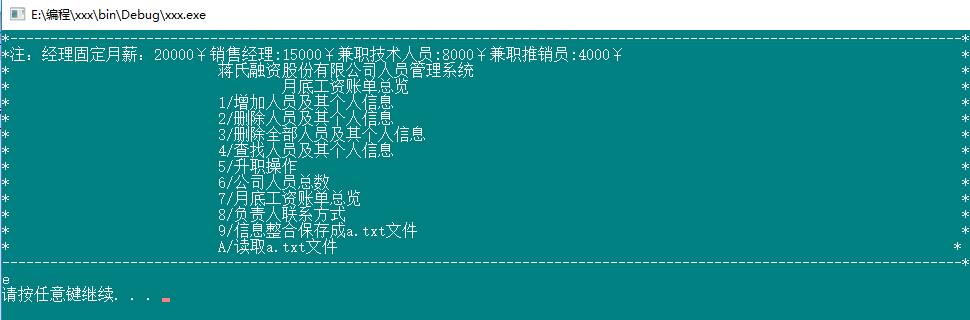
**8/负责人联系方式**

**9/信息整合保存成a.txt文件**

**A/读取a.txt文件**

****

**在系统功能界面选择操作序列，如果输入的数字不在序列里，则会报错，如图所示：**

****

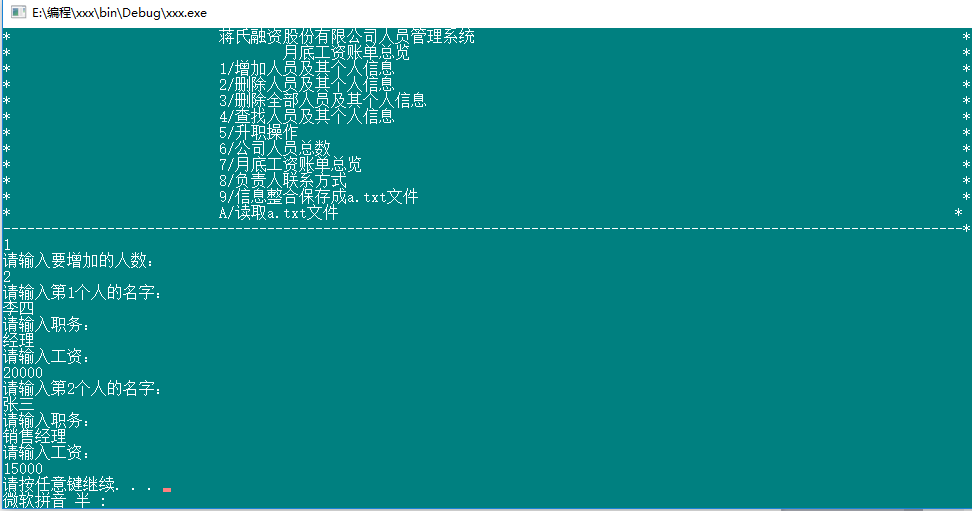
1. **人员信息修改管理：**

**在系统功能界面选择1，显示增加操作 如图所示：**

**首先输入要操作的人数 然后依次输入第几个人的名字，职务和工资。**

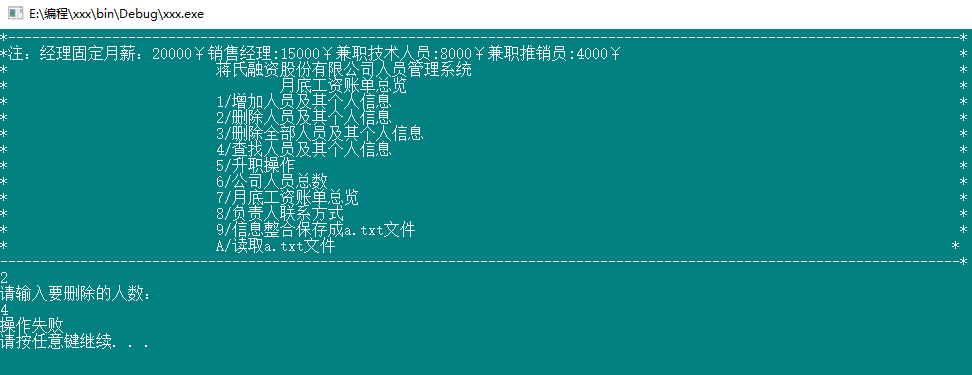
**然后按任意键继续。**

程 序 设 计 报 告 单

****

**选择2进入删除操作界面，如果输入的数字大于实际数据则会报错，按任意键继续**

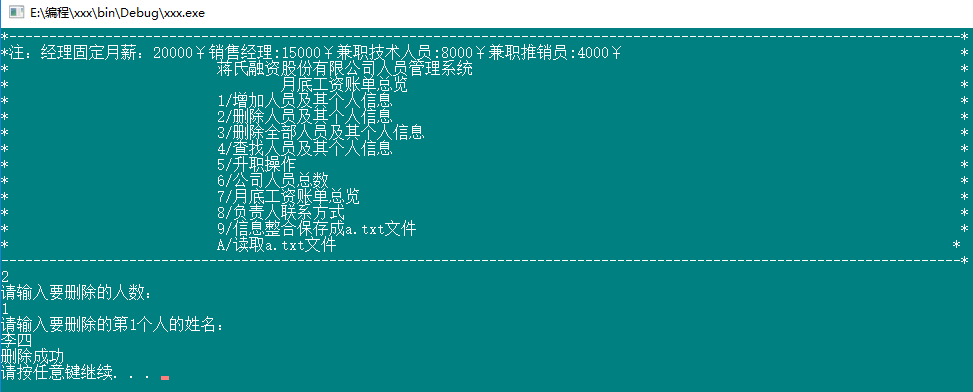
**如图所示：**

****

**输入正确数字后进行实际的删除操作，按任意键继续；**

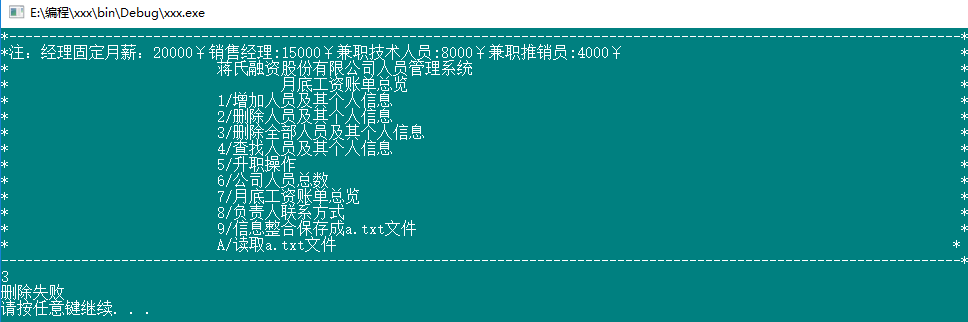
**如图所示：**

程 序 设 计 报 告 单

****

**选择3是执行删除全部人数的操作，如果人数为0时，操作失败，按任意键继续；**

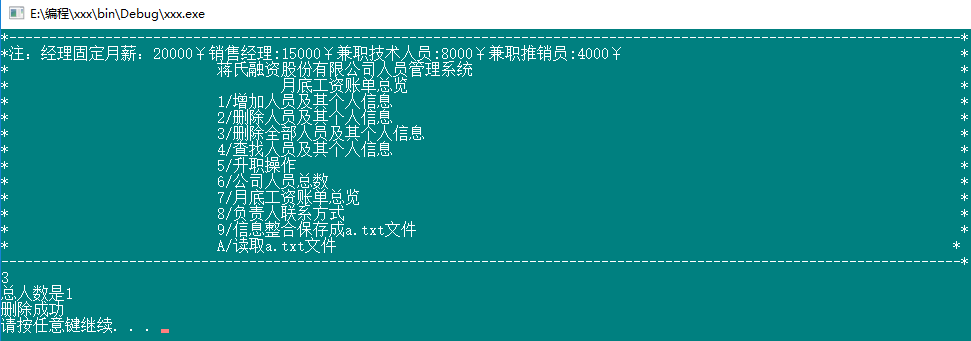
**如图所示：**

****

**选择3 如果人数大于0，人员全部删除 ，按任意键继续；**

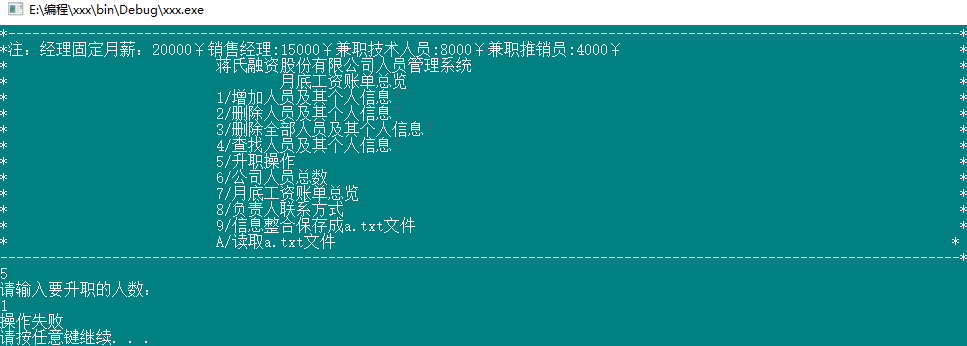
**如图所示：**

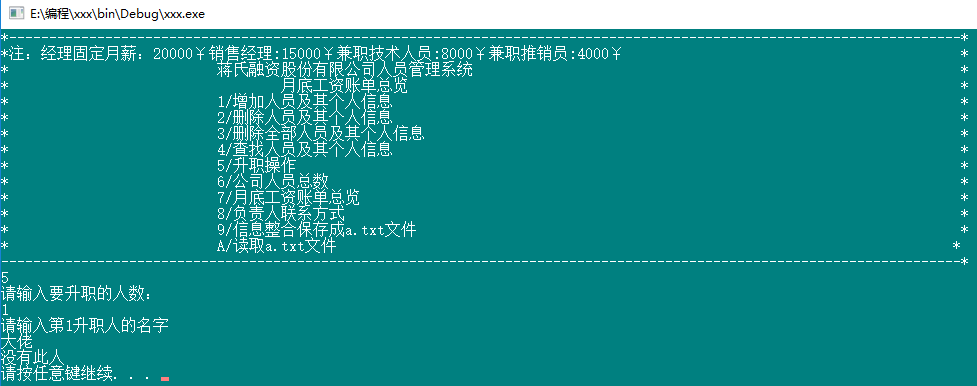
程 序 设 计 报 告 单

****

**选择5，进入升职界面。如果输入人数大于实际人数则会报错，如果输入人 员不在内存也会报错，按任意键继续；**

**如图所示：**

****

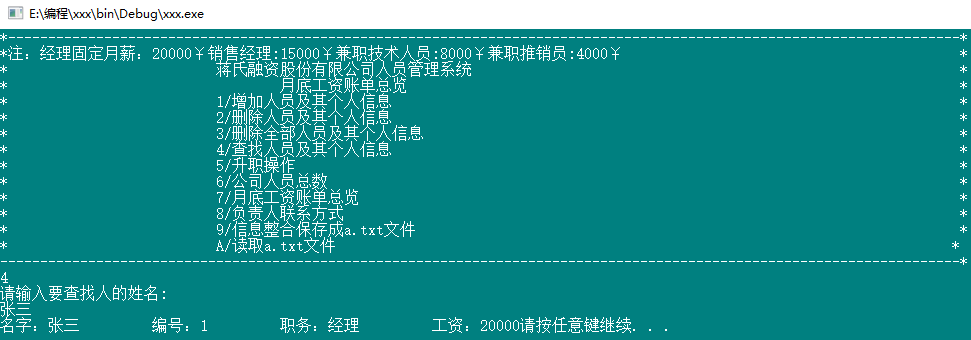
****

程 序 设 计 报 告 单

**3.人员信息查询管理：**

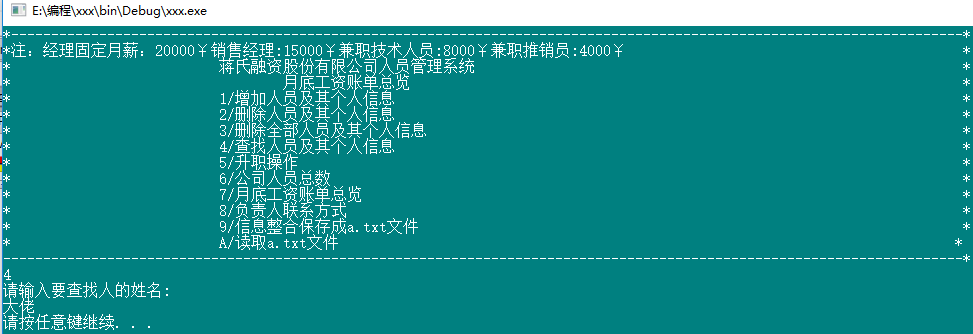
**选择4，进入查询界面，按任意键继续；**

**如图所示：**

****

**选择4，如果查询的人员不在内存里 则会操作失败，按任意键继续；**

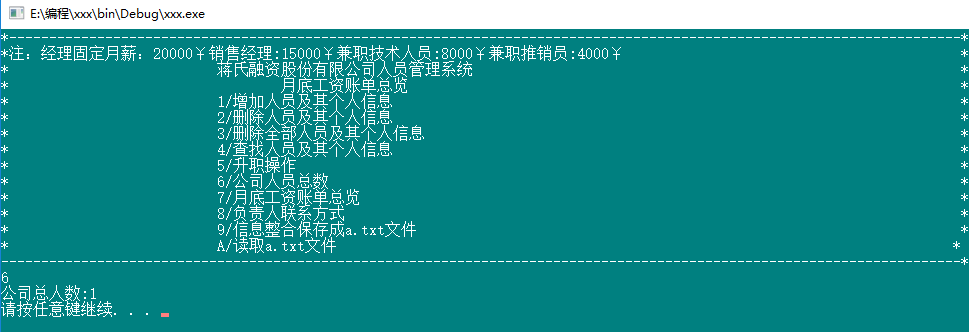
**如图所示：**

****

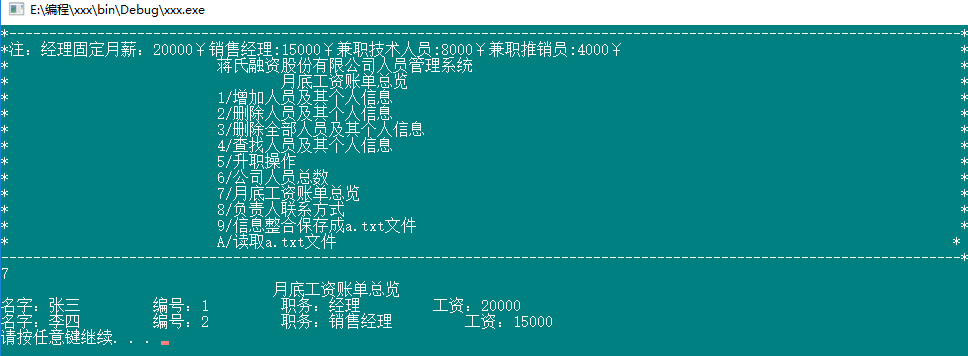
**选择6，显示实际人数,按任意键继续；**

程 序 设 计 报 告 单

**如图所示：**

****

**选择7，显示月底工资账单总览，如图所示：**

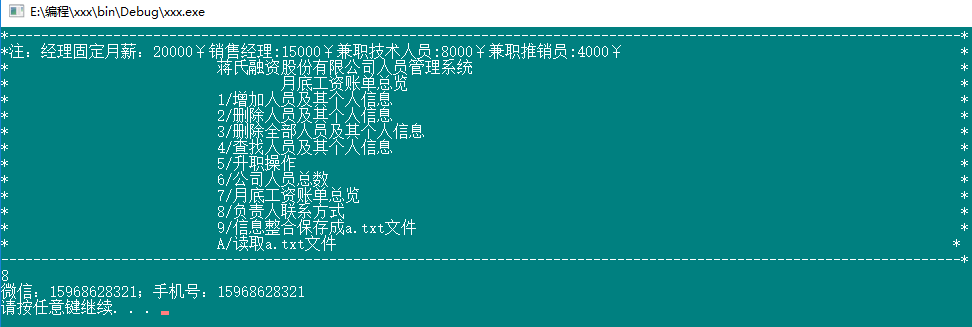
****

**按任意键继续；**

**选择8，显示负责人联系方式（我自己的），按任意键继续，**

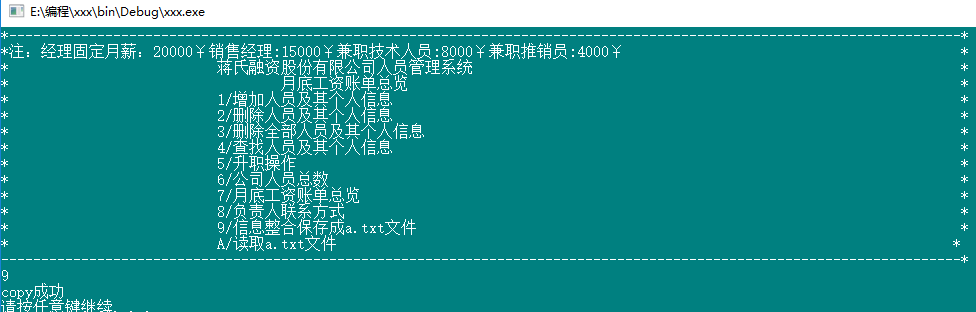
**如图所示：**

程 序 设 计 报 告 单

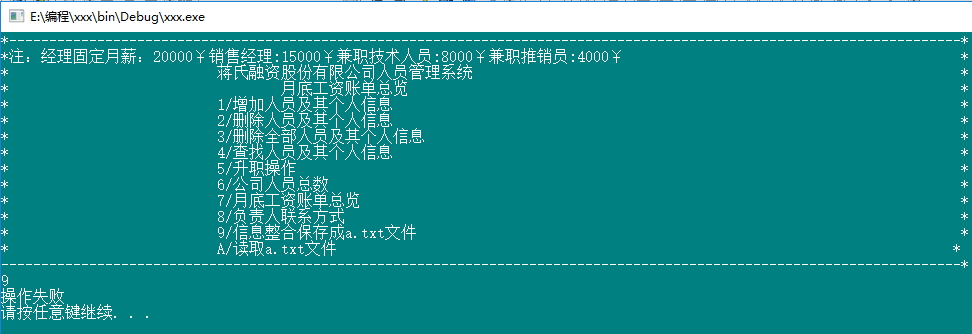
****

**4.人员信息文件管理：**

**选择9，将内存中的数据存入a.txt外部文档，如图所示:**

****

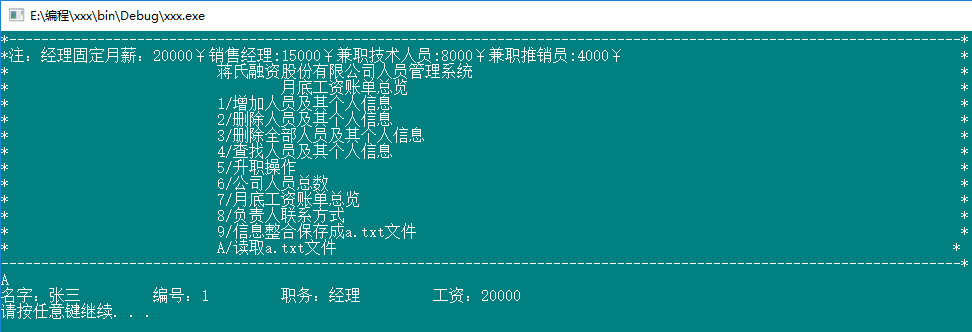
**选择9 ，如果内存中无数据 ，则存储失败，如图显示：**

****

**按任意键继续；**

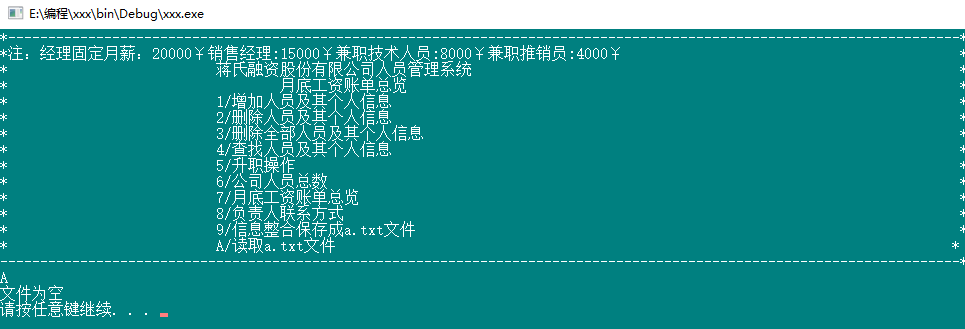
程 序 设 计 报 告 单

**选择A，读取a.txt文件中的内容，如图所示：**

****

**按任意键继续；**

**如果操作A，如果文件不存在或者为空，则会报错，如图所示：**

****

**按任意键继续；**

1. **小结**

1 每次操作完要用 cls 进行画面清除 保持了界面的整洁和易于操作。

2 用getchar（）来吸收键盘输入的字符。

3 用switch（）函数来进行选择操作的函数

程 序 设 计 报 告 单

4 system("pause");按任意键继续，实现了画面冻结

5 最后也要加getchar（）； 这个getchar（）是为了吸收删一个输入的 回车键，防止下次循环中num吸收了这个回车键

6 存放类成员用容器比较方便，易于操作

7 可以用 system("mode con cols=150 lines=30"); system("color 3f");这样的函数来美化界面