**软 件 与 微 电 子 学 院**

数据结构大作业报告

**题 目： 员工信息管理系统**

**班 级： 软件19-3**

**姓 名： 付艺泽**

**学 号： 1914010303**

**一、需求分析与功能描述**

**需求分析**

某公司为了方便公司的日常管理以及人员流动,决定设计一套员工管理系统需要统计员工的姓名,工号,岗位,工资,并且能进行录入员工信息,浏览员工信息,插入员工信息,查询员工信息,删除员工信息,按照工资排序等功能.

**功能描述:**

1输入功能

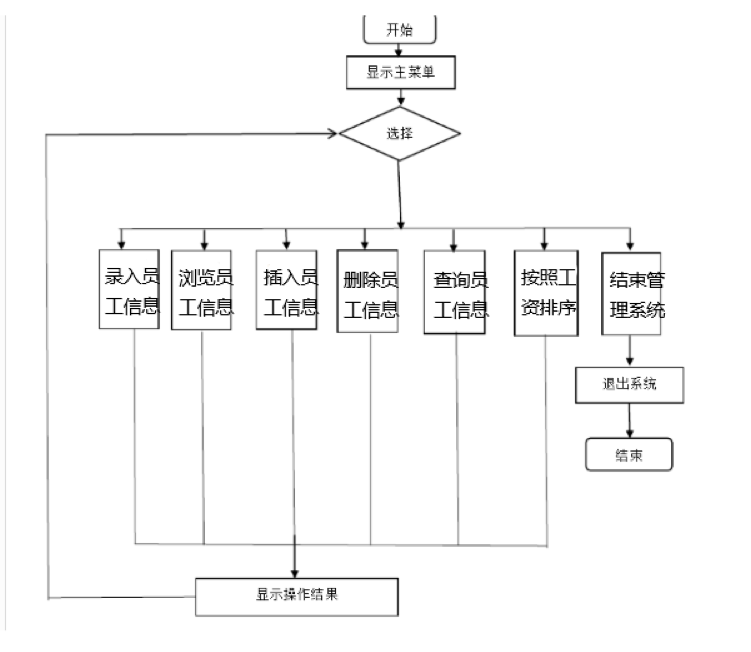
 输入员工信息并保存  
2.浏览功能  
 列出所有记录（每项记录只列出姓名,工号,工资,岗位）；  
 可选定一项记录，显示所有域；  
3. 查询功能  
 通过姓名和工号查询；  
4. 插入功能  
 增加新的联系人及其信息；  
5. 删除功能  
 删除已有联系人及信息，提供确认机制；

6.排序功能

 按照工资的高低,将员工信息由低到高排序

7.退出

退出管理系统



“二、数据结构的选择与设计”中，介绍面向的数据是什么、数据元素与数据项；通过分析数据元素间关系再结合需求，确定了哪类数据结构；介绍由于什么原因选择该数据结构的存储结构（比如顺序结构或链式结构），进而描述数据结构的设计（包括逻辑结构定义及操作）。

**面向的数据:**

string name;//姓名

string post;//岗位

int stuffid;//工号

float salary; //工资

**选择了链式存储结构单链表,** **在插入和删除操作时，只需要修改被删节点上一节点的链接地址，不需要移动元素，从而改进了在顺序存储结构中的插入和删除操作需要移动大量元素的缺点,因为次系统主要的功能有插入和删除,选择了单链表.**

“三、算法的选择与设计”中，概述为了满足哪条需求，设计了哪类算法（比如查找、排序） ，进而说明为什么选择了最终这种算法（比如折半查找算法、快速排序算法），如果实现了多个算法，需分别说明；可以通过“伪代码”、“流程图”或“源代码”等形式介绍算法的结构。

**算法主要在查找和删除功能中使用了较为简单的顺序查找, 算法简洁，且对表的结构无任何要求，**无论是用向量还是用链表来存放结点，也无论结点之间是否按关键字有序，它都同样适用。

**查询:**int Company::Searchdata()

{

string xname;

int xstuffid;

cout << "输入员工姓名:";

cin >> xname;

cout << "输入工号:";

cin >> xstuffid;

Data\* p = head->next;

while (p)

{

if (xstuffid == p->stuffid)

{

if (xname == p->name)

{

return 1; //成功找到

}

return -1; //错误，工号重复

}

p = p->next;

}

return 0; //没有查询到该员工信息

}

**删除**

void Company::Deletedata()

{

string xname;

int xstuffid;

cout << "输入要删除的员工姓名:" ;

cin >> xname;

cout << "输入工号:";

cin >> xstuffid;

Data\* p = head, \* s = p->next;

while (s)

{

if (xstuffid == s->stuffid)

{

if (xname == s->name)

{

p->next = s->next;

delete s;

return;

}

cout << endl << "出错，工号与姓名不匹配" << endl;

return;

}

p = s;

s = p->next;

}

cout << "没有查找到相关员工信息,可能已离职" << endl;

this->stuffnum-1;

}

当沿着单链表查询到与表中信息符合的员工时会将该员工的所有信息删除

在排序功能中使用了起泡排序, 起泡排序比较简单，空间复杂度较低，是稳定的所以选择了起泡排序

void Company::Sequence()

{

for (Data\* a1 = this->head; a1!= NULL;a1 = a1->next)

{

for (Data\* a2 = this->head;a2->next != NULL; a2 = a2->next)

{

if (a2->salary > a2->next->salary) {

swap(a2, a2->next);

}

}

}

}

void swap(Data\* A, Data\* B)

{

Data temp;

temp.name = A->name;

A->name = B->name;

B->name = temp.name;

temp.post = A->post;

A->post = B->post;

B->post = temp.post;

temp.salary = A->salary;

A->salary = B->salary;

B->salary = temp.salary;

temp.stuffid = A->stuffid;

A->stuffid = B->stuffid;

B->stuffid = temp.stuffid;

}

“四、主要代码”中，列出除二、三所示部分外的程序核心代码。

#include <iostream>

#include <iomanip>

#define I 0

using namespace std;

typedef struct Data{

string name;//姓名

string post;//岗位

int stuffid;//工号

float salary; //工资

Data\* next;

}Data;

void swap(Data\* A,Data\* B);

class Company//公司

{

public:

Company();

Company(string);//公司名称

~Company();

/\*创建员工列表\*/

void Setup(int);

void print(); //打印员工列表

void Insertdata();//插入新的员工信息

int Searchdata();//查找。输入姓名和工号

void Deletedata();//删除，输入姓名和工号

void Sequence();//排序

int Functionselection();//功能选择

private:

Data\* head;//头节点

string Companyname;//公司名称

int stuffnum;//公司员工数

};

Company::Company()

{

head = new Data;

head->next = NULL;

this->stuffnum = 0;

this->Companyname = "育碧蒙特利尔工作室";

}

Company::~Company()

{

delete head;

}

void Company::Setup(int n)//共插入i个员工

{

Data\* s, \* p = head;

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

{

//对单个员工进行录入信息

s = new Data;

cout << "请输入第" << i + 1 << "个员工的姓名:" ;

cin >> s->name;

cout << "请输入第" << i + 1 << "个员工的工号:" ;

cin >> s->stuffid;

cout << "请输入第" << i + 1 << "个员工的岗位:" ;

cin >> s->post;

cout << "请输入第" << i + 1 << "个员工的工资:" ;

cin >> s->salary;

cout << "第" << i + 1 << "个员工输入完成\a" << endl << endl;

s->next = p->next;

p->next = s; Companyname;

p = s;

this->stuffnum++; //员工人数+1

}

}

void Company::print()

{

cout << endl;

cout << " "<< "公司名称：" << this->Companyname << endl;

cout << " " << "编号" << " " << "姓名" << " " << "工号" << " " << "工资" << " " <<"岗位"<< endl;

Data\* p = this->head->next;

int i = 1;

while (p)

{

cout << " " << i++ << " " << p->name << " " << p->stuffid << " " << p->salary<< " " <<p->post << endl;

p = p->next;

}

}

void Company::Insertdata()

{

string xname;

int xstuffid;

float xsalary;

string xpost;

cout<<"姓名:";

cin >> xname;

cout<<"工号:";

cin >> xstuffid;

cout<<"工资:";

cin >> xsalary;

cout<<"岗位:";

cin >> xpost;

Data\* s, \* p = head;

s = new Data;

s->name = xname;

s->stuffid = xstuffid;

s->post = xpost;

s->salary = xsalary;

s->next = p->next;

p->next = s;

p = s;

}

int Company::Searchdata()

{

string xname;

int xstuffid;

cout << "输入员工姓名:";

cin >> xname;

cout << "输入工号:";

cin >> xstuffid;

Data\* p = head->next;

while (p)

{

if (xstuffid == p->stuffid)

{

if (xname == p->name)

{

return 1; //成功找到

}

return -1; //错误，工号重复

}

p = p->next;

}

return 0; //没有查询到该员工信息

}

void Company::Deletedata()

{

string xname;

int xstuffid;

cout << "输入要删除的员工姓名:" ;

cin >> xname;

cout << "输入工号:";

cin >> xstuffid;

Data\* p = head, \* s = p->next;

while (s)

{

if (xstuffid == s->stuffid)

{

if (xname == s->name)

{

p->next = s->next;

delete s;

return;

}

cout << endl << "出错，工号与姓名不匹配" << endl;

return;

}

p = s;

s = p->next;

}

cout << "没有查找到相关员工信息,可能已离职" << endl;

this->stuffnum-1;

}

void Company::Sequence()

{

for (Data\* a1 = this->head; a1!= NULL;a1 = a1->next)

{

for (Data\* a2 = this->head;a2->next != NULL; a2 = a2->next)

{

if (a2->salary > a2->next->salary) {

swap(a2, a2->next);

}

}

}

}

int Company::Functionselection() //功能选择

{

char k;

do {

system("cls");

cout << " 欢迎使用 育碧蒙特利尔工作室员工管理系统\n";

cout<<" 功能选择\n";

cout<<" 1.录入员工信息\n";

cout<<" 2.浏览员工信息\n";

cout<<" 3.插入员工信息\n";

cout<<" 4.删除员工信息\n";

cout<<" 5.查询员工信息\n";

cout<<" 6.按照工资额排序\n";

cout<<" 7.退出管理系统\n";

cout <<" 你要选择的功能是(请按数字键（1~7）):";

k = getchar();

} while (k < '1' || k>'7');

return(k - '0');

}

void swap(Data\* A, Data\* B)

{

Data temp;

temp.name = A->name;

A->name = B->name;

B->name = temp.name;

temp.post = A->post;

A->post = B->post;

B->post = temp.post;

temp.salary = A->salary;

A->salary = B->salary;

B->salary = temp.salary;

temp.stuffid = A->stuffid;

A->stuffid = B->stuffid;

B->stuffid = temp.stuffid;

}

int main()

{

Company company;

int n;

string name;

for (;;)

{

switch (company.Functionselection())

{

case 1:

cout << "1.录入员工信息" << endl;

company.Setup(5);

system("pause");

break;

case 2:

cout << "2.浏览员工信息" << endl;

company.print();

system("pause");

break;

case 3:

cout << "3.插入员工信息" << endl;

company.Insertdata();

system("pause");

break;

case 4:

cout << "4.删除员工信息" << endl;

company.Deletedata();

system("pause");

break;

case 5:

cout << "5.查询员工信息:" << endl;

n=company.Searchdata();

if (n == 1)

{

cout << "确有此人"<<endl;

}

else

{

cout << "查无此人" << endl;

}

system("pause");

break;

case 6:

company.Sequence();

cout << "排序完成"<<endl;

system("pause");

break;

case 7:

cout << "信息已保存,再见" << endl;

exit(0);

break;

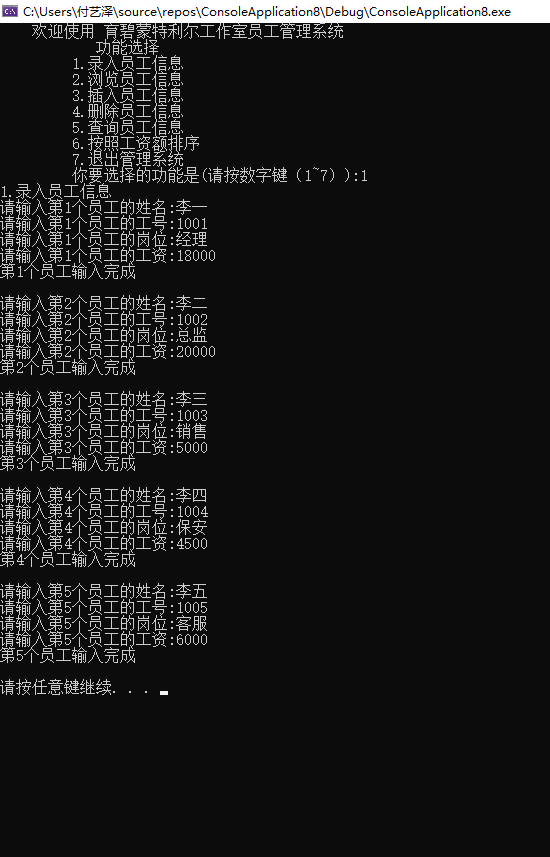
}

}

}

**五、测试结果**

录入员工信息



浏览员工信息

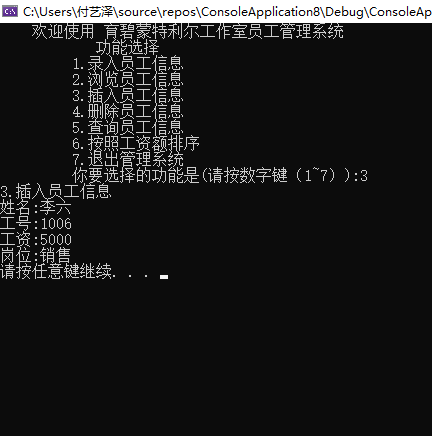


插入员工信息

插入前

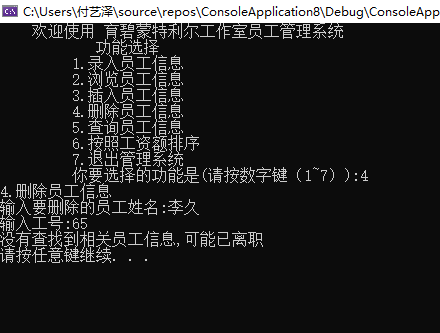


插入后





删除的失败



删除成功







查询的成功与失败





按照工资排序(排序前后)







**六.分析与总结**

本次作业基本满足了前述的需求,数据结构使用了链式存储结构单链表, 优点在插入和删除操时，只需要修改被删节点上一节点的链接地址，不需要移动元素，从而改进了在顺序存储结构中的插入和删除操作需要移动大量元素的缺点。缺点：没有解决连续存储分配带来的表长难以确定的问题,失去了顺序存储结构随机存取的特性。

顺序查找**算法简单，且对表的结构无任何要求**，无论是用向量还是用链表来存放结点，也无论结点之间是否按关键字有序，它都同样适用,缺点是效率比较低.起泡排序的优点时算法简单,缺点是效率较低.

本次作业遇到了很多的问题,但在解决问题的过程中也有了不少的收获,从选题的确认,以及代码的编写,和bug的修改,倒让我有了不小的收获,但是程序界面的设计不是很美观,程序也有些不成熟,算法也没有很精密,后续会继续努力,弥补不足,继续深入的学习数据结构.