

# 南京信息工程大学

## 《数据结构》课程设计



题    目    用双向链表建立通讯录

---

学生姓名    刘祥宇

学    号    202183290006

学    院    计算机与软件学院

专    业    计算机科学与技术

指导教师    马瑞

二〇二二 年 十二 月 二十三 日

# 目 录

(空 1 行, 五号)

<b>1 设计背景</b>	<b>1</b>
1.1 问题描述	1
1.2 问题分析	1
<b>2 设计方案</b>	<b>1</b>
2.1 算法思想	1
2.2 功能模块分析	1
2.3 功能模块实现	1
<b>3 设计结果</b>	<b>1</b>
3.1 预期结果	1
3.2 运行结果	1
<b>4 总结</b>	<b>1</b>

# 用双向链表建立通讯录

姓名：刘祥宇

南京信息工程大学计算机与软件学院，江苏 南京 210044

## 1 设计背景

### 1.1 问题描述

用双向链表作数据结构，编写一个通讯录管理系统。

每条信息至包含：姓名（NAME）街道（STREET）城市（CITY）邮编（EIP）国家（STATE）几项，且应完成一下几方面的功能：

- （1）输入信息——enter（）；
- （2）显示信息——display（）；
- （3）查找以姓名作为关键字——search（）；
- （4）删除信息——delete（）；
- （5）存盘——save（）；
- （6）装入——load（）；

### 1.2 问题分析

首先题目明确要求要用双链表进行通讯录的建立，所以要先建立一个双向链表并进行初始化；

功能方面：

输入信息是双向链表的插入；

显示信息是双向链表的输出；

查找以姓名作为关键字是双向链表的查找问题；

删除信息是双向链表的删除问题；

存盘与装入即将程序运行中的通讯录数据存入文件中；

## 2 设计方案

### 2.1 算法思想

首先定义一个链表，内容包括通讯录中人的姓名，街道，城市，邮编，国家，定义前指针和后指针；

定义全局变量头结点和尾结点并进行初始化，定义新的全局变量结点 L 指向头结点，方便函数中直接引用。

插入信息：采用头插法，建立双向链表的自编函数将新建的链表结点的尾指针指向头结点的尾指针指向的结点，将头结点的尾指针指向新建结点，将新建结点前指针指向头结点；

具体的信息的添加由新构造的 Add\_one 函数完成：通过 input 中的操作将结点作为参数传递给该函数中，定义各种数据类型，进行输入，从而完成通讯录信息的输入。

删除信息：在主函数中将要删除的人的姓名传入删除函数中，为了避免同名的人，再要求用户输入余下的信息，定义 p 结点为尾结点，利用 for 循环，对比各个结点中的所有信息，若相同，删除此节点，否则，p 等于它的前结点，p 指向 NULL 时退出循环。

查询信息：在主函数中将要查询的人的姓名传入到查询信息函数中，定义新结点 p 并将其指向头结点的下一结点，利用 for 循环，对比各个结点中的姓名，若相同，则输出，否则 p 指向它的下一节点，p 指向 NULL 时退出循环。

输出当前通讯录：定义新节点 p，并将其指向头结点的下一节点，利用 for 循环，输出结点中存储的数据，并使 p 指向下一结点，当 p->next=NULL 是退出循环.

输入到文件中和从文件中输出：使用标准化函数，建立新节点 p 使其指向头指针的下一结点，利用 for 循环，将程序中的数据输入到文件中；设置循环条件 flag=1，利用 if....else.. 对 flag 进行赋值，if 中，如果输入的不为空，则 flag=1，否则进入 else，flag=0，从而完成数据从文件输入到程序中。

### 2.2 功能模块分析

根据对系统的分析，该系统包括：选择模块，输入模块，查找模块，删除模块，输出模块，输出文件模块，输入文件模块，结构图如图 2-1 所示



图 2-1 系统功能结构图

(1) 选择模块

主要任务：对要进行的操作进行选择，通过输入的选择来调用其他功能部分；

选择为 1 时，调用输入模块；

选择为 2 时，调用删除模块；

选择为 3 时，调用查找模块；

选择为 4 时，调用输出模块；

选择为 5 时，调用输出文件模块；

如图 2-2 所示。

(2) 输入模块

主要任务：输入通讯录成员的数据；

(3) 查找模块

主要任务：根据输入的姓名，对对应的通讯录人员进行查找

(4) 删除模块

主要任务：根据输入的姓名，街道，国家，邮编，城市信息，把对应的通讯录人员删除

(5) 输出模块

主要任务：输出整张通讯录

(6) 输出文件模块

主要任务：将文件中的数据输入到程序中

(7) 输入文件模块

主要任务：将程序中的数据输入到文件中，由程序直接调用；如图 2-3 所示。

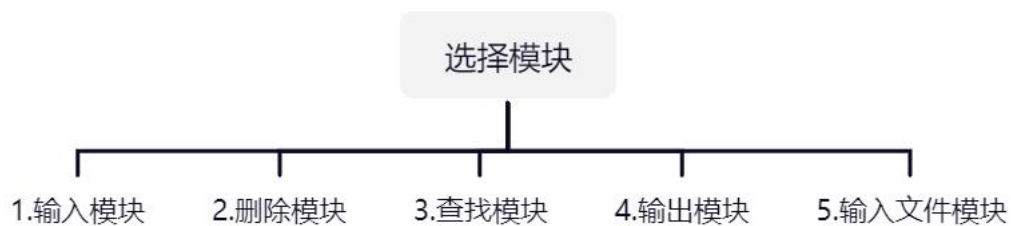


图 2-2 模块调用关系图

通讯录管理系统

文件输出模块

图 2-3 输出模块调用图

### 2.3 功能模块实现

```
void input_listfile()
{
    //读取文件操作
    ifstream ifs;
    ifs.open("address_book.txt", ios::in);
    if (!ifs.is_open()) {
        cout << "文件打开错误" << endl;
    }
    DoubleList *p, *s;
    p = &p0;
    int flag = 1;
    while (flag == 1)
    {
        s = new LNode;
        p->next = s; p = s;
        if (ifs >> s->name >> s->street >> s->city >> s->eip >> s->state >> s->num)
        {
            flag = 1;
        }
        else {
            p->next = NULL;
            flag = 0;
        }
    }
    //关闭文件
    ifs.close();
}
```

```

void display_listfiile()
{
    DoubleList *p;
    p=&p0;
    int i = 1;
    ofstream fout("address_book.txt");
    p = p->next;
    while(p->next != NULL)
    {
        fout << p->name <<endl;
        fout <<" " << p->street <<endl;
        fout <<" " << p->city <<endl;
        fout <<" " << p->eip <<endl;
        fout <<" " << p->state <<endl;
        fout <<" " <<p->num<<endl;
        fout <<endl;
        p = p->next;
        i++;
    }
    fout.close();
}

```

```

void ADD_ONE(DoubleList *F)
{
    string name;
    string street;
    string city;
    string eip;
    string state;
    string num;
    cout<<"输入姓名: ";
    cin>>name;
    cout<<"输入街道: ";
    cin>>street;
    cout<<"输入城市: ";
    cin>>city;
    cout<<"输入邮编: ";
    cin>>eip;
    cout<<"输入国家: ";
    cin>>state;
    cout<<"请输入电话号码: ";
    cin>>num;
    F->name=name;
    F->street=street;
    F->city=city;
    F->eip=eip;
    F->state=state;
    F->num=num;
}

DoubleList* ADE()
{
    DoubleList *F=new LNode,*L;
    L=&p0;
    ADD_ONE(F);
    F->next=L->next;
    L->next->prior=F;
    L->next=F;
    F->prior=L;
    return F;
}

```

```

int Delete()
{
    string name;
    string street;
    string city;
    string eip;
    string state;
    string num;
    DoubleList *F,*L;
    F=&pd;
    L=&pd;
    cout<<"请输入要删除的对象,街道,城市,邮编,国家,电话号码:<<endl;
    cin>>name>>street>>city>>eip>>state>>num;
    for(F;F->next!=NULL;F=F->next)
    {
        if(F->next->name==name&&F->next->street==street&&F->next->city==city&&F->next->eip==eip&&F->next->state==state&&F->next->num==num)
        {
            F->next=F->next->next;
            F->next->next->prior=F;
            // L->prior;
            // p->prior->next=p->next;
            // p->next->prior=p->prior;
            free(F);
            cout<<"删除成功"<<endl;
            return 0;
        }
    }
    cout<<"no find"<<endl;
    return 1;
}

void search(string name)
{
    string street;
    string city;
    string eip;
    string state;
    string num;
    int count=0;
    DoubleList *F;
    F=&pd;
    F=F->next;
    for(F;F->next!=NULL;F=F->next)
    {
        if(F->name==name)
        {
            count++;
            display(F);
            //cout<<"姓名: "<<p->name<<" "<<"街道: "<<p->street<<" "<<"城市: "<<p->city<<" "<<"邮编: "<<p->eip<<" "<<"国家: "<<p->state<<" "<<"电话号码: "<<p->num<<endl;
        }
    }
    if(count==0)
        cout<<"no find"<<endl;
}

void display(DoubleList *F)
{
    if(F==NULL)
        cout<<"no find"<<endl;
    cout<<"姓名: "<<F->name<<" "<<"街道: "<<F->street<<" "<<"城市: "<<F->city<<" "<<"邮编: "<<F->eip<<" "<<"国家: "<<F->state<<" "<<"电话号码: "<<F->num<<endl;
    //cout<<p->name<<","<<p->street<<","<<p->city<<","<<p->eip<<","<<p->state<<endl;
}

void display_list()
{
    DoubleList *F;
    F=&pd;
    F=F->next;
    int i=1;
    if(F==NULL)
        cout<<"没有通讯录,请输入信息:<<endl;
    while(F->next!=NULL)
    {
        cout<<"目前通讯录为:<<endl;
        cout<<"姓名: "<<F->name<<" "<<"街道: "<<F->street<<" "<<"城市: "<<F->city<<" "<<"邮编: "<<F->eip<<" "<<"国家: "<<F->state<<" "<<"电话号码: "<<F->num<<endl;
        F=F->next;
        i++;
    }
    cout<<endl;
}

```

## 3 设计结果

### 3.1 预期结果

先将文件中的通讯录输入到程序中，输入 5 后可以查看到与文件一致的通讯录；

出现菜单界面

输入 1 后，出现要求添加的内容指示，根据指示添加完所有数据后，出现添加成功的提示，并重新出现菜单界面；

输入 2 后，出现输入删除对象的提示，如果输入的对象通讯录中不存在则提示 no find，重新出现选择菜单，如果存在则进行删除操作，删除成功后出现删除成功提示语，重新



出现主菜单；

输入 3 后，出现输入查询对象的提示，如果查询的对象不存在，出现 no find，否则，输出查询到的所有该名字的通讯录成员。

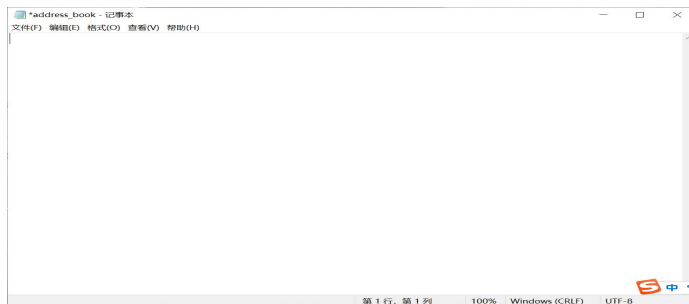
输入 4 后，输出整个通讯录，如果当前通讯录为空，则输出“当前通讯录为空”。

输入 5 后，将通讯录输入到文件中，打开对应文件，里面有程序中的数据

输入 6 后，退出程序。

### 3.2 运行结果

首先让文本为空



显示菜单

C:\Users\19942\Desktop\数据结构\通讯录问题.exe

输入要进行的操作

1. 添加 2. 删除 3. 查找 4. 输出通讯录 5. 输出到文件中 6. 退出

添加操作

C:\Users\19942\Desktop\数据结构\通讯录问题.exe

输入要进行的操作

1. 添加 2. 删除 3. 查找 4. 输出通讯录 5. 输出到文件中 6. 退出

1

输入姓名: 张三

输入街道: 盘城街道

输入城市: 南京

输入邮编: 123

输入国家: 中国

请输入电话号码: 12345

添加成功

输入要进行的操作

1. 添加 2. 删除 3. 查找 4. 输出通讯录 5. 输出到文件中 6. 退出

1

输入姓名: 李四

输入街道: 盘城街道

输入城市: 南京

输入邮编: 123

输入国家: 中国

请输入电话号码: 23456

添加成功

输入要进行的操作

1. 添加 2. 删除 3. 查找 4. 输出通讯录 5. 输出到文件中 6. 退出

查询操作

输入要进行的操作

1. 添加 2. 删除 3. 查找 4. 输出通讯录 5. 输出到文件中 6. 退出

3

输入要查找的姓名

张三

姓名: 张三 街道: 盘城街道 城市: 南京 邮编: 123 国家: 中国 电话号码: 12345

输入要进行的操作

1. 添加 2. 删除 3. 查找 4. 输出通讯录 5. 输出到文件中 6. 退出

删除操作

```
请输入要删除的对象, 街道, 城市, 邮编, 国家, 电话号码
李四
盘城街道
南京
123
中国
23456
删除成功

输入要进行的操作
1. 添加 2. 删除 3. 查找 4. 输出通讯录 5. 输出到文件中 6. 退出
4

目前通讯录为
1: 姓名: 张三 街道: 盘城街道 城市: 南京 邮编: 123 国家: 中国 电话号码: 12345

输入要进行的操作
1. 添加 2. 删除 3. 查找 4. 输出通讯录 5. 输出到文件中 6. 退出
```

把数据输入到文本中

```
输入要进行的操作
1. 添加 2. 删除 3. 查找 4. 输出通讯录 5. 输出到文件中 6. 退出
5
操作完成
```

```
address_book - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
张三
盘城街道
南京
123
中国
12345
```

## 4 总结

在本次课程设计中，完成了题目中提出的所有要求，充分利用了双向链表的特性，比如

可从尾结点开始向前遍历整个链表，完成了通讯录的输入，人员的查找，删除，通讯录的输出，和在文件中的保存这些功能。

在分析问题阶段，我对题目中的条件和要求进行了细致的分析和划分，确定了解决问题的方案；在设计功能阶段，利用结构图清楚表示了各个模块之间的关系和应该用什么代码实现；在编码阶段，利用双向链表的算法对通讯录功能的实现进行的编程；在测试阶段，运行程序后，输入测试数据，对各个功能进行测试，得到实验结果，并根据实验结果中不符合预期的地方对代码进行修改，最终得到正确的输出结构和正确的代码；

不足的地方：在查询和删除函数中，对于确定输入的姓名是否存在的问题上，没有通过定义函数来实现，造成了代码不够简洁。通讯录可以实现的功能不够多，还可以增加如根据国家统计人数的功能。

对课程设计的整个过程做概括性总结，报告各阶段所做的工作、目前的成果、有哪些不足或可以改进的地方