

**数据结构课程实验报告**

**（1）**

**姓名：王秋实**

**学号：2135070824**

**专业：计科**

**完成日期：2023/3/6**

**目录**

[目录 - 1 -](#_Toc434170429)

[1 设计要求 - 2 -](#_Toc434170430)

[2 程序功能框图 - 2 -](#_Toc434170431)

[3 数据结构说明 - 2 -](#_Toc434170432)

[4 重要算法核心代码 - 2 -](#_Toc434170433)

[5 测试运行界面 - 2 -](#_Toc434170434)

[6 完整源程序 - 3 -](#_Toc434170435)

**数据结构实验一（线性表）**

**<通讯录的制作> 设计要求**

【问题描述】

编写一个小型通讯录管理系统，完成相关人员信息的建立、显示、查找、插入、删除等操作。要有良好的界面和较强的容错性。

【基本要求】

应包含以下几方面的功能：

1. 输入信息
2. 显示信息
3. 查找信息，以姓名为关键字
4. 删除信息
5. 信息存储到磁盘文件
6. 信息从磁盘文件导入

【数据】

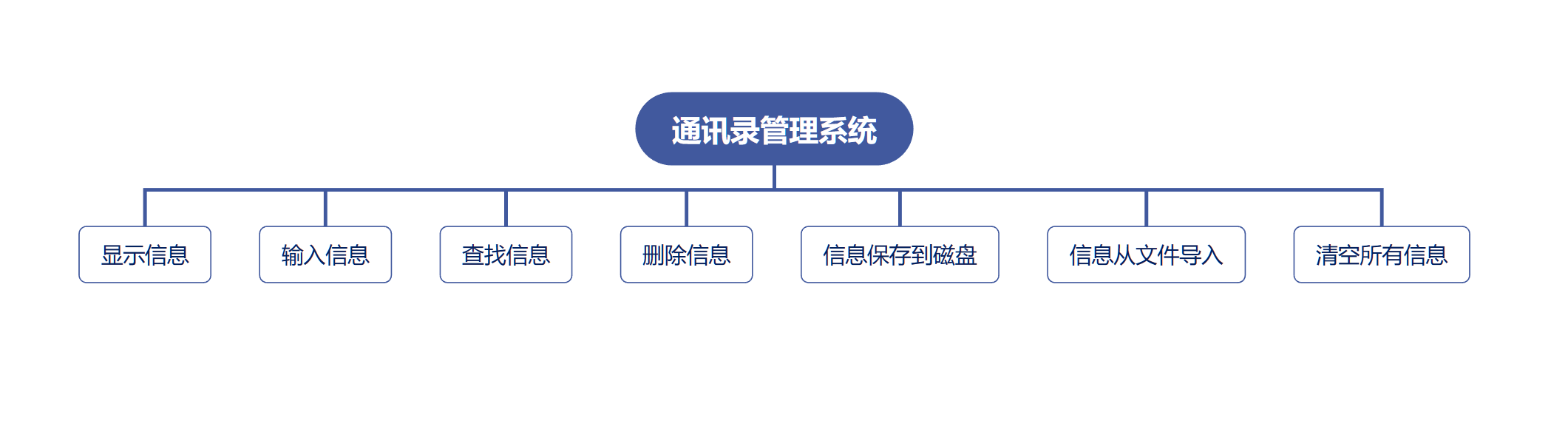
通讯录的每条信息至少包含以下几项内容：

1. 姓名
2. 街道
3. 城市
4. 国家
5. 电话号码

【实现方法】

使用双向链表进行设计。需要用到C语言中文件读写函数进行磁盘文件访问。

**程序功能框图**

****

**数据结构说明**

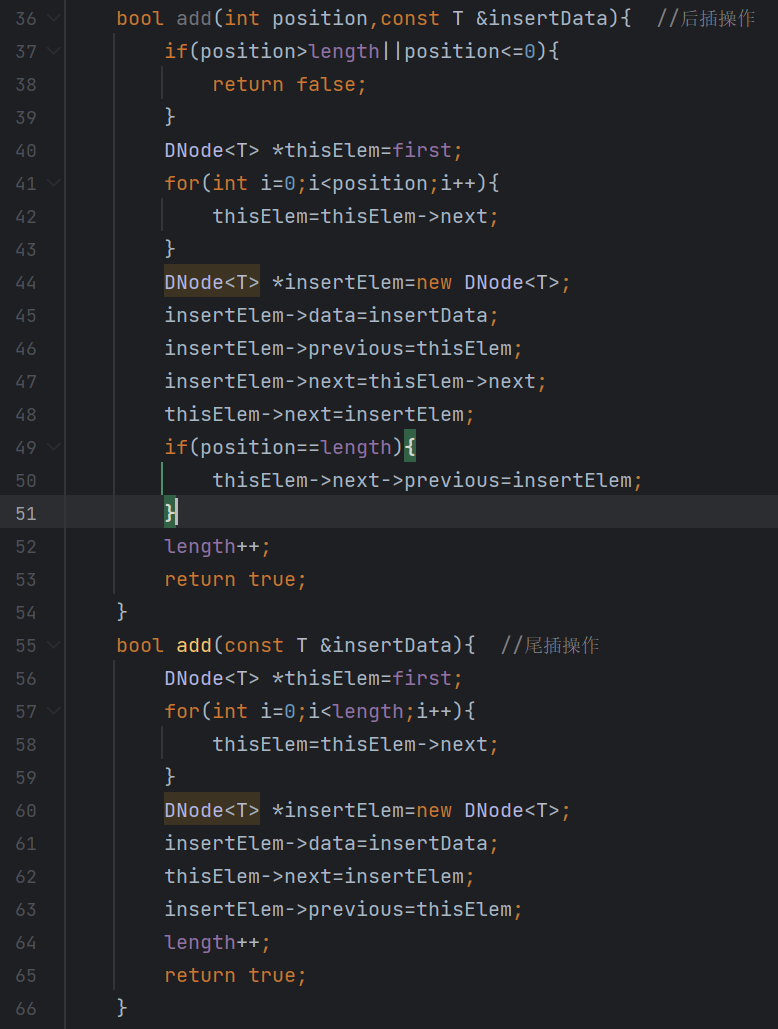
本程序使用双向链表存储，创建了DLinkedList.h头文件用于实现双向链表，其中定义了DNode数据类型和DLinkedList<T>类。前者有用于存储数据的data部分，和分别指向前驱元素和后继节点的previous和next指针。后者定义了头结点和线性表长度，构造和析构函数，print，后插和尾插add，remove，replace，isEmpty，clear和get函数，用于对链表的各种操作。

**重要算法核心代码**

1、构造函数和析构函数

构造函数会根据传入的数据类型申请一片对应的内存空间，同时将头结点指针指向该内存。

析构函数会先执行一遍链表的清空节点操作，再把头结点删除，完成对链表的完全清理。



2、给定位置的后插操作和尾插操作

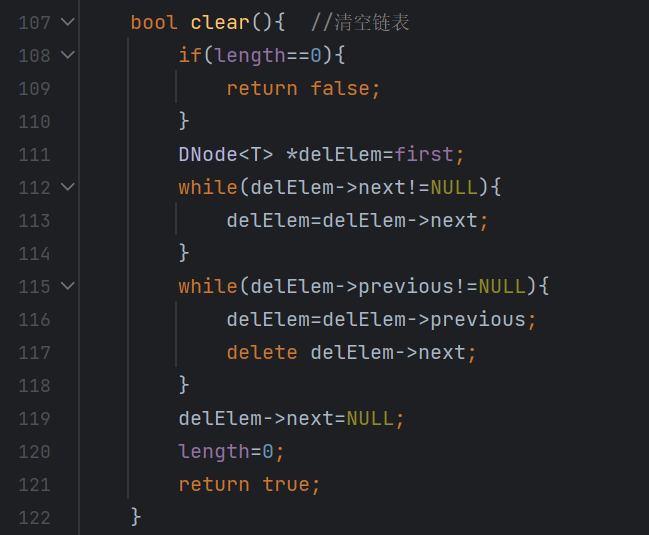
后插操作首先会根据位置定位到要后插的元素，然后更改前后指针以及插入元素的指针，此时会进行判尾操作，如果是在尾部插入，那么更改后继结点的指针操作不会执行

尾插操作类似于后插操作，会直接定位到链表尾再执行插入操作



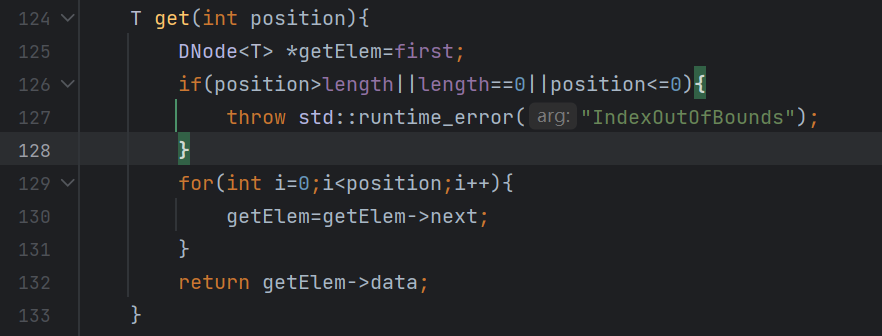
3、给定位置删除、修改信息

和插入操作的第一步类似，先要定位删除/修改的元素，然后对节点进行删除/修改操作并改变前后指针。不同的是删除前要对链表判空，如果是空链表则返回false。



4、清空链表

先对链表判空，如果已空则不需要清空。若不空，则定位到链表的末尾，从末尾开始逐个删除只保留头结点

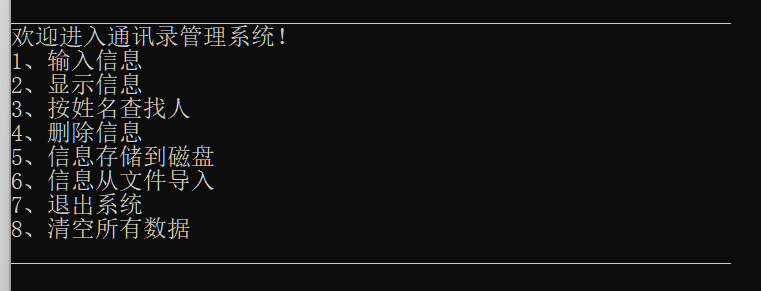


5、给定位置获取信息

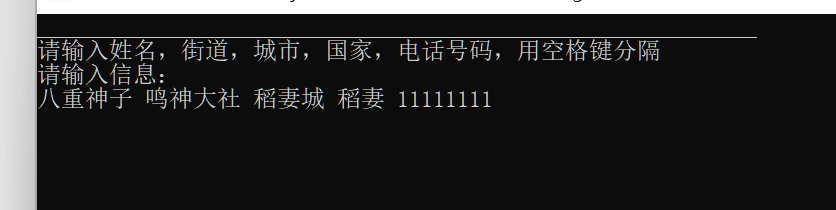
给定不合法的位置会抛出异常，若给定位置合法，则返回该位置对应的数据

**测试运行界面**

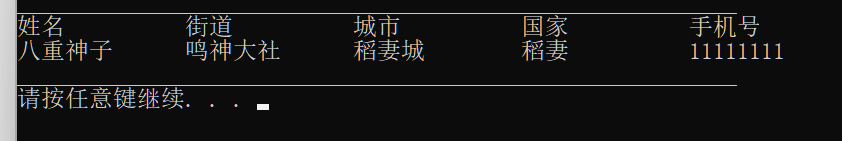
1. **主界面**

****

1. **输入信息**

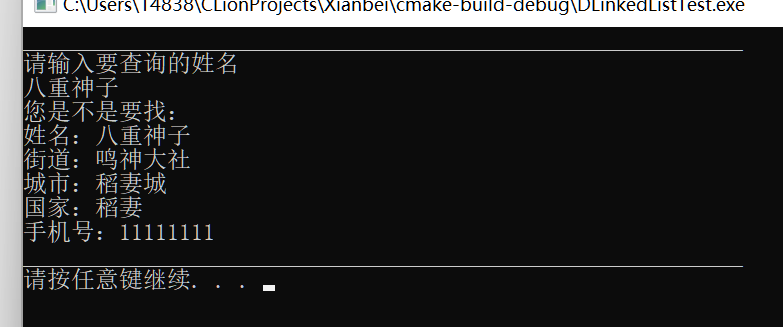
****

**显示：**

****

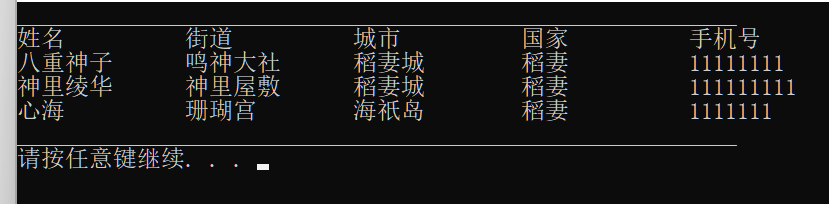
1. **按姓名查找人**

****

****

1. **删除信息**

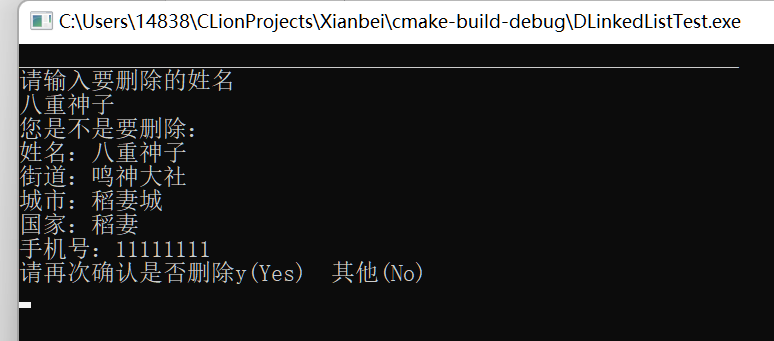
**现有的信息：**

****

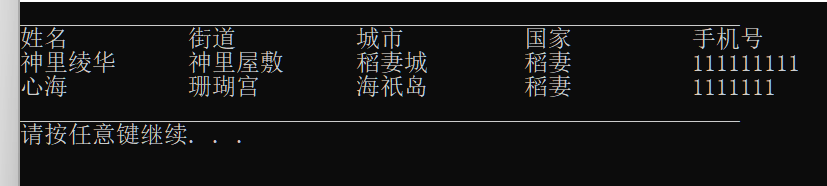
**输入需要删除的人名：**

****

**确认信息：**

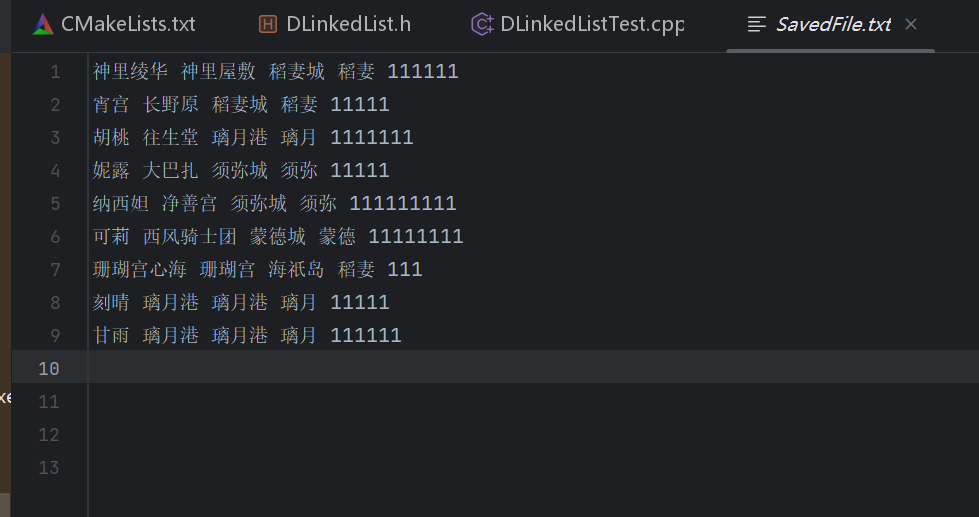
****

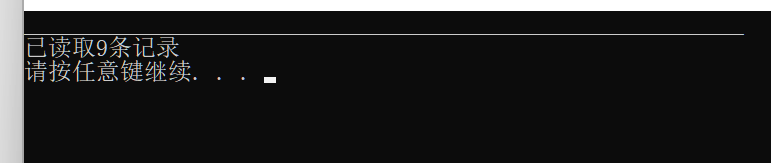
**输入y后即可删除**

****

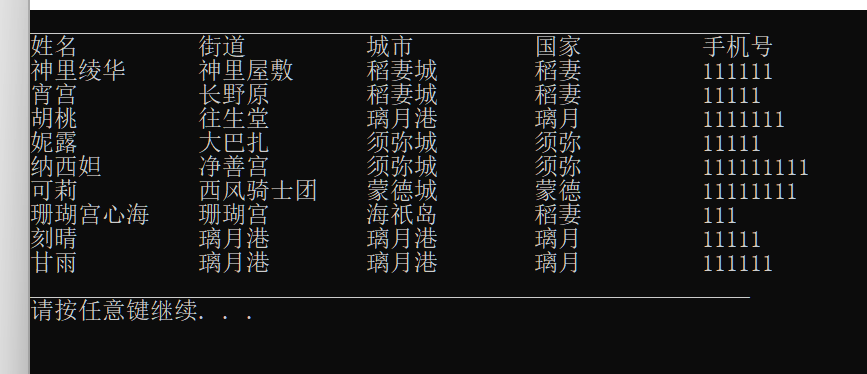
1. **信息从磁盘中读取**

**默认从程序存放目录下读取SavedFile.txt**

****

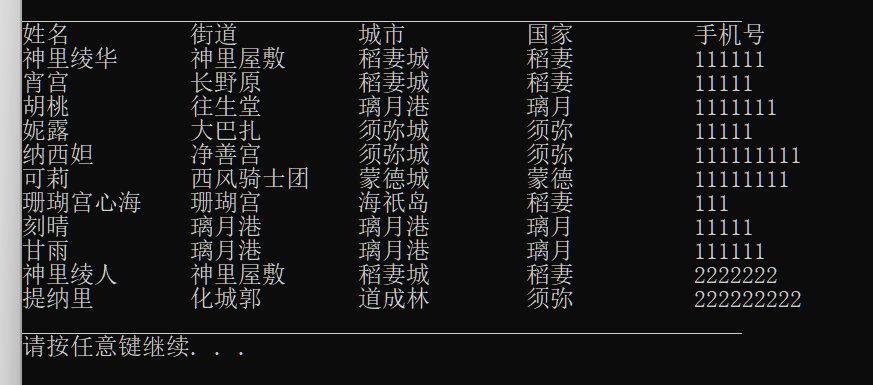
****

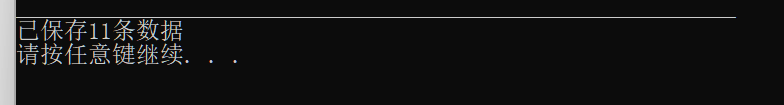
**可以在显示菜单里看到已读取的内容**

****

1. **信息写入磁盘**

**和上次相比，添加了两条信息**

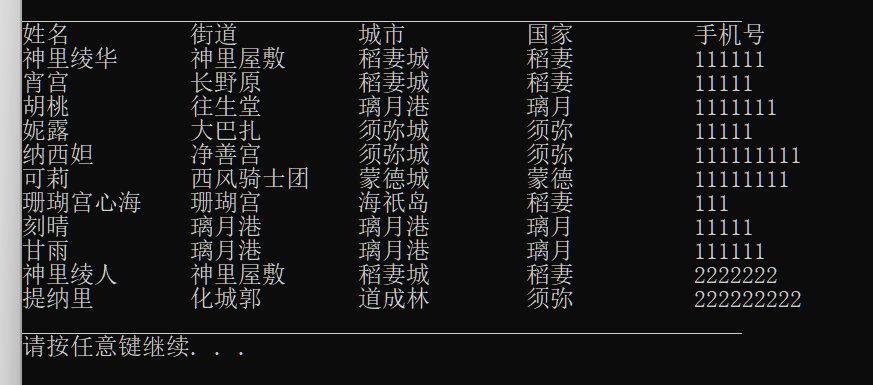
****

****

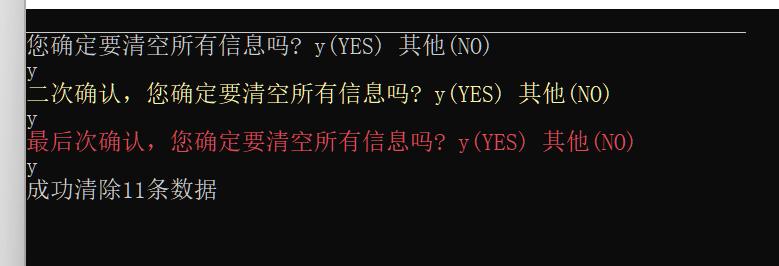
**文件已被修改，默认也是写入SavedFile.txt**

****

1. **清空所有数据**

****

**清空时需要三次确认**

****

****

**完整源程序：**

**头文件部分**

**#pragma once**

**#ifndef XIANBEI\_DLINKEDLIST\_H**

**#define XIANBEI\_DLINKEDLIST\_H**

**#include "iostream"**

**#include "exception"**

**template<typename T>**

**struct DNode{**

**T data;**

**DNode<T> \*previous=NULL;**

**DNode<T> \*next=NULL;**

**};**

**template<typename T>**

**class DLinkedList {**

**public:**

**int length;**

**DLinkedList(){ //构造函数**

**length=0;**

**first=new DNode<T>;**

**}**

**~DLinkedList(){ //析构函数**

**clear();**

**if(first!=NULL){**

**delete first;**

**first=NULL;**

**}**

**}**

**void print(){ //打印链表数据**

**DNode<T> \*thisElem;**

**thisElem=first->next;**

**while(thisElem!=NULL){**

**std::cout<<thisElem->data<<std::endl;**

**thisElem=thisElem->next;**

**}**

**}**

**bool add(int position,const T &insertData){ //后插操作**

**if(position>length||position<=0){**

**return false;**

**}**

**DNode<T> \*thisElem=first;**

**for(int i=0;i<position;i++){**

**thisElem=thisElem->next;**

**}**

**DNode<T> \*insertElem=new DNode<T>;**

**insertElem->data=insertData;**

**insertElem->previous=thisElem;**

**insertElem->next=thisElem->next;**

**thisElem->next=insertElem;**

**if(position==length){**

**thisElem->next->previous=insertElem;**

**}**

**length++;**

**return true;**

**}**

**bool add(const T &insertData){ //尾插操作**

**DNode<T> \*thisElem=first;**

**for(int i=0;i<length;i++){**

**thisElem=thisElem->next;**

**}**

**DNode<T> \*insertElem=new DNode<T>;**

**insertElem->data=insertData;**

**thisElem->next=insertElem;**

**insertElem->previous=thisElem;**

**length++;**

**return true;**

**}**

**bool remove(int position){ //给定位置删除**

**if(position>length||length==0||position<=0){**

**return false;**

**}**

**DNode<T> \*delElem=first;**

**for(int i=0;i<position;i++){**

**delElem=delElem->next;**

**}**

**if(position==length){**

**DNode<T> \*previousElem=delElem->previous;**

**delete delElem;**

**previousElem->next=NULL;**

**}else{**

**DNode<T> \*nextElem=delElem->next;**

**DNode<T> \*previousElem=delElem->previous;**

**delete delElem;**

**previousElem->next=nextElem;**

**nextElem->previous=previousElem;**

**}**

**length--;**

**return true;**

**}**

**bool replace(int position,const T &insertData){ //给定位置修改值**

**if(position>length||length==0||position<=0){**

**return false;**

**}**

**DNode<T> \*replaceElem=first;**

**for(int i=0;i<position;i++){**

**replaceElem=replaceElem->next;**

**}**

**replaceElem->data=insertData;**

**return true;**

**}**

**bool isEmpty(){ //判空**

**if(length==0){**

**return true;**

**}else{**

**return false;**

**}**

**}**

**bool clear(){ //清空链表**

**if(length==0){**

**return false;**

**}**

**DNode<T> \*delElem=first;**

**while(delElem->next!=NULL){**

**delElem=delElem->next;**

**}**

**while(delElem->previous!=NULL){**

**delElem=delElem->previous;**

**delete delElem->next;**

**}**

**delElem->next=NULL;**

**length=0;**

**return true;**

**}**

**T get(int position){**

**DNode<T> \*getElem=first;**

**if(position>length||length==0||position<=0){**

**throw std::runtime\_error("IndexOutOfBounds");**

**}**

**for(int i=0;i<position;i++){**

**getElem=getElem->next;**

**}**

**return getElem->data;**

**}**

**private:**

**DNode<T> \*first;**

**};**

**测试类部分（通讯录管理系统）**

**#include "iostream"**

**#include "string"**

**#include "DLinkedList.h"**

**#include "windows.h"**

**#include "fstream"**

**using namespace std;**

**class Person{**

**public:**

**string name;**

**string road;**

**string city;**

**string nation;**

**string phone;**

**Person(){};**

**Person(string name,string road,string city,string nation,string phone){**

**this->name=name;**

**this->road=road;**

**this->city=city;**

**this->nation=nation;**

**this->phone=phone;**

**}**

**};**

**DLinkedList<Person> d;**

**void Menu(){**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**cout<<"欢迎进入通讯录管理系统！"<<endl;**

**cout<<"1、输入信息"<<endl;**

**cout<<"2、显示信息"<<endl;**

**cout<<"3、按姓名查找人"<<endl;**

**cout<<"4、删除信息"<<endl;**

**cout<<"5、信息存储到磁盘"<<endl;**

**cout<<"6、信息从文件导入"<<endl;**

**cout<<"7、退出系统"<<endl;**

**cout<<"8、清空所有数据"<<endl;**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**}**

**void Input(){**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**cout<<"请输入姓名，街道，城市，国家，电话号码，用空格键分隔"<<endl;**

**cout<<"请输入信息："<<endl;**

**string name,road,city,nation,phone;**

**cin>>name>>road>>city>>nation>>phone;**

**Person p=\*new Person(name,road,city,nation,phone);**

**bool isRepeat=false;**

**for(int i=1;i<=d.length;i++){**

**Person p1=d.get(i);**

**if(p1.name==p.name){**

**isRepeat=true;**

**break;**

**}**

**}**

**if(isRepeat){**

**cout<<"同名记录已存在"<<endl;**

**}else{**

**d.add(p);**

**cout<<"添加成功！"<<endl;**

**}**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**Sleep(1000);**

**system("cls");**

**Sleep(300);**

**}**

**void PrintAll(){**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**cout<<"姓名 街道 城市 国家 手机号 "<<endl;**

**for(int i=1;i<=d.length;i++){**

**Person p=d.get(i);**

**printf("%-14s%-14s%-14s%-14s%-14s\n",p.name.c\_str(),p.road.c\_str(),p.city.c\_str(),p.nation.c\_str(),p.phone.c\_str());**

**}**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**system("pause");**

**}**

**void Search(){**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**cout<<"请输入要查询的姓名"<<endl;**

**string name;**

**cin>>name;**

**bool find=false;**

**Person p;**

**for(int i=1;i<=d.length;i++){**

**p=d.get(i);**

**if(name==p.name){**

**find=true;**

**break;**

**}**

**}**

**if(find){**

**cout<<"您是不是要找："<<endl;**

**cout<<"姓名："<<p.name<<endl;**

**cout<<"街道："<<p.road<<endl;**

**cout<<"城市："<<p.city<<endl;**

**cout<<"国家："<<p.nation<<endl;**

**cout<<"手机号："<<p.phone<<endl;**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**system("pause");**

**}else{**

**cout<<"您要找的人不存在"<<endl;**

**Sleep(3000);**

**}**

**}**

**void Delete(){**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**cout<<"请输入要删除的姓名"<<endl;**

**string name;**

**cin>>name;**

**bool find=false;**

**Person p;**

**int position=0;**

**for(int i=1;i<=d.length;i++){**

**p=d.get(i);**

**if(name==p.name){**

**find=true;**

**position=i;**

**break;**

**}**

**}**

**if(find){**

**cout<<"您是不是要删除："<<endl;**

**cout<<"姓名："<<p.name<<endl;**

**cout<<"街道："<<p.road<<endl;**

**cout<<"城市："<<p.city<<endl;**

**cout<<"国家："<<p.nation<<endl;**

**cout<<"手机号："<<p.phone<<endl;**

**cout<<"请再次确认是否删除y(Yes) 其他(No)"<<endl;**

**char option;**

**cin>>option;**

**if(option=='y'){**

**d.remove(position);**

**cout<<"删除成功！"<<endl;**

**}else{**

**cout<<"您取消了操作"<<endl;**

**}**

**}else{**

**cout<<"您要删除的人不存在"<<endl;**

**}**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**system("pause");**

**}**

**void Save(){**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**ofstream wFile("SavedFile.txt",ios::out);**

**if(!wFile){**

**cerr<<"文件写入失败，请检查程序写入权限或磁盘坏道!"<<endl;**

**}else{**

**for(int i=1;i<=d.length;i++){**

**Person p=d.get(i);**

**wFile<<p.name<<" "<<p.road<<" "<<p.city<<" "<<p.nation<<" "<<p.phone<<endl;**

**}**

**cout<<"已保存"<<d.length<<"条数据"<<endl;**

**}**

**system("pause");**

**}**

**void InputFromFile(){**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**ifstream rFile("SavedFile.txt",ios::in);**

**if(!rFile){**

**cerr<<"文件读取失败，请检查程序读取权限或磁盘坏道!"<<endl;**

**}else{**

**int i=0;**

**for(;;i++){**

**string name;**

**string road;**

**string city;**

**string nation;**

**string phone;**

**rFile>>name>>road>>city>>nation>>phone;**

**if(name==""){**

**break;**

**}**

**Person p(name,road,city,nation,phone);**

**d.add(p);**

**}**

**cout<<"已读取"<<i<<"条记录"<<endl;**

**system("pause");**

**}**

**}**

**void Clear(){**

**cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;**

**string confirm1;**

**cout<<"您确定要清空所有信息吗? y(YES) 其他(NO)"<<endl;**

**cin>>confirm1;**

**if(confirm1!="y"){**

**cout<<"用户取消操作"<<endl;**

**return;**

**}**

**string confirm2;**

**cout<<"\033[33;1m二次确认，您确定要清空所有信息吗? y(YES) 其他(NO)\033[0m"<<endl;**

**cin>>confirm2;**

**if(confirm1!="y"){**

**cout<<"用户取消操作"<<endl;**

**return;**

**}**

**string confirm3;**

**cout<<"\033[31;1m最后次确认，您确定要清空所有信息吗? y(YES) 其他(NO)\033[0m"<<endl;**

**cin>>confirm3;**

**if(confirm1!="y"){**

**cout<<"用户取消操作"<<endl;**

**return;**

**}**

**int size=d.length;**

**d.clear();**

**cout<<"成功清除"<<size<<"条数据"<<endl;**

**Sleep(3000);**

**}**

**int main(){**

**while(true){**

**system("cls");**

**Menu();**

**int choose;**

**cin>>choose;**

**switch (choose) {**

**case 1:**

**system("cls");**

**Input();**

**break;**

**case 2:**

**system("cls");**

**PrintAll();**

**break;**

**case 3:**

**system("cls");**

**Search();**

**break;**

**case 4:**

**system("cls");**

**Delete();**

**break;**

**case 5:**

**system("cls");**

**Save();**

**break;**

**case 6:**

**system("cls");**

**InputFromFile();**

**break;**

**case 7:**

**system("cls");**

**cout<<"感谢您的使用"<<endl;**

**system("pause");**

**exit(0);**

**case 8:**

**system("cls");**

**Clear();**

**break;**

**default:**

**system("cls");**

**cerr<<"您输入的指令有误，请重新输入"<<endl;**

**system("pause");**

**break;**

**}**

**}**

**}**