项目说明文档

数据结构课程设计

——家谱管理系统

作 者 姓 名： 陈君

学 号： 2250420

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji Universit

# 1 分析

## 1.1 背景分析

家谱是一种以表谱形式，记载一个以血缘关系为主体的家族世袭繁衍和重要任务事迹的特殊图书体裁。家谱是中国特有的文化遗产，是中华民族的三大文献（国史，地志，族谱）之一，属于珍贵的人文资料，对于历史学，民俗学，人口学，社会学和经济学的深入研究，均有其不可替代的独特功能。本项目兑对家谱管理进行简单的模拟，以实现查看祖先和子孙个人信息，插入家族成员，删除家族成员的功能。

## 1.2 功能分析

本项目的实质是完成对家谱成员信息的建立，查找，插入，修改，删除等功能，可以首先定义家族成员数据结构，然后将每个功能作为一个成员函数来完成对数据的操作，最后完成主函数以验证各个函数功能并得到运行结果。

# 2 设计

## 2.1 数据结构设计

如上功能分析所述，这个家谱是有祖先子女一代一代像树的分支一样发散开的，所以可以采用树这个数据结构来存储家谱，同时存储方式为了方便插入删除以及节省空间，使用链表来存储每个结点。

## 2.2 类结构设计

本系统采用设计家谱Family类来存储家谱，用nodelist类来存储结点的信息，同时将Family类定义为nodelist的友元，方便反问结点信息。

## 2.3 成员与操作设计

**家谱类（listnode）**

class Family;

class listnode

{

public:

char\* get\_name() { return name; }

private:

friend Family;

listnode\* children[100] = {NULL};

char name[100];

int childrennum = 0;

};

class Family

{

public:

Family();//默认构造函数

~Family();//析构函数

listnode\* find(char str[100], listnode\* p);//查找函数，寻找这个人所在位置

void improve\_family();//完善家谱

void add\_member();//添加家庭成员

void release\_member();//解散局部家庭

void reset\_member();//更改家庭成员姓名

listnode\* get\_root() { return root; }//获取根结点的地址

void show\_family\_tree(listnode\* current);

void show\_family\_tree\_layer(listnode\* root);

private:

listnode\* root;

## };2.4 系统设计

主程序先调用initialzeio函数初始化io界面然后再声明一个Family对象创建家谱，然后使用一个switch语句，跳转到完善家谱，增加成员，解散家庭，修改成员，退出五个操作。

# 3 实现

## 3.1Family::Family()//默认构造函数

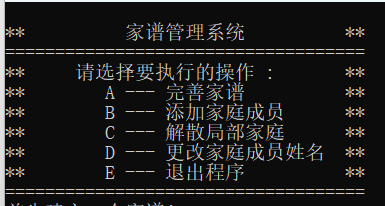
提示用户输入祖先的姓名，返回“此家谱的祖先是：”。New操作建立一个结点，然后将祖先名字赋值到结点的姓名，并且将孩子地址赋值为NULL

### 3.2.3 构造函数功能截屏示例

## 3.2 void initializeio()//初始化io界面

Cout输出题目中展示的画面

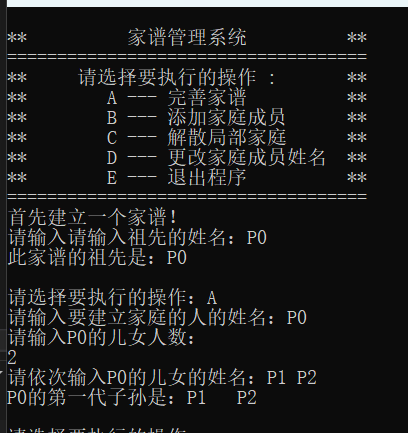
### 3.2.1 void initializeio()功能截屏示例



## 3.3 void Family::improve\_family()//完善家谱功能

具体实现：提示用户输入姓名，儿女人数，儿女姓名，定义一个临时的vector<string>的容器，for循环依次new一个新的结点，存储每个子节点的地址到vetor，并且将子女的姓名存入新节点中。调用find函数找到要建立家庭的人current，将孩子容器vector赋值给current->children。接着展示这些子女。

### 3.3.2 putout功能截图示例



## 3.4 Family::~Family()

## {if (root != NULL)delete root;}//如果根节点不为空，就删除根节点

## 3.5 listnode::listnode()//结点类的构造函数

将孩子成员赋值为NULL，name赋值为“”。

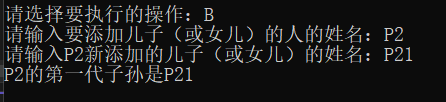
## 3.6 listnode\* Family::find(string str,listnode\*p)//查找函数寻找这个人所在位置

具体实现：如果p结点的name=str，就直接返回p结点的地址，如果p结点的孩子结点为NULL，就返回NULL。如果都不是就使用一个for循环调用find函数寻找p的每个孩子结点中有没有这个名字，如果这个结点的所有子树都没有找到就返回NULL；

## 3.7 void Family::add\_member()//添加成员

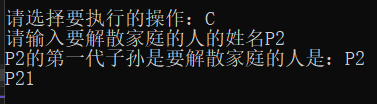
提示用户输入相关信息，然后用find函数找到添加成员的父节点，然后new操作划分一个新的结点，将这个结点pushback进父节点的children成员中。然后for循环输出每个孩子的姓名。

截图展示



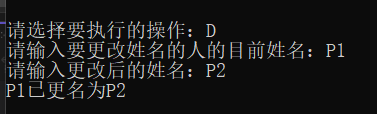
## 3.8 void Family::release\_member()//解散局部家庭

具体实现：提示用户输入相关信息，用find找到要解散的家庭结点，然后输出这个结点的信息，最后for循环输出信息，然后delete结点p。



## 3.9 void Family::reset\_member()//更改家庭成员姓名

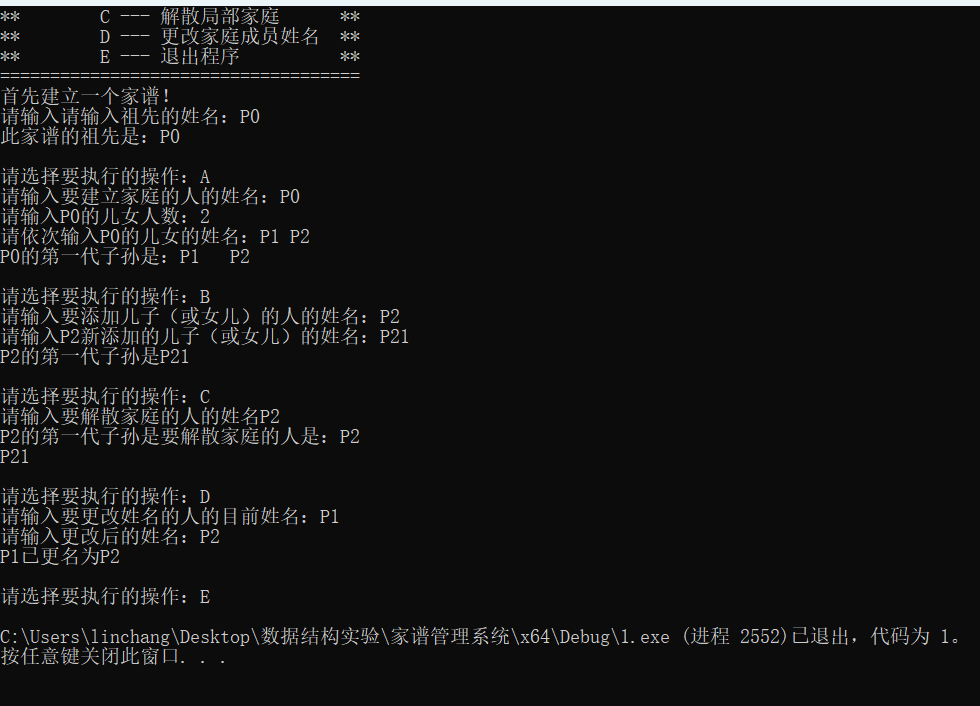
具体实现，提示用户输入相关信息，然后用find找到要更改的结点，将新名字赋值给p->name。



## 3.6 总体系统的实现

主程序先调用initialzeio函数初始化io界面然后再声明一个Family对象创建家谱，然后使用一个switch语句，跳转到完善家谱，增加成员，解散家庭，修改成员，退出五个操作。

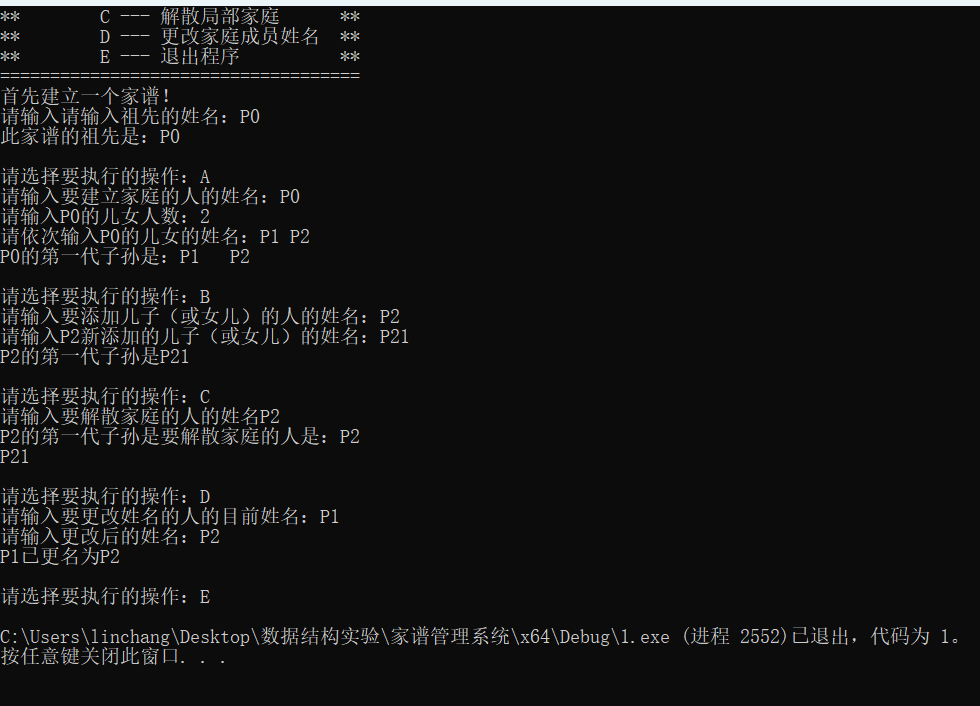
### }3.6.1 总体系统截屏示例



# 4 测试

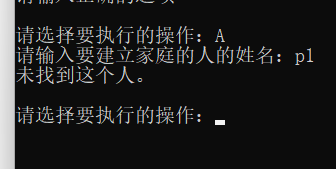
## 4.1 系统总体功能测试功能测试

### 4.1.1 截屏展示



4.2 输入错误建立家庭的人的姓名

4.2.1截屏展示



4.3 输入错误操作数

4.3.1 截屏展示

