数据结构项目六文档

同济大学 软件学院 15级2班 1552651 王依睿

- 使用说明
  - 操作手册
    - 开始
    - 建立家谱
    - 完善家谱
    - 添加家庭成员
    - 解散局部家庭成员
    - 更改家庭成员姓名
    - 退出程序
    - 整体预览
  - 注意事项
- 概述
  - 项目功能要求
  - 程序设计目的
  - 算法思路
  - 数据结构
  - 文件目录
- 函数接口
  - 多叉树节点（TreeNode）接口
    - 多叉树节点类（TreeNode）代码实现
  - 多叉树（Tree）接口
    - 多叉树类（Tree）代码实现

使用说明

操作手册

开始

运行程序后，显示家谱管理系统的操作说明。

建立家谱

请求用户输入家谱祖先的姓名，建立一个家谱。

输出祖先姓名。

完善家谱

- 选择操作A。
- 输入要建立家庭的人的姓名。
- 输入要建立家庭的人的儿女人数。
- 依次输入儿女的姓名。

输出建立家庭的人及儿女的姓名。

添加家庭成员

- 选择操作B。
- 输入要添加子女的人的姓名。
- 输入要添加的子女的姓名。

```
请选择要执行的操作：B
请输入要添加儿子(或女儿)的人的姓名：P2
请输入P2新添加儿子(或女儿)的姓名：P21
```

输出要添加子女的人的全部子女的姓名。

```
P2的第一代子孙是：P21
```

解散局部家庭成员

- 选择操作C。
- 输入要解散家庭的人的姓名。

```
请选择要执行的操作：C
请输入要解散家庭的人的姓名：P2
```

输出要解散家庭的人姓名。

```
请选择要执行的操作：C
请输入要解散家庭的人的姓名：P2
```

更改家庭成员姓名

- 选择操作D。
- 输入要更改姓名的人的目前姓名。
- 输入更改后的姓名。

```
请选择要执行的操作：D
请输入要更改姓名的人的目前姓名：P13
请输入更改后的姓名：P14
```

输出更改前姓名和更改后姓名。

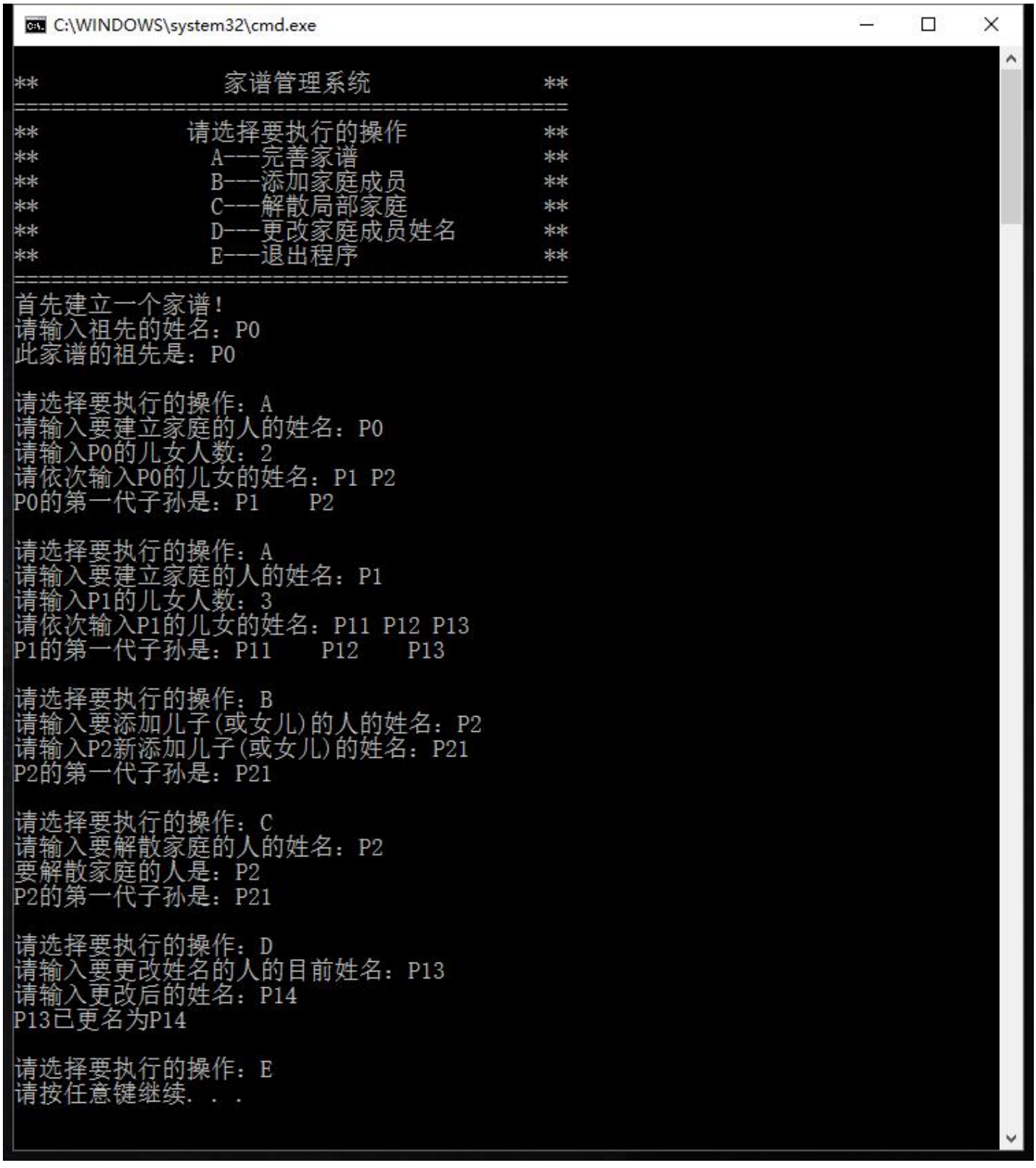
```
P13已更名为P14
```

退出程序

选择操作E。

```
请选择要执行的操作：E
请按任意键继续. . .
```

整体预览



注意事项

- 若尝试对一个不在家谱中建立过家庭的人进行检索操作，程序会输出“家谱中不存在此人！”字样。
- 若尝试对一个已在家谱中建立过家庭的人进行建立家庭的操作，程序会输出“此人已有家庭，不可重复建立！”字样。

概述

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。

项目功能要求

本项目的实质是完成兑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。

程序设计目的

实现一个利用多叉树存储信息的简单程序，练习多叉树的编写实现，熟悉面向对象程序设计。

算法思路

以遍历多叉树为基本算法，而后通过改变节点的指针实现删除与插入。

数据结构

- 将多叉树节点封装成一个类。
- 将多叉树封装成一个类。

文件目录

- 6_1552651_wangyirui.cpp（主文件）
- tree_class.h（多叉树类）
- 6_1552651_wangyirui.exe（可执行文件）
- 6_1552651_wangyirui.pdf（项目文档）

函数接口

多叉树节点（TreeNode）接口

多叉树节点类（TreeNode）代码实现

```
struct  TreeNode{
    string name_;
    PtrToNode firstchild_ = NULL;                                   //第一个孩子
    PtrToNode presibling_ = NULL;                                   //前一个兄弟姐妹
    PtrToNode nextsibling_ = NULL;                                  //后一个兄弟姐妹
};
```

多叉树（Tree）接口

成员函数名	公有/私有性	功能	参数	返回值类型
~Tree()	public	析构函数，通过调用函数RecursionForRemove，释放之前用new方法为节点分配的空间	/	/
void Init(string name)	public	初始化，加入根节点（祖先）	祖先名字string	/
void BuildFamily(string dadname, int sonnum, string *sonname)	public	完善家庭	父亲的姓名string，子女的数量int，子女姓名的数组指针string*	/
void AddChild(string dadname, string sonname)	public	添加家庭成员	父亲的姓名string，子女的姓名string	/
void Remove(string name)	public	调用RecursionForRemove，解散局部家庭	姓名string	/
void RecursionForRemove(PtrToNode)	public	用于函数remove的递归函数		
void Modify(string oldname, string newname)	public	更改家庭成员姓名	更改前姓名string，更改后姓名string	/
void PrintAncestor()	public	打印祖先	/	/
void PrintSon(string name)	public	打印儿子	父亲姓名string	/
void PrintNewName(string oldname, string name)	public	打印更改前后的姓名	更改前姓名string，更改后姓名string	/

多叉树类（Tree）代码实现

```
class Tree {
public:
    ~Tree();
    PtrToNode Find(string name, PtrToNode t);                          //根据姓名称寻找节
点
    void Init(string name);                                            //初始化，加入根节
点
    void BuildFamily(string dadname, int sonnum, string *sonname);//完善家庭
    void AddChild(string dadname, string sonname);                     //添加家庭成员
    void Remove(string name);                                          //解散局部家庭
    void RecursionForRemove(PtrToNode);                                //用于remove的递归
函数
    void Modify(string oldname, string newname);                       //更改家庭成员姓名
    void PrintAncestor();                                              //打印祖先
    void PrintSon(string name);                                        //打印儿子
    void PrintNewName(string oldname, string name);                    //打印更改前后的姓
名
private:
    PtrToNode root_ = NULL;                                            //祖先
};
```