

数据结构课程设计
项目说明文档

家谱管理系统

软件工程

张靖凯

2151396



目录

项目简介	3
项目功能要求	3
项目示例	3
项目设计	4
数据结构设计	4
Family 类	4
SilbingTree 类	4
数据结构关键代码	5
解题思路	6
设计亮点	7
代码注释规范	7
输入错误处理	7
项目测试	8
Linux 测试运行 (Ubuntu)	8

项目简介

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。

项目示例

```
***          家谱管理系统          ***
=====
**      请选择要执行的操作：      **
**      A --- 完善家谱            **
**      B --- 添加家庭成员        **
**      C --- 解散局部家庭        **
**      D --- 更改家庭成员姓名    **
**      E --- 退出程序            **
=====
首先建立一个家谱！
请输入祖先的姓名：P0
此家谱的祖先是：P0

请选择要执行的操作：A
请输入要建立家庭的人的姓名：P0
请输入P0的儿女人数：2
请依次输入P0的儿女的姓名：P1 P2
P0的第一代子孙是：P1 P2

请选择要执行的操作：A
请输入要建立家庭的人的姓名：P1
请输入P1的儿女人数：3
请依次输入P1的儿女的姓名：P11 P12 P13
P1的第一代子孙是：P11 P12 P13

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新添加的儿子（或女儿）的姓名：P21
P2的第一代子孙是：P21

请选择要执行的操作：C
请输入要解散家庭的人的姓名：P2
要解散家庭的人是：P2
P2的第一代子孙是：P21

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
P13已更名为P14

请选择要执行的操作：E
Press any key to continue_
```

项目设计

数据结构设计

采用左子女右兄弟二叉树实现，其中兄弟指针为双向链表，方便函数递归操作。Family 类，对应有父指针和子指针以及前后兄弟指针，将整个家庭构造成树的结构。在添加新家庭成员时，只需在原链表尾处添加新节点并标记其父指针为原表尾。同样，解散家庭成员时，根据该成员所处位置，将其前后兄弟指针相连，并减少父节点的子孙数后删除该节点。完善家庭成员时，先输入子女个数，之后为每个子女建立新节点并与父节点相连，每个子女节点之间用兄弟指针相连，便于查找家庭成员。

Family 类

Public Attributes	
string	name 成员名字 More...
int	childCnt 该成员所有孩子数 More...
FamilyPtr	father 父指针 More...
FamilyPtr	broNext 兄弟后指针 More...
FamilyPtr	broPrev 兄弟前指针 More...
FamilyPtr	child 孩子指针 More...

SilbingTree 类

Public Types	
using	FamilyPtr = Family *
Public Member Functions	
	SiblingTree ()
	~SiblingTree ()
FamilyPtr	findMember (string name, FamilyPtr key)
void	destroy (FamilyPtr familyNode)
void	setAncestor ()
void	showChild (FamilyPtr familyNode) const
operations	
void	perfectFamily ()
void	addMember ()
void	divMember ()
void	changeMember ()
Private Attributes	
FamilyPtr	ancestor 祖先 More...

数据结构关键代码

解散家庭递归算法如下：

```
void SiblingTree::destroy(FamilyPtr familyNode)
{
    FamilyPtr temp = familyNode;

    while (temp->broNext)
        temp = temp->broNext;
    while (temp->broPrev)
    {
        temp = temp->broPrev;
        temp->broNext->broPrev = nullptr;
        temp->broNext->broNext = nullptr;
        destroy(temp->broNext);    // 孤立该节点后删除此节点(含
child)
    }
    if (temp->child)
        destroy(temp->child);
    delete familyNode;
}
```

寻找成员递归算法如下：

```
Family* SiblingTree::findMember(string name, FamilyPtr key)
{
    FamilyPtr ans = nullptr, p = key;
    if (p->name == name)
        return p;
    // 兄弟指针中寻找
    while (p->broNext)
    {
        ans = findMember(name, p->broNext);
        if (ans)
            return ans;
        p = p->broNext;
    }
    // 孩子中寻找
    p = key;
    if (p->child)
    {
        ans = findMember(name, p->child);
        if (ans)
            return ans;
    }
    return ans;
}
```

解题思路

操作 A:

完善家谱：在得到家庭成员的姓名后，用查找函数在家庭树中查找该成员。在对其子指针进行判断，若不空则已建立家庭不进行操作。若为空则需要输入子女的个数和姓名，之后用循环添加新子女到家庭树中。

操作 B:

添加家庭成员：同操作 A，得到家庭成员姓名后先在家庭树中找到该人，之后输入该成员的姓名，在家庭树中建立新的节点，方法与操作 A 相同。

操作 C:

解散局部家庭：首先找到要删除的家庭成员的人，之后进行分情况讨论，要判断该人是否有兄弟指针，如果有则需将前后兄弟指针相连或置空，并让其父节点的子孙数减一，之后删除该节点。

操作 D:

更改家庭成员姓名，找到该家庭成员后更改其 name 即可。

查找函数:

收先判断传入函数的 key 是否就是所找家庭成员，若不是，在对其兄弟节点和孩子节点进行递归查找。

展示家庭成员:

得到 parent 节点后，将其子节点赋给 temp，之后展示 temp 的所有兄弟指针所处节点。

设计亮点

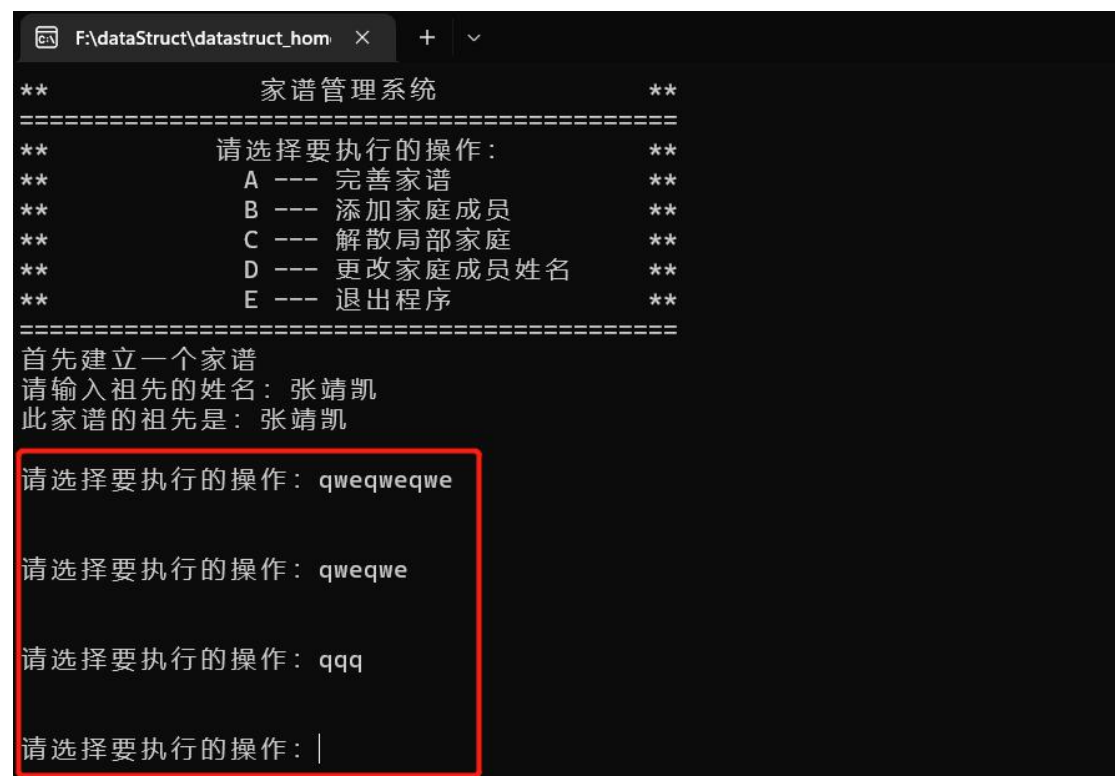
代码注释规范

采用了 doxygen 注释规范，对类、函数等有简要说明，命名采用驼峰命名法，类内的各个声明规范，方便本人回顾之前写过的代码和 code reviewer 查看，使 API 规范，帮助这个开发流程高效、规范地进行。

```
/// @name operations
/// @{
void perfectFamily();
void addMember();
void divMember();
void changeMember();
/// @}
// end of operations

private:
    FamilyPtr ancestor; ///< 祖先
};
```

输入错误处理



```
F:\dataStruct\datastruct_hom X + v

**                      家谱管理系统                      **
=====
**      请选择要执行的操作：      **
**      A --- 完善家谱          **
**      B --- 添加家庭成员      **
**      C --- 解散局部家庭      **
**      D --- 更改家庭成员姓名  **
**      E --- 退出程序          **
=====
首先建立一个家谱
请输入祖先的姓名：张靖凯
此家谱的祖先是：张靖凯

请选择要执行的操作：qweqweqwe

请选择要执行的操作：qweqwe

请选择要执行的操作：qqq

请选择要执行的操作：|
```

项目测试

```
F:\dataStruct\datastruct_hom  X  +  v

**                      家谱管理系统                      **
=====
**      请选择要执行的操作：      **
**      A --- 完善家谱            **
**      B --- 添加家庭成员        **
**      C --- 解散局部家庭        **
**      D --- 更改家庭成员姓名    **
**      E --- 退出程序            **
**      =====      **

首先建立一个家谱
请输入祖先的姓名：张靖凯
此家谱的祖先是：张靖凯

请选择要执行的操作：A
请输入要建立家庭的人的姓名：张靖凯
请输入张靖凯的儿女个数：2
请依次输入张靖凯的儿女的姓名：小凯 小凯凯
张靖凯的第一代子孙是：小凯 小凯凯

请选择要执行的操作：B
请输入要添加儿子（女儿）的人的姓名：小凯
请输入小凯新添加的儿子（或女儿）的姓名：Benz
小凯的第一代子孙是：Benz

请选择要执行的操作：C
请输入要解散的家庭的人的姓名：小凯
要解散家庭的人是：小凯
小凯的第一代子孙是：Benz
```

Linux 测试运行（Ubuntu）

```
kk@LAPTOP-UJDPHKT8: ~/da  X  +  v

1 /* @author 张靖凯 */
2 #ifndef _SIBLING_
3 #define _SIBLING_
4
5 #include <string>
6 #include <iostream>
7 using namespace std;
8
9 /// @brief 家庭类
10 struct Family
11 {
12     using FamilyPtr = Family*;
13
14     string name; >--->---< 成员名字
15     int childCnt; >--->---< 该成员所有孩子数
16     FamilyPtr father; >---< 父指针
17     FamilyPtr broNext; >---< 兄弟后指针
18     FamilyPtr broPrev; >---< 兄弟前指针
19     FamilyPtr child; >---< 孩子指针
20
21     Family() : name(""), childCnt(0), father(nullptr), broNext(nullptr), broPrev(nullptr), child(nullptr) {}
22 };
23
24 /**
25  * @class SiblingList
26  * @brief 兄弟姐妹树
27  */
SiblingTree.hpp 1,1
"SiblingTree.hpp" 265L, 6014C
```



```
kk@LAPTOP-UJDPHKT8: ~/da  X + v
kk@LAPTOP-UJDPHKT8:~$ ls
dataStruct
kk@LAPTOP-UJDPHKT8:~$ cd dataStruct/
kk@LAPTOP-UJDPHKT8:~/dataStruct$ ls
homework1 homework2 homework3 homework4 homework5 homework6 homework7 test
kk@LAPTOP-UJDPHKT8:~/dataStruct$ cd homework6
kk@LAPTOP-UJDPHKT8:~/dataStruct/homework6$ ls
SiblingTree.hpp h6.cpp run
kk@LAPTOP-UJDPHKT8:~/dataStruct/homework6$ ./run
**                家谱管理系统                **
=====
**                请选择要执行的操作：                **
**                A --- 完善家谱                **
**                B --- 添加家庭成员                **
**                C --- 解散局部家庭                **
**                D --- 更改家庭成员姓名                **
**                E --- 退出程序                **
=====
首先建立一个家谱
请输入祖先的姓名：张靖凯
此家谱的祖先是：张靖凯

请选择要执行的操作：A
请输入要建立家庭的人的姓名：张靖凯
请输入张靖凯的儿女个数：1
请依次输入张靖凯的儿女的姓名：小凯
张靖凯的第一代子孙是：小凯

请选择要执行的操作：|
```