



数据结构课程设计文档

题目：家谱管理系统

姓名： 赵卓冰

学号： 2252750

专业： 软件工程

年级： 2023 级

指导教师： 张颖

2024 年 11 月 19 日

运行环境与开发工具

项目要求

1 功能要求

项目简介

测试界面展示

1 Windows 界面

2 Linux 界面

代码架构

1 String 类

2 Vector 类

3 GeneTree 类

4 其他辅助类

功能模块介绍

1 祖先家谱树建立

2 家族成员的查找与信息更新

3 家谱树展示

用户交互

1 示例交互

性能考虑

错误处理和输入验证

未来改进

心得体会

1. 运行环境与开发工具

本项目支持在以下开发环境和编译运行环境中运行：

- **Windows 操作系统：**

- 版本：Windows 10 x64
- IDE：Visual Studio 2022 (Debug 模式)
- 编译器：MSVC 14.39.33519

- **Linux 操作系统：**

- 版本：Ubuntu 20.04.6 LTS
- IDE：VS Code
- 编译器：gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04.2)

2. 项目要求

本项目旨在实现一个家谱管理系统，通过家族成员的建立、查找、插入、修改和删除操作，支持对家谱的维护。用户可以通过系统对家族成员信息进行管理。具体实现上，家谱系统定义了家族成员的数据结构，并以成员函数的形式实现各项操作，主函数则负责控制台交互以验证各功能的实现。

2.1. 功能要求

- 输入说明：**用户可以通过控制台输入选择操作。根据操作指令，输入家族成员信息（如姓名、儿女数量及其姓名、要更改的姓名等）。
- 输出说明：**
 - 若执行操作成功，输出相关信息，例如建立的祖先名称、添加的儿女、家谱结构等。
 - 若成员查找失败或输入有误，则输出提示信息，告知用户操作失败。

3. 项目简介

本项目实现了一个家谱管理系统，支持家族成员的信息添加、查找、修改和删除等操作。通过树状结构存储家谱数据，以实现祖先与子孙间的多层关系表示，用户可以动态管理家谱成员。项目主要功能包括：

- 祖先家谱树建立：**通过输入祖先姓名，建立根节点代表家族的祖先。
- 成员信息操作：**包括为成员添加儿女、查找成员、更改成员姓名、解散局部家庭等功能。
- 家谱树的结构展示：**以缩进格式展示家谱树结构，清晰呈现代际关系。

4. 测试界面展示

4.1. Windows 界面

- 测试用例：**

○ 祖先家谱树建立

```
-----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 --- 打印家谱
F --- 退出程序

首先建立一个家谱！
请输入祖先的姓名：张三
此家族的祖先是：张三

请选择要执行的操作：A
请输入要建立家庭的成员姓名：张三
请输入张三的儿女人数：2
请依次输入张三的儿女的姓名：李四 王五
张三的第一代后代是：李四    王五

请选择要执行的操作：E

-----家谱的目录结构(缩进表示下一代)-----
张三
    李四
    王五
-----
```

○ 添加成员

```
请选择要执行的操作：B
请输入要添加儿/女的成员的姓名：王五
请输入王五新添加的儿/女的姓名：小冰
王五的第一代后代是：小冰

请选择要执行的操作：E

-----家谱的目录结构(缩进表示下一代)-----
张三
    李四
    王五
        小冰
-----
```

○ 解散局部家庭

请选择要执行的操作：C
请输入要解散家庭的人的姓名：
李四

请选择要执行的操作：E

-----家谱的目录结构(缩进表示下一代)-----
张三
 王五
 小冰

○ 更改家庭成员姓名

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：张三
请输入更改后的姓名：活神仙
张三已更名为活神仙

请选择要执行的操作：E

-----家谱的目录结构(缩进表示下一代)-----
活神仙
 王五
 小冰

4.2. Linux 界面

```
bing@bing-virtual-machine:~/ds$ ./genealogy
-----
                        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 --- 打印家谱
                F --- 退出程序
-----

首先建立一个家谱！
请输入祖先的姓名： jack
此家族的祖先是： jack

请选择要执行的操作： A
请输入要建立家庭的成员姓名： jack
请输入jack的儿女人数： 3
请依次输入jack的儿女的姓名： mike tom tim
jack的第一代后代是： mike      tom      tim

请选择要执行的操作： E

-----家谱的目录结构(缩进表示下一代) -----
jack
    mike
    tom
    tim
-----

请选择要执行的操作： B
请输入要添加儿/女的成员的姓名： tom
请输入tom新添加的儿/女的姓名： sarah
tom的第一代后代是： sarah
```

5. 代码架构

项目主要包含三个核心类：`String` 类、`Vector` 类以及 `GeneTree` 类，分别用于表示字符串、动态数组以及家谱树。

5.1. String 类

`String` 类用于存储和操作字符串，支持字符串的动态内存分配和常用操作。

- **成员函数：**
 - `Clear()`：清空字符串。
 - `Append()`：向字符串末尾添加字符。
 - `Erase()`：从指定位置开始删除字符。
 - `operator[]`：重载索引运算符，用于访问字符串中的字符。
 - `C_str()`：返回 C 风格字符串。

5.2. Vector 类

`Vector` 类是一个动态数组，用于存储成员列表，支持动态扩容和增删元素等操作。

- **主要方法：**
 - `PushBack()`：向数组末尾添加元素。
 - `PopBack()`：从数组末尾删除元素。
 - `Reverse()`：反转数组中的元素顺序。
 - `operator[]`：重载索引运算符，用于访问数组中的元素。

5.3. GeneTree 类

`GeneTree` 类用于构建家谱树，管理家族成员并执行增、删、改等操作。

- **成员函数：**
 - `BuildGeneTree()`：创建以给定名字为祖先的家谱树。
 - `FindMember()`：通过深度优先查找，找到指定姓名的家族成员节点。
 - `AddChildren()`：为指定家族成员添加儿女。
 - `DeleteSubTree()`：删除指定成员的局部家谱。
 - `DispTree()`：打印整个家谱树结构。
 - `ClearTree()`：清空家谱树。

5.4. 其他辅助类

- **Node**：家谱树节点类，表示家族成员，包含成员姓名、配偶、代际层次及其儿女节点列表。

6. 功能模块介绍

6.1. 祖先家谱树建立

用户通过输入祖先姓名来创建家谱树根节点，代表家族的祖先。系统会存储该节点并以其为根节点生成后续子树。

6.2. 家族成员的查找与信息更新

通过 `FindMember()` 函数查找指定成员的节点。用户可以选择以下操作来管理家谱：

- **添加儿女：** `AddChildren()` 方法用于为已存在的成员添加儿女节点。
- **更改成员姓名：** 可对找到的成员节点直接更新姓名属性。
- **解散局部家庭：** `DeleteSubTree()` 方法用于删除指定成员及其所有后代。

6.3. 家谱树展示

通过 `DispTree()` 函数打印整个家谱树结构。程序会根据代际层次对输出进行缩进，使家谱层次关系一目了然。

7. 用户交互

用户在命令行中选择对应的操作并输入相关信息，系统将执行以下操作：

1. 输入家族祖先的姓名并生成家谱树。
2. 按照用户选择的操作执行家谱成员的添加、查找、修改和删除等操作。
3. 输出家谱树的整体结构，以缩进表示代际关系。

7.1. 示例交互

- **输入：**

```
1  请输入祖先的姓名：李三
2  请选择要执行的操作：A
3  请输入要建立家庭的成员姓名：李三
4  请输入李三的儿女人数：2
5  请依次输入李三的儿女的姓名：李四 王五
```

- **输出：**

```
1  李三的第一代后代是：李四 王五
2  -----家谱的目录结构(缩进表示下一代) -----
3  李三
4      李四
5      王五
6  -----
```

8. 性能考虑

家谱树的成员查找、插入和删除操作的时间复杂度均为 $O(n)$ ，其中 n 是家族成员数量。由于采用深度优先搜索，程序对层次较多的家谱可能会消耗较多内存和时间，但适合处理中小规模的家族树数据。

9. 错误处理和输入验证

为了保证输入的合法性，本项目对输入的成员信息进行了严格的验证：

- **成员查找失败**：在添加或修改成员信息时，若未找到指定成员，系统会提示操作失败。
- **子树删除的边界条件检查**：确保只有非根节点才能被删除。
- **字符大小写兼容**：用户输入的操作字符不区分大小写。

若检测到非法输入，系统会输出提示信息并结束当前操作。

10. 未来改进

1. **增加成员配偶信息**：当前系统未实现对配偶信息的详细管理，未来可扩展支持。
2. **优化展示效果**：使用图形界面进一步优化家谱结构的展示。
3. **添加家谱保存与加载**：提供文件保存和读取功能，使家谱数据可持久化。
4. **扩展成员属性**：支持记录成员的生日、职业等额外信息，使家谱系统更具现实意义。

11. 心得体会

在本项目的开发过程中，加深了我对树结构和深度优先搜索的理解，尤其是在实现家庭成员的层次管理和节点间的动态连接时，遇到较多挑战。同时，通过设计自定义的数据结构 `string` 和 `vector`，增强了对 C++ 内存管理和模板编程的认识。