

Project 3

家谱管理

计科一班 17341009 曾天宇

计科一班 17341015 陈鸿峥

计科二班 17341059 黄杨峻

1 题目要求

本次project要求用树形结构实现家族成员信息管理，如建立、删除、查询、统计、打印等。我们经过讨论，实现了以下六个功能：

- 实现树的输入与生成，录入家族成员的姓名、性别、出生年份、目前年龄(或享年)、以及家庭关系信息，输入采用dot格式输入¹，形如：

```
digraph genealogy{
    HYJ [gender="M", birth="1978", age="40"];
    CHZ [gender="M", birth="1999", age="19"];
    ZTY [gender="M", birth="1998", age="20"];
    WIFE [gender="F", birth="2000", age="18"];
    HYJ -> WIFE [label="mate"];
    HYJ -> CHZ [label="kid"];
    HYJ -> ZTY [label="kid"];
}
```

- 实现树的节点增删，通过人名来检索需要增删的目标。
- 实现树节点的查询，并打印节点，使用人名搜索，若查无此人，输出error message
- 实现树的统计，输出总人数、总代数信息。
- 实现树的打印，输出一张csv表格，并用opml的方式在workflowy上可视化展示。

2 数据结构与算法

我们首先定义了Person类，用于记录每一个家庭成员的信息，并设置了各类信息的get、set函数（如getName、setAge等）。然后我们定义了Family类，供家族系统增删查改信息。我们还定义了一个map<string,Person*>类型的映射，将每一个家族成员的姓名映射到节点，方便查询，以下详细说明程序的实现方法。

2.1 数据结构实现

在本实验中，我们采用了如下的数据结构：

- **Person类**：对单一个人进行储存，存储内容为：姓名、出生年份、年龄、性别、配偶（可选）、父亲姓名、孩子指针列表。整个结构数据默认初始化为：

"" - 1990 - 0 - Male - nullptr - "" - null vector

- **Family类**：对家谱进行整体存储。这里不仅用了树形的数据结构，还使用标准库中的map对索引进行加速。

¹文本图形描述语言dot,[https://en.wikipedia.org/wiki/DOT_\(graph_description_language\)](https://en.wikipedia.org/wiki/DOT_(graph_description_language))

在实现方法上，我们为Person和Family类设计了尽可能详细的访问函数，将尽可能多的代码简并。特别的是，在插入新的节点的过程中，需要同时对家谱树和索引进行添加；同理删除也是需要这两步操作的。由于前面设计了map的数据结构进行辅助，因此在查找的时间复杂度只有 $O(1)$ 。

2.2 输入实现

在本实验中，我们支持多种输入输出方式。考虑到家谱可能十分庞大，故通过一个个结点插入的方式读入显然是不可行的。因此我们采用图形描述语言dot来对家谱进行读入。用于只需按照一定的规则在记事本里定义好家族各个成员之间的联系，并将文件存储为dot格式，即可被我们的程序成功读入。

具体的dot文件撰写分为两个部分，先要声明所有结点（人）的信息，然后对这些结点建立关系。

- 结点：

```
<name> [gender = <M/F>, birth = <year>, age = <age>];
```

- 关系：

```
<name1> -> <name2> [label = <mate/kid>];
```

举例来说，我们要建立名为HYJ这个人的信息，已知他是男的，于1978年出生，年龄40，则我们可以键入

```
HYJ [gender="M", birth="1978", age="40"];
```

如果他有一个妻子，名为WIFE，2000年出生，年龄20，则我们可以键入

```
WIFE [gender="F", birth="2000", age="18"];
```

同时添加关系

```
HYJ -> WIFE [label = "mate"]
```

注意：为了防止有些人早逝导致信息有误，我们还是强制用户输入每个人的年龄。

具体实施采用以下两条正则表达式对输入字符串进行处理

```
*\[ *gender *= *" *| " *\];| +|", *birth *= *"|", age *= *"
*\[ *label *= *"| " *\];| *-> *| +
```

2.3 输出实现

1. xml格式²输出

由于xml本身就具有树的性质，故我们可以利用xml格式进行输出，通过输出结点的层次结构可直观看到家谱的形状。例子如下。

```
<family_tree>
  <node>
    <name>HYJ</name>
    <birth>1978</birth>
    <age>20</age>
    <kids>
      <node>
        <name>CHZ</name>
        <birth>1999</birth>
      </node>
    </kids>
    <mate>WIFE</mate>
  </node>
</family_tree>
```

2. csv表格³输出

利用csv格式输出也是十分直观的方式，将每个人的信息一一罗列在同一行上，输出文件用Excel表即可打开（该方法借鉴了银行流水报告的输出）

3. workflowy⁴可视化

我们对前面的xml文件进一步升级，得到opml格式文件⁵，以便输入workflowy进行可视化

3 测试数据、结果及分析

实验测试结果如图1至图7所示。可以看见，我们的程序正常执行，输出了正确的结果，并且给出了多种直观的可视化方案。

²可扩展标记语言xml,<https://en.wikipedia.org/wiki/XML>

³逗号分隔符csv,https://en.wikipedia.org/wiki/Comma-separated_values

⁴Workflowy,<https://workflowy.com/>

⁵大纲处理标记语言opml,<https://en.wikipedia.org/wiki/OPML>

```

Welcome to the GenealogyManagement system.

Copyright (C) 2018 Tianyu Zeng, Hongzheng Chen, Yangjun Huang. All rights reserved.

Help -h (by file tree.in)
Input / Refresh Info -r
Insert -i
Summary -S
Search Info -s (by name)
Delete Info -d (by name)
Display Info -D
Exit -e
>>> r
Begin parsing...
Parsed dot file successfully!

Help -h (by file tree.in)
Input / Refresh Info -r
Insert -i
Summary -S
Search Info -s (by name)
Delete Info -d (by name)
Display Info -D
Exit -e
>>> D
Starting output...
Name Gender Age Birth Parent Mate
LuoJun Male 62 1956
HYJ Male 42 1976 LuoJun HYJWIFE
CHZ Male 20 1998 HYJ
ZTY Male 20 1998 HYJ
WYF Male 28 1990 HYJ CXK
Finish output. All the data have been outputted to ./tree.opml and ./tree.csv !

```

图 1: 输入输出(IO)

```

>>> i
Please enter the name of the person:
Kid
Please enter the gender of the person: (M)ale (F)emale
M
Please enter the age of the person:
19
Please enter the birth year of the person:
1988
Please enter the father's name of the person:
LuoJun
Please enter the mate's name of the person (If no mate just enter a 'null')
null
Insert successfully!

```

图 2: 插入结点

```

>>> S
The total generation of your family is: 3
There are 9 person in your family.

Help -h (by file tree.in)
Input / Refresh Info -r
Insert -i
Summary -S
Search Info -s (by name)
Delete Info -d (by name)
Display Info -D
Exit -e
>>> s
Please enter the name of the person: Kid
Name: Kid
Gender: Male
Age: 19
Birth Year: 1988

```

图 3: 单一数据查询及多数据展示

```
>>> d
Please enter the name of the person: Kid
Delete Kid successfully!
```

图 4: 删除结点

| | A | B | C | D | E | F |
|---|--------|--------|-----|-------|--------|---------|
| 1 | Name | Gender | Age | Birth | Parent | Mate |
| 2 | LuoJun | Male | 62 | 1956 | | LJWIFE |
| 3 | HYJ | Male | 42 | 1976 | LuoJun | HYJWIFE |
| 4 | CHZ | Male | 20 | 1998 | HYJ | |
| 5 | ZTY | Male | 20 | 1998 | HYJ | |
| 6 | WYF | Male | 28 | 1990 | HYJ | CXK |

图 5: csv文件输出

4 分工、贡献与自我评分

| | 分工 | 贡献度 | 自我评分 |
|-----|--------------------------------|------|-------|
| 曾天宇 | 实现Person类、Family类，测试修改代码、写实验报告 | 0.33 | 10/10 |
| 陈鸿峥 | 实现输入输出功能、测试修改代码、实验报告撰写 | 0.33 | 10/10 |
| 黄杨峻 | 实现输入输出界面、生成输出、测试修改代码、写实验报告 | 0.33 | 10/10 |

5 项目总结

这次项目难度不高，涉及到的知识都是树最基础的删改查找，但最困扰我们的问题就是输入输出，我们几乎花了一节课的时间去讨论这个问题，直到最后才删删改改实现了一个大家都比较满意的方案。不过我们这次项目开始之前，先做好了document，规范好了接口，并在个人服务器上建立git仓库，共享代码（图8是我们的一些commit的记录），给各自的工作带来不少便利，写代码效率高了不少。

```

1  <?xml version="1.0"?>
2  <opml version="2.0">
3    <head>
4      <ownerEmail>1023198294@qq.com</ownerEmail>
5    </head>
6    <body>
7      <outline text = "family_tree" >
8
9        <outline text ="person" >
10         <outline text ="name : LuoJun" />
11         <outline text ="birthyear : 1956" />
12         <outline text ="age : 62" />
13         <outline text ="mate : LJWIFE" />
14         <outline text ="kid :" >
15           <outline text ="person" >
16             <outline text ="name : HYJ" />
17             <outline text ="birthyear : 1976" />
18             <outline text ="age : 42" />
19             <outline text ="mate : HYJWIFE" />
20             <outline text ="kid :" >
21               <outline text ="person" >
22                 <outline text ="name : CHZ" />
23                 <outline text ="birthyear : 1998" />
24                 <outline text ="age : 20" />
25               </outline>
26             </outline>
27             <outline text ="kid :" >
28               <outline text ="person" >
29                 <outline text ="name : ZTY" />
30                 <outline text ="birthyear : 1998" />
31                 <outline text ="age : 20" />
32               </outline>
33             </outline>
34             <outline text ="kid :" >
35               <outline text ="person" >
36                 <outline text ="name : WYF" />
37                 <outline text ="birthyear : 1990" />
38                 <outline text ="age : 28" />
39                 <outline text ="mate : CXK" />
40               </outline>
41             </outline>
42           </outline>
43         </outline>
44       </outline>
45     </outline>
46   </body>
47 </opml>

```

图 6: OPML输出

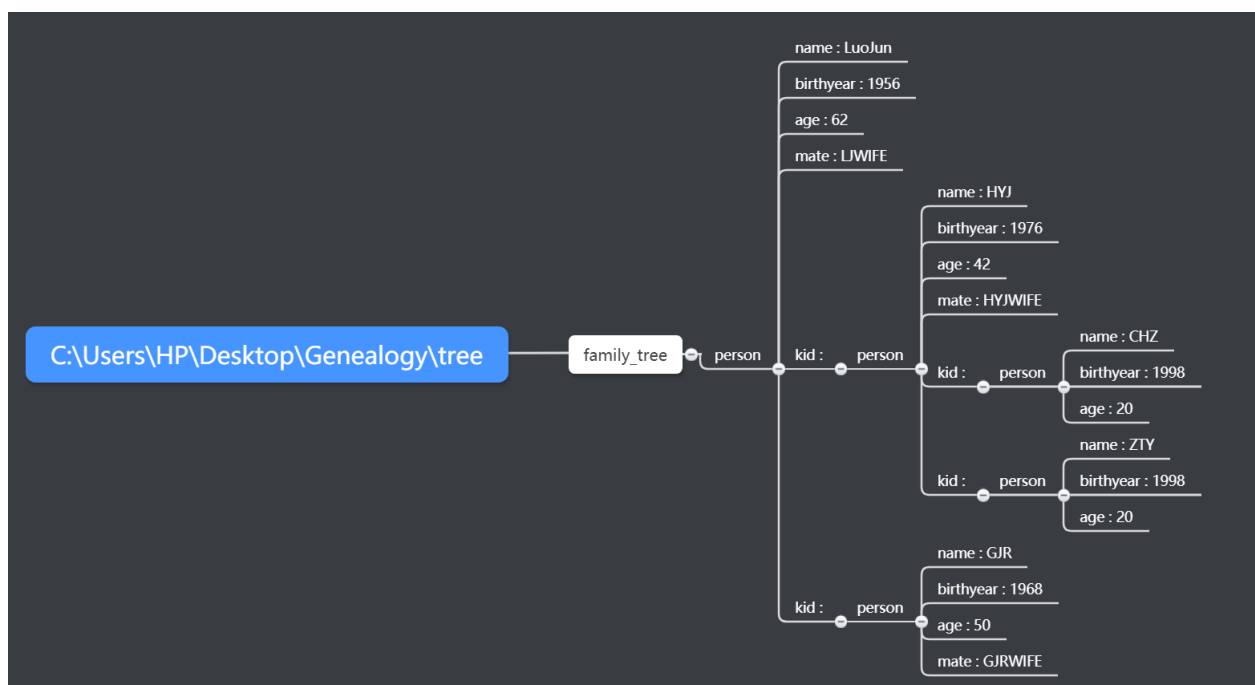


图 7: 可视化结果

```

commit 45ff93baf2e682d6248b945639c63355634e8e1c
Author: Zeng Tianyu <zengtianyu1215@gmail.com>
Date:   Mon Nov 12 22:04:48 2018 +0800

    fix delete bugs

commit 2979ae3af0d52ffe0827383b9fd8c9aa62afd977
Author: Huang Yangjun <1023198294@qq.com>
Date:   Sun Nov 11 14:50:08 2018 +0800

    hyj edit output function changes

commit 3e07e0a2429ba622378af22a018f08a8739b538a
Author: Huang Yangjun <1023198294@qq.com>
Date:   Sun Nov 11 14:04:00 2018 +0800

    hyj changes on output functions

commit 6eb22507ba446589fa055585f94f4b1ee4dfff5d
Author: chhzh123 <30967989+chhzh123@users.noreply.github.com>
Date:   Sat Nov 10 23:34:09 2018 +0800

    Finish IO

commit 478fd741ae895618a2220fe53c97936ee91207a1
Author: chhzh123 <30967989+chhzh123@users.noreply.github.com>
Date:   Sat Nov 10 18:25:28 2018 +0800

    add const
  
```

图 8: Git commit记录