

Project3

家族族谱树

班级：教务 4 班 / 行政 8 班

姓名：钟文谦 学号：16340309

姓名：周伟标 学号：16340310

姓名：周泽昊 学号：16340312

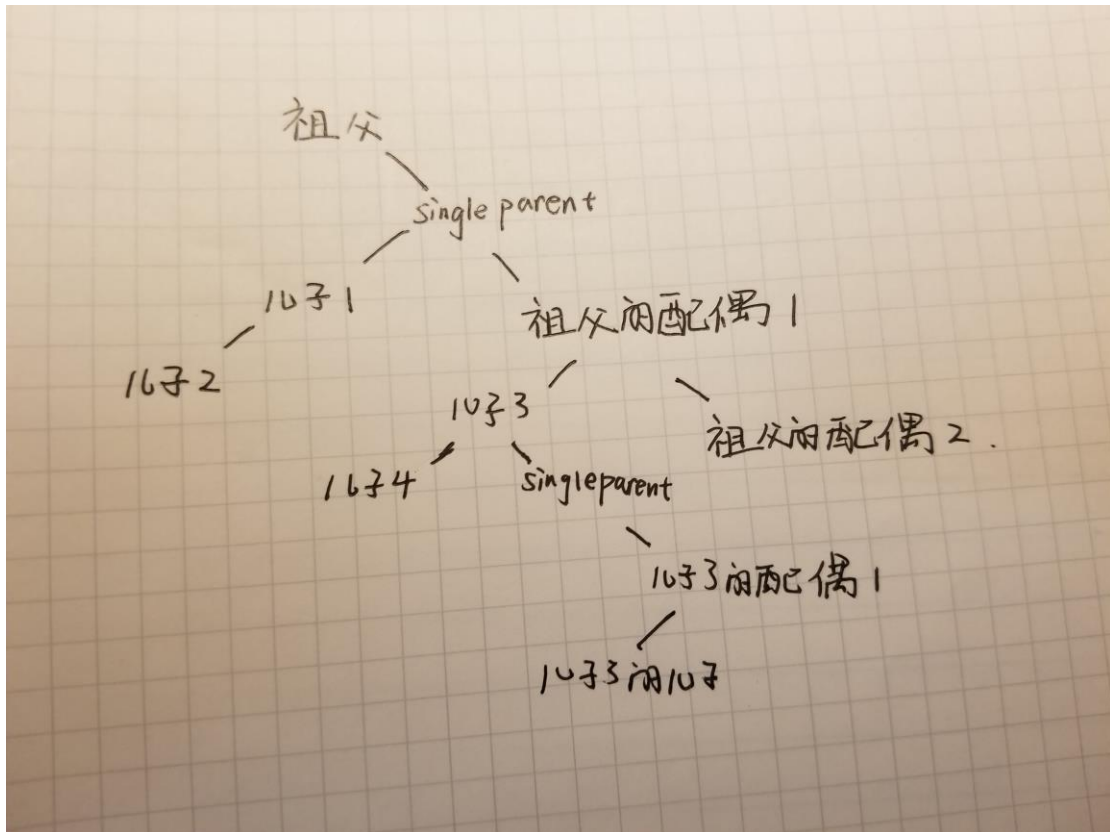
【题目要求】

用树的结构完成家谱管理程序，其基本的功能含有插入孩子、插入配偶、删除家庭成员、更新家庭成员信息、查询家庭成员、查询某家庭成员的双亲以及打印家谱。

我们小组在基础功能之上加入了通过多种信息/条件筛选符合条件的成员、对该家庭中的成员进行数据统计分析（如：男女比例、家庭人数、平均死亡年龄等的信息），也做了文件储存家谱以及对已有的家谱进行读取的功能。

【数据结构与算法】

经过对于该程序的需求的多次辩论以及思考，我们决定使用一棵带有特殊关系的树来存储与表示家庭族谱。本家中成员的右侧节点全部为该成员的配偶，而这些配偶节点的做节点全部为该配偶所生的子。就如下图一般



(singleParent 节点是我们的程序赋予本家的人的一个特性，收养的孩子就属于该节点)

结合上图以及上面的介绍就可以很清楚地知道该树是如何表达各成员的关系的。

下面我们分 FamilyTree 以及 UI 部分对算法进行分析：

FamilyTree 部分：

FamilyTree 部分添加、删除节点没什么特别的地方，我们比较特别的地方就是受到大一的实训项目启发，想到了建立一个 queryPerson 函数，通过这个函数使得 UI 部分的编写更加灵活，FamilyTree 部分也因此减少了一些不必要的函数、代码。

各个函数分析如下：

1) 添加祖先：void insertAncestor(personNode* person)

直接更改根节点 root 为 person 即可

2) 添加孩子: `void insertChild(personNode* person, personNode* parent);`

由于通过 UI 判断了 parent 是否合法, 这里可以直接找到 parent 左边节点的最后一个节点插入 person 即可

3) 添加配偶: `void insertPartner (personNode* person, personNode* partner)`

同理, 可以直接通过在 partner 的最右边节点插入 person 即可

4) 删除成员: `void deleteMember (personNode* person)`

这里分为 2 种情况

① 如果是本家的人那么删除会将他(她)的右子树(即配偶和孩子)全部删掉

② 如果是入赘或者是嫁入的人那么会将他(她)的左子树(即孩子)全部删掉

5) 筛选符合条件的人:

```
vector<personNode*> queryPerson(function<bool (const  
    personNode*)>filter)
```

```
vector<personNode*>queryPersonFromVector (vector<personNode*>  
    data, function<bool (const personNode*)> filter)
```

通过递归的方式遍历, 寻找符合该函数的成员即可 (C++11 新特性)

6) 对选中成员进行更改:

`vector<personNode*>updatePerson(`

`function<bool(const personNode*)> filter,`

`function<bool(const personNode*)> updater)`

通过递归遍历，对符合 `filter` 的成员执行 `updater` 操作

7) 寻找双亲：`vector<personNode*> queryParent(personNode* person)`

先遍历寻找到从根节点到 `person` 路径，将路径存储在 `vector` 中，

在寻找从 `person` 到根节点路径中第一个入赘或嫁入(即 `foreign ==`

`1`) 的人然后再找到第一个是本家的人(即 `foreign == 0`)

8) 对符合条件的人数进行统计：`int count(function<bool(const personNode*)> filter)`

直接调用 `queryPerson`，返回 `vector` 的 `size`

9) 输出：`void printAllPerson()`

通过递归调用输出树形图，如果没有领养孩子不会输出

`singleParent`，配偶前面是|__，孩子前面是||_，配偶会在其丈夫

(妻子)的正下方，孩子会向右偏移一格，以此达到家谱效果的输出

10) 判断是否为空：`bool empty()`

直接返回根节点是否为空

11) 从文件读取：`bool read(string filename);`

从文件读取(按顺序储存的顺序读入数据)

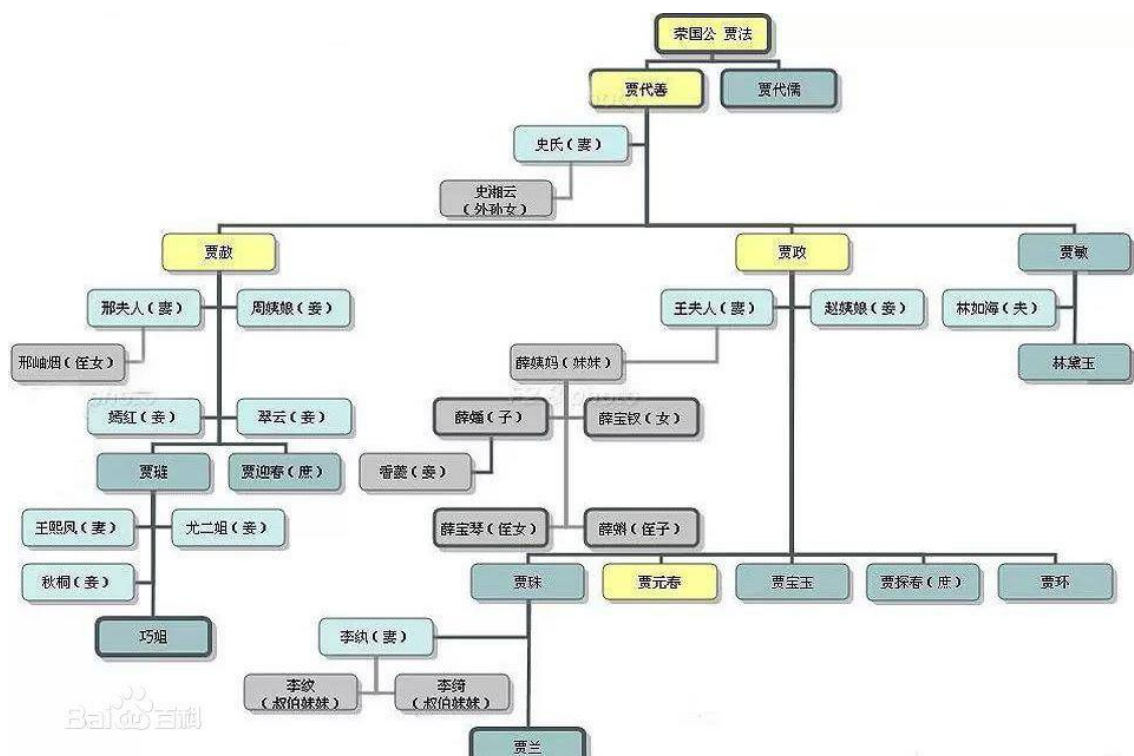
12) 保存到图片：`bool save(string filename)`

保存到文件（通过顺序遍历）

FamilyTreeUI 部分:

经过后端，即 FamilyTree 合理的设计，在 UI 部分，各个函数的实现都很简单，基本就是通过 queryPerson 函数找到对应需要操作的节点然后传给 FamilyTree 处理。我们的数据统计部分也因为这种合理的设计，得以在前端 UI 部分进行处理，让后端只需处理树的部分。这样对于我们的合作开发更加有利。因为 UI 部分函数较多且没有什么特殊的部分，UI 部分在这里就不一一分析函数了。

【测试数据、结果与分析】



我们寻找的测试案例就是红楼梦的家谱（如上图），对于一些通过百度不知道母亲是谁的人，我们用单亲节点来定义它们，侄子侄女脱离范

围，不计入该家谱。

(注：我们的表示规则为本家节点的配偶节点为其正下方所有用|___链接着的所有节点直到下一个||___即直到下一个本家节点，其孩子则为生下该孩子的配偶节点名字正下方用||___接着的所有节点直到其下一个配偶节点或其兄弟节点。本程序采用的打印家谱方式按照辈分进行缩进。我们可以根据缩进很清晰发现同代的成员和其下一代成员)

1) 添加祖先

```
选择D:\文档\项目1.exe
你有已有的家谱进行输入吗(0: 无, 1: 有)? 0

请输入祖先的信息:
格式为:
[名字] [性别(0:男, 1: 女)] [年龄 (age > 0)] [是否死亡(0: false/1: true)]
贾法 0 80 1

-----家谱小程序-----
0 - 读取已有家谱
1 - 插入孩子
2 - 插入配偶
3 - 删除家庭成员
4 - 更新家庭成员信息
5 - 查找符合信息的成员
6 - 查询家庭某成员的双亲
7 - 打印家谱
8 - 数据统计
9 - 存储家谱
10 - 离开

-----
请输入你的选择: 1

请输入要插入的孩子的信息:
格式为:
[名字] [性别(0:男, 1: 女)] [年龄 (age > 0)] [是否死亡(0: false/1: true)]
贾代善 0 60 0
请输入母亲或者父亲的名字: 贾法
```

2) 加孩子，若输入名字为本家中的人(即非配偶)，我们会找出其配偶节点给用户进行选择

若此时只有一个配偶节点，即其自带的特性 singleParent 节点，我们自动将其插入 singleParent 节点

```

选择D:\文档\项目1.exe
10 - 离开
-----
请输入你的选择: 1
请输入要插入的孩子的信息:
格式为:
[名字] [性别(0:男, 1: 女)] [年龄 (age > 0)] [是否死亡(0: false/1: true)]
贾代善 0 60 0
请输入母亲或者父亲的名字: 贾法
-----家谱小程序-----
0 - 读取已有家谱
1 - 插入孩子
2 - 插入配偶
3 - 删除家庭成员
4 - 更新家庭成员信息
5 - 查找符合信息的成员
6 - 查询家庭某成员的双亲
7 - 打印家谱
8 - 数据统计
9 - 存储家谱
10 - 离开
-----
请输入你的选择: 7
贾法(死)
|__singleParent
|  |__贾代善

```

3) 加配偶

```

请输入你的选择: 2
请输入要插入的配偶的信息:
格式为:
[名字] [性别(0:男, 1: 女)] [年龄 (age > 0)] [是否死亡(0: false/1: true)]
史氏 1 55 0
请输入该插入节点配偶的名字: 贾代善
-----家谱小程序-----
0 - 读取已有家谱
1 - 插入孩子
2 - 插入配偶
3 - 删除家庭成员
4 - 更新家庭成员信息
5 - 查找符合信息的成员
6 - 查询家庭某成员的双亲
7 - 打印家谱
8 - 数据统计
9 - 存储家谱
10 - 离开
-----
请输入你的选择: 7
贾法(死)
|__singleParent
|  |__贾代善
|  |__史氏
|  |__贾代儒

```

4) 红楼梦家谱按照图片全部录入完成结果如下:


```
-----家谱小程序-----
0 - 读取已有家谱
1 - 插入孩子
2 - 插入配偶
3 - 删除家庭成员
4 - 更新家庭成员信息
5 - 查找符合信息的成员
6 - 查询家庭某成员的双亲
7 - 打印家谱
8 - 数据统计
9 - 存储家谱
10 - 离开
```

请输入你的选择: 7

```
贾法(死)
|__singleParent
|   |__贾代善
|   |   |__史氏
|   |       |__贾赦
|   |           |__singleParent
|   |               |__贾琏
|   |                   |__王熙凤
|   |                       |__巧姐(死)
|   |                           |__尤二姐(死)
|   |                               |__秋桐
|   |                                   |__邢夫人
|   |                                       |__周姨娘
|   |                                           |__贾迎春
|   |                                               |__嫣红
|   |                                                   |__翠云
|   |                                                       |__贾政
|   |                                                           |__王夫人
|   |                                                               |__贾珠(死)
|   |                                                                   |__李纨
|   |                                                                       |__贾兰
|   |                                                                           |__贾元春
|   |                                                                               |__贾宝玉
|   |                                         |__赵姨娘
|   |                                             |__贾探春
|   |                                                 |__贾环
|   |             |__贾敏
|   |                 |__林如海
|   |                     |__林黛玉
|   |                         |__贾代儒
```

5) 查询双亲

```

选择D:\文档\项目1.exe
6 - 查询家庭某成员的双亲
7 - 打印家谱
8 - 数据统计
9 - 存储家谱
10 - 离开

-----
请输入你的选择: 6
请输入你想查询双亲的家庭成员的名字: 巧姐
母亲:

-----
Index: 13
姓名: 王熙凤
年龄: 18
性别: 女
是否为外来人口: True
是否已去世: Alive
是否已离婚: True

-----
父亲:

-----
Index: 8
姓名: 贾琏
年龄: 25
性别: 男
是否为外来人口: False
是否已去世: Alive
是否已离婚: False

```

6) 删除成员（会删除附属该节点的所有节点）

```

-----家谱小程序-----
0 - 读取已有家谱
1 - 插入孩子
2 - 插入配偶
3 - 删除家庭成员
4 - 更新家庭成员信息
5 - 查找符合信息的成员
6 - 查询家庭某成员的双亲
7 - 打印家谱
8 - 数据统计
9 - 存储家谱
10 - 离开

-----
请输入你的选择: 3
请输入你想删除的家庭成员的名字: 贾敏

```

请输入你的选择：7

贾法(死)

|__singleParent

|__贾代善

|__史氏

|__贾赦

|__singleParent

|__贾琏

|__王熙凤

||__巧姐(死)

||__尤二姐(死)

||__秋桐

|__邢夫人

|__周姨娘

|__贾迎春

|__嫣红

|__翠云

|__贾政

|__王夫人

|__贾珠(死)

|__李纨

||__贾兰

|__贾元春

|__贾宝玉

|__赵姨娘

|__贾探春

|__贾环

||__贾代儒

7) 筛选符合条件的成员

```
请重新输入:5
这里有五个提供筛选的标准:
0: 姓名, 1: 年龄, 2: 性别, 3: 是否死亡, 4: 是否离婚
输入格式为一串以空格为间隔, 以-1为结尾的数字串, 数字串的内容即为你要选择得到标准。
例子: 0 1 4 -1 为选择姓名、年龄、是否离婚作为筛选标准
请输入你的选择: 1 2 -1
请输入你要筛选的人的年龄的范围(格式为前一个数字是该范围的下限, 后面的是该范围的上限, 中间以空格隔开):
例子: 13 40 即为筛选13-40岁的人
请输入: 20 60
请输入你要筛选的人的性别(0: 男, 1: 女):0
筛选出的结果为:
结果 0:
```

```
-----
Index: 1
姓名: 贾代善
年龄: 60
性别: 男
是否为外来人口: False
是否已去世: Alive
是否已离婚: False
-----
```

结果 1:

```
-----
Index: 4
姓名: 贾赦
年龄: 40
性别: 男
是否为外来人口: False
是否已去世: Alive
是否已离婚: False
-----
```

结果 2:

```
-----
Index: 6
姓名: 贾琏
年龄: 25
性别: 男
是否为外来人口: False
是否已去世: Alive
是否已离婚: False
-----
```

结果 3:

Index: 5
姓名: 贾政
年龄: 38
性别: 男
是否为外来人口: False
是否已去世: Alive
是否已离婚: False

结果 4:

Index: 20
姓名: 贾宝玉
年龄: 35
性别: 男
是否为外来人口: False
是否已去世: Alive
是否已离婚: False

结果 5:

Index: 2
姓名: 贾代儒
年龄: 59
性别: 男
是否为外来人口: False
是否已去世: Alive
是否已离婚: False

成功筛选出我们想选中的成员

8) 信息更新

请输入你的选择: 4
请输入你要更新信息的成员的名称: 巧姐
当前信息如下:

Index: 16
姓名: 巧姐
年龄: 10
性别: 女
是否为外来人口: False
是否已去世: Dead
是否已离婚: False

请输入更新后的信息:

格式为:
[名字] [性别(0:男, 1: 女)] [年龄] [是否死亡(0: false/1: true)] [是否已离婚(0: false/1: true)]
巧姐 1 10 0

0

```

请输入你的选择: 7
贾法(死)
|_singleParent
|   |_贾代善
|       |_史氏
|           |_贾赦
|               |_singleParent
|                   |_贾琏
|                       |_王熙凤
|                           |_巧姐
|                               |_尤二姐(死)
|                                   秘制

```

更新完成

9) 统计功能: 平均年龄, 死亡年龄, 男女占比, 总人数

```

-----数据分析仪-----
1 - 平均年龄
2 - 平均死亡年龄
3 - 男女比例及人数
4 - 总人数

```

```

请输入你的选择: 1.
这里有 22 个健在的家庭成员!
平均年龄(存活的人)为 33.7273!

```

```

-----数据分析仪-----
1 - 平均年龄
2 - 平均死亡年龄
3 - 男女比例及人数
4 - 总人数

```

```

请输入你的选择: 2
这里有 3 个已经去世的家庭成员!
平均死亡年龄为 46!

```

```

-----数据分析仪-----
1 - 平均年龄
2 - 平均死亡年龄
3 - 男女比例及人数
4 - 总人数

```

```

请输入你的选择: 3
这里有 7 个男成员, 18 个女成员!
男性占比 28%, 女性占比 72%.

```

请输入你的选择：8

数据分析仪

1 - 平均年龄

2 - 平均死亡年龄

3 - 男女比例及人数

4 - 总人数

请输入你的选择：4

这里总共有 25 个家庭成员(包括已去世与健在的人)!

【分工、贡献%、自我评分】

钟文谦：完成程序基本架构的设计，完成程序中的 FamilyTreeUI, 35%,
90 分

周泽昊：完成程序中的 FamilyTree 部分，35%，90 分

周伟标：测试程序，30%，90 分

【项目总结】

经过这一次项目，

1. 我们的小组对于 Tree 的遍历，插入节点，删除节点，寻找双亲，树的储存以及还原更加熟悉，理解更加深刻，平常有做到的各种算法题中学到的算法也在部分函数中得以应用。
2. 我们也对于如何根据需求确认相应的数据结构有了更深的理解，像该 project 中对于如何用树来表示父子、配偶、母子、兄弟等关系的设计，这些操作对于我们实际开发能力的增长都有所帮助。
3. 对于如何设计程序架构，使得各部分的功能的开发可以更好地分离，从而方便合作开发，使得我们的开发内容不会重叠，方便了我们各自的开发。
4. 这一次次项目，让我们发现了编程这一样东西，可以解决我

们生活中的很多的实际问题。在今后的生活中，我们会更加地注意去发现生活中的各种问题，或者各种有趣的想法，然后通过编程去解决，应用我们平时学到的各方面的知识，从而在编程这一块获得更多的乐趣与享受，当我们完成一样这样的事情时，我们会获得非凡的满足感和成就感。从而激励我们的继续前进。

【程序清单】

(注:该程序因为应用了 `c++11` 特性，编译时请添加 `-std=c++11` 编译选项)

程序的源代码都置于 `code` 文件夹下，因为长度关系，这里就不一一粘贴了。