项目说明文档

数据结构课程设计

——家谱管理系统

作 者 姓 名：安江涛

学　　号：1952560

指 导 教 师：张颖

学院、 专业：软件学院 软件工程

同济大学

Tongji University

目　　录

1 分析

1.1 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。

2 设计

2.1 数据结构设计

由上可知，需要实现的功能为查找，插入，修改，删除等，要实现搞笑的插入和删除，并且支持父子节点的数据结构显然要使用链表。

2.2 类结构设计

本项目的链表实现共有两个类，一个是 node 类表示一个节点的信息，包括本身的名字，父节点，子节点以及兄弟节点。一个是 tree 类，是 node 类的一个集合，表示整个家谱信息。

2.3 成员与操作设计

node 类

```
struct node {
    node() {
        parent = son = brother = nullptr;
    }
    std::string name;//存名字
    node* parent;//存父节点
    node* son;//存儿子
    node* brother;//存兄弟
};
```

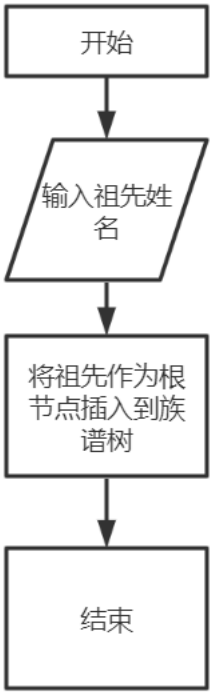
tree 类

```
class tree {
public:
    tree() :root(new node()) {}
    tree(node* r) :root(r) {}
    ~tree() { clear(this->root); }

    node* get_root()const { return this->root; }
    void init();
    void complete();
    void add();
    void change();
    void clear(node* r);
    void erase();
    void print(node* r, int deep, std::vector<int> width);

private:
    bool cin_check();//恢复cin
    void insert(node* r, std::string str);
    node* find(node* r, std::string str);

private:
    node* root;
    std::vector<int> width;
};
```

3 实现

3.1 初始化

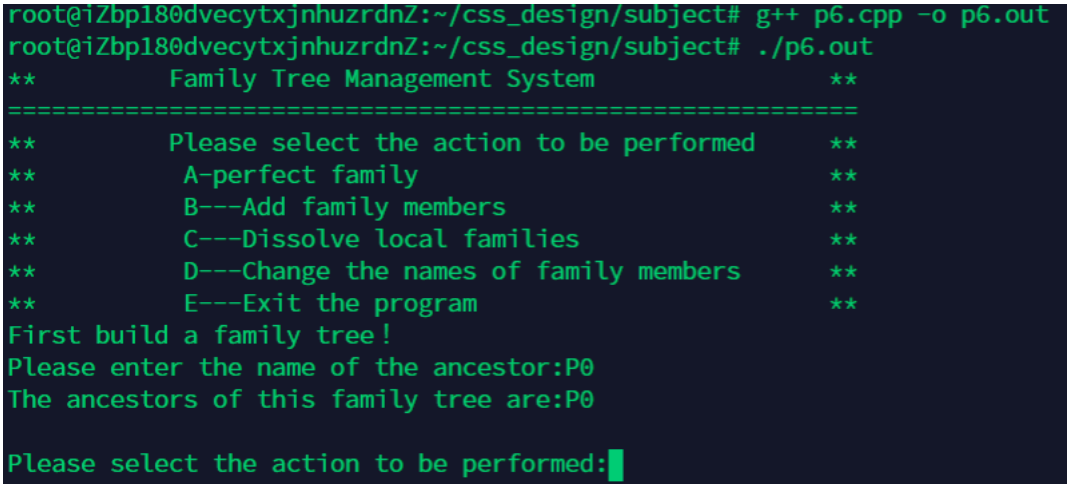
3.1.1 流程图



3.1.2 核心代码

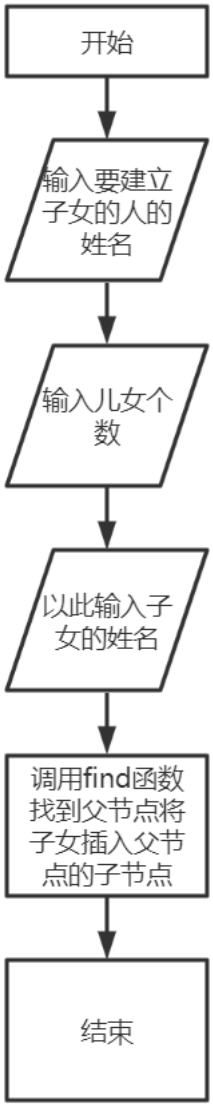
```
void tree::init() {
    std::cout << "First build a family tree！ " << '\n';
    std::cout << "Please enter the name of the ancestor:";
    std::string str;
    std::cin >> str;
    while (!cin_check())
        std::cin >> str;
    this->root->name = str;
    std::cout << "The ancestors of this family tree are:" << str << '\n' << '\n';
}
```

3.1.3 截屏示例



3.2 完善家庭

3.2.1 流程图

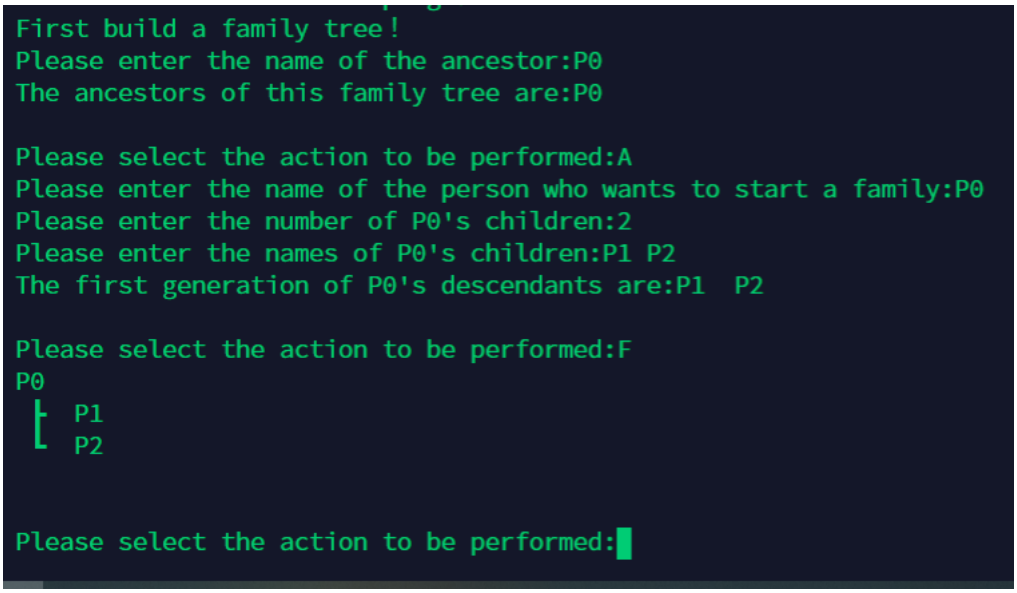


3.2.2 核心代码

```
void tree::insert(node* r, std::string str) {
    if (!r)
        return;
    if (!r->son) {
        r->son = new node();
        r->son->name = str;
        r->son->parent = r;
    }
    else {
        r = r->son;
        while (r->brother)
            r = r->brother;
        r->brother = new node();
        r->brother->name = str;
    }
}

node* tree::find(node* r, std::string str) {
    node* ans = new node();
    ans = nullptr;
    if (!r)
        return ans;
    if (r->name == str)
        return r;
    else {
        node* q = r->son;
        while (q && !(ans = find(q, str)))
            q = q->brother;
    }
    return ans;
}
```

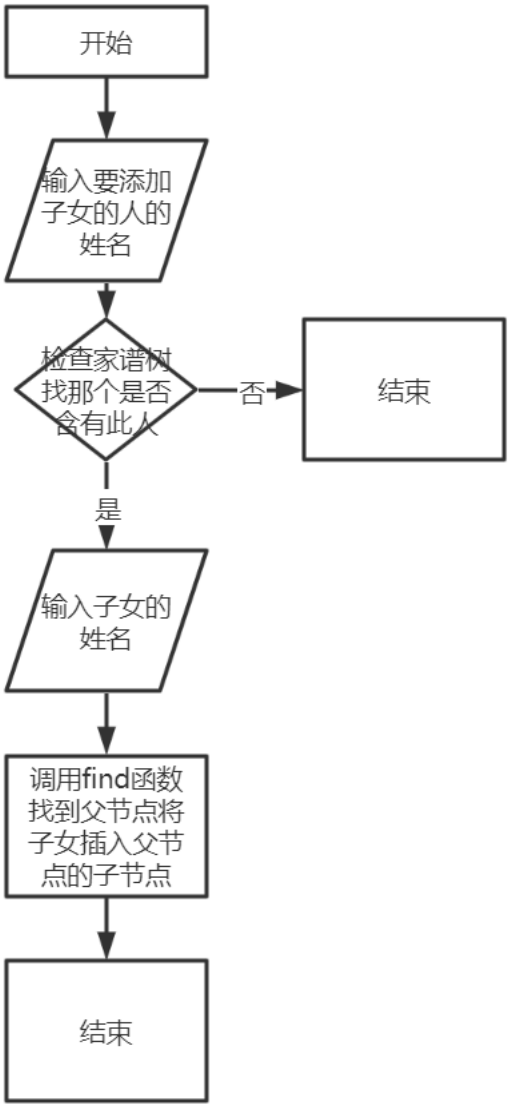
```
void tree::complete() {
    std::cout << "Please enter the name of the person who wants to start a family:";
    std::string str;
    std::cin >> str;
    while (!cin_check())
        std::cin >> str;
    node* r = this->find(this->root, str);
    if (!r) {
        std::cout << "The person's name is not in the family tree！ " << '\n' << '\n';
        return;
    }
    int num = 0;
    std::vector<std::string> vec;
    std::cout << "Please enter the number of " << str << "'s children:";
    std::cin >> num;
    while (!cin_check())
        std::cin >> num;
    std::cout << "Please enter the names of " << str << "'s children:";
    while (num--) {
        std::string name;
        std::cin >> name;
        while (!cin_check())
            std::cin >> name;
        vec.push_back(name);
    }
    for (int i = 0; i < vec.size(); i++) {
        this->insert(r, vec[i]);
    }
    std::cout << "The first generation of "<<str << "'s descendants are:";
    for (int i = 0; i < vec.size(); i++)
        std::cout << vec[i] << "  ";
    std::cout << '\n' << '\n';
}
```

3.2.3 截屏示例


First build a family tree!
Please enter the name of the ancestor:P0
The ancestors of this family tree are:P0
Please select the action to be performed:A
Please enter the name of the person who wants to start a family:P0
Please enter the number of P0's children:2
Please enter the names of P0's children:P1 P2
The first generation of P0's descendants are:P1 P2
Please select the action to be performed:F
P0
├ P1
└ P2
Please select the action to be performed:

3.3 添加家庭成员

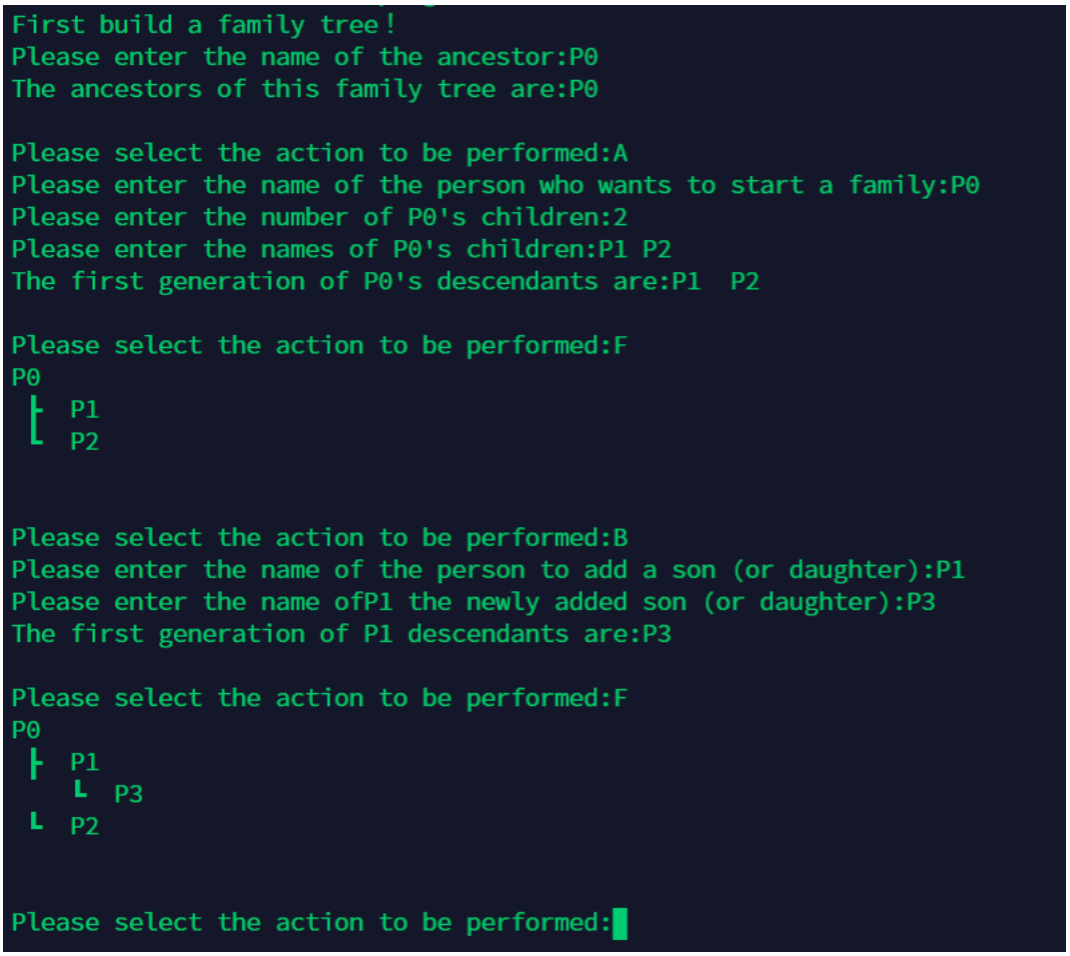
3.3.1 流程图



3.3.2 核心代码

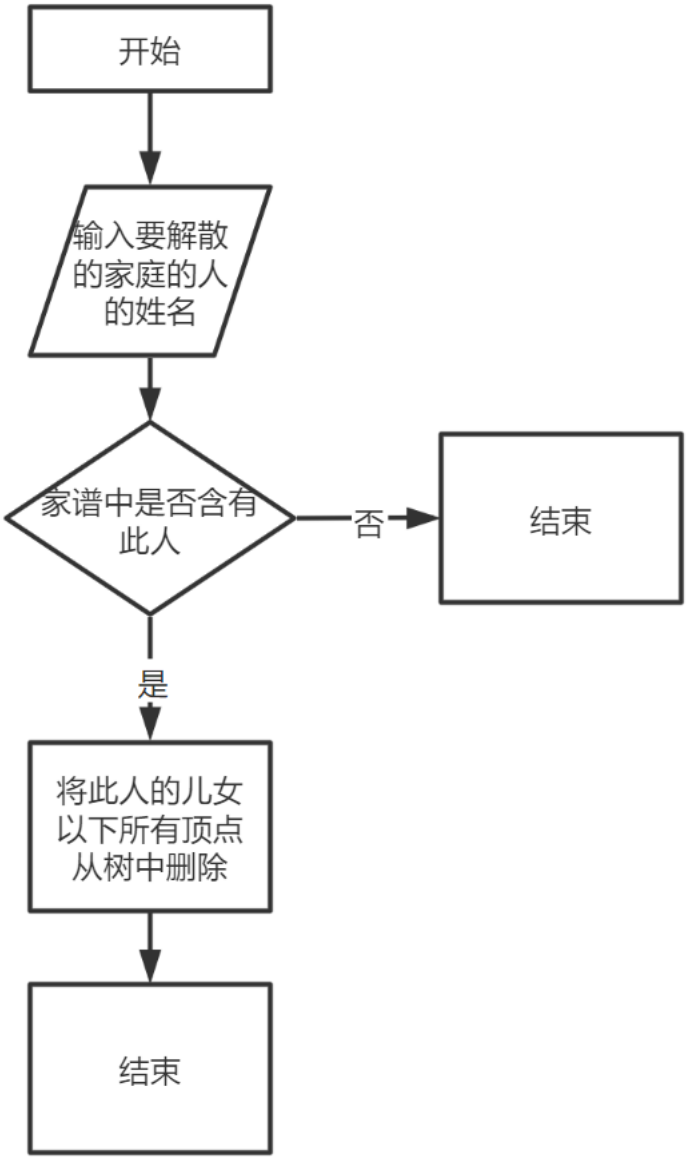
```
void tree::add() {
    std::cout << "Please enter the name of the person to add a son (or daughter):";
    std::string str;
    std::cin >> str;
    while (!cin_check())
        std::cin >> str;
    node* r = find(root, str);
    if (!r) {
        std::cout << "No such person in the family tree！ " << '\n' << '\n';
        return;
    }
    std::cout << "Please enter the name of" << str << " the newly added son (or daughter):";

    std::string name;
    std::cin >> name;
    while (!cin_check())
        std::cin >> name;
    this->insert(r, name);
    std::cout <<"The first generation of "<< str << " descendants are:";
    r = r->son;
    while (r) {
        std::cout << r->name << "  ";
        r = r->brother;
    }
    std::cout << '\n' << '\n';
}
```

3.3.3 截屏示例


First build a family tree!
Please enter the name of the ancestor:P0
The ancestors of this family tree are:P0
Please select the action to be performed:A
Please enter the name of the person who wants to start a family:P0
Please enter the number of P0's children:2
Please enter the names of P0's children:P1 P2
The first generation of P0's descendants are:P1 P2
Please select the action to be performed:F
P0
P1
P2
Please select the action to be performed:B
Please enter the name of the person to add a son (or daughter):P1
Please enter the name ofP1 the newly added son (or daughter):P3
The first generation of P1 descendants are:P3
Please select the action to be performed:F
P0
P1
P3
P2
Please select the action to be performed:

3.4 解散局部家庭

3.4.1 流程图



3.4.2 核心代码

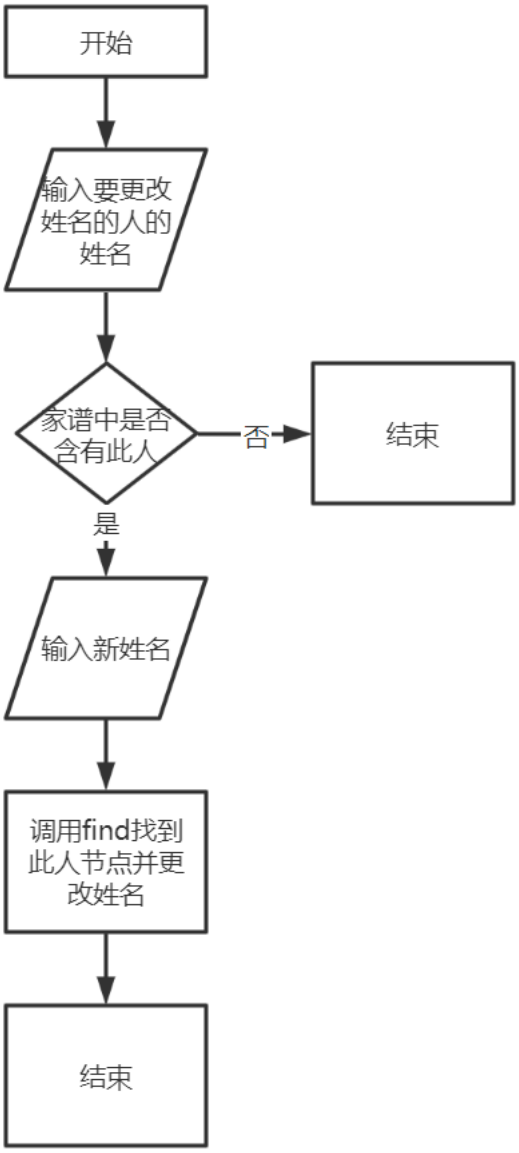
```
void tree::erase() {
    std::cout << "Please enter the name of the person in the family to be dissolved：";
    std::string str;
    std::cin >> str;
    while (!cin_check()) {
        std::cin >> str;
    }
    node* r = this->find(root, str);
    if (!r) {
        std::cout << "No such person in the family tree！" << '\n' << '\n';
        return;
    }
    node* p, * q;
    p = r->son;
    std::cout << "The person who wants to dissolve the family is:" << str << '\n';
    std::cout <<"The first generation of "<< str << " descendants are:";
    while (p) {
        std::cout << p->name << "  ";
        q = p->brother;
        delete p;
        p = q;
    }
    r->son = nullptr;
    std::cout << '\n' << '\n';
}
```

3.4.3 截屏示例

```
First build a family tree!
Please enter the name of the ancestor:P0
The ancestors of this family tree are:P0

Please select the action to be performed:A
Please enter the name of the person who wants to start a family:P0
Please enter the number of P0's children:2
Please enter the names of P0's children:P1 P2
The first generation of P0's descendants are:P1  P2

Please select the action to be performed:F
P0
 ├─ P1
 └─ P2


Please select the action to be performed:B
Please enter the name of the person to add a son (or daughter):P1
Please enter the name ofP1 the newly added son (or daughter):P3
The first generation of P1 descendants are:P3

Please select the action to be performed:F
P0
 ├─ P1
 │   └─ P3
 └─ P2


Please select the action to be performed:C
Please enter the name of the person in the family to be dissolved：P1
The person who wants to dissolve the family is:P1
The first generation of P1 descendants are:P3

Please select the action to be performed:F
P0
 ├─ P1
 └─ P2


Please select the action to be performed:C
Please enter the name of the person in the family to be dissolved：P3
No such person in the family tree!

Please select the action to be performed:
```

3.5 更改家庭成员姓名

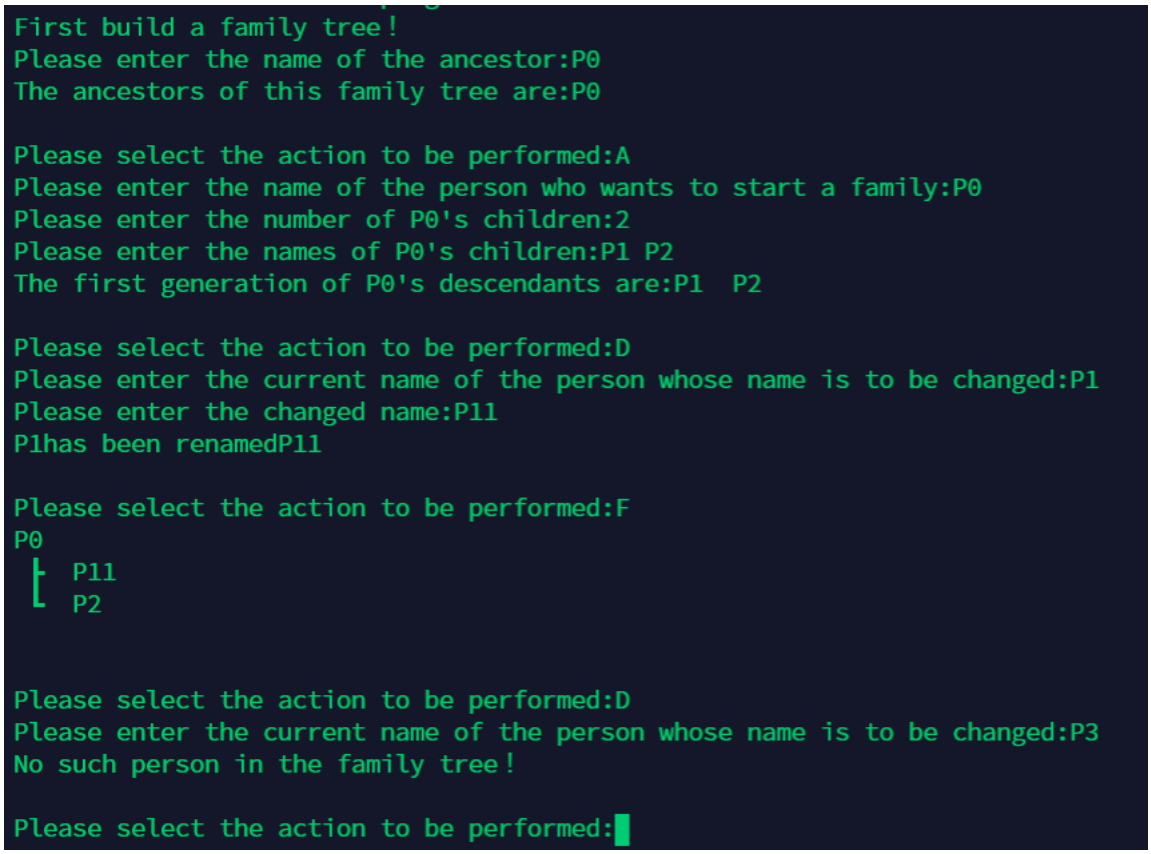
3.5.1 流程图



3.5.2 核心代码

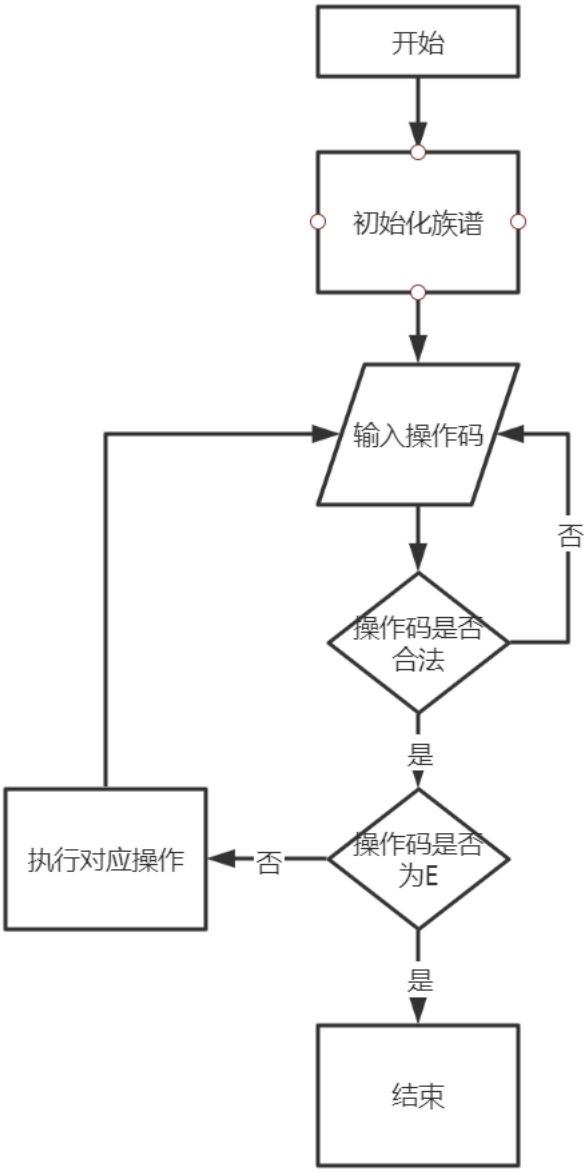
```
void tree::change() {
    std::cout << "Please enter the current name of the person whose name is to be changed:
";
    std::string str, name;
    std::cin >> str;
    while (!cin_check())
        std::cin >> str;
    node* r = find(root, str);
    if (!r) {
        std::cout << "No such person in the family tree！" << '\n' << '\n';
        return;
    }
    std::cout << "Please enter the changed name:";
    std::cin >> name;
    while (!cin_check())
        std::cin >> name;
    r->name = name;
    std::cout << str << "has been renamed" << name;
    std::cout << '\n' << '\n';
}
```

3.5.3 截屏示例


First build a family tree!
Please enter the name of the ancestor:P0
The ancestors of this family tree are:P0
Please select the action to be performed:A
Please enter the name of the person who wants to start a family:P0
Please enter the number of P0's children:2
Please enter the names of P0's children:P1 P2
The first generation of P0's descendants are:P1 P2
Please select the action to be performed:D
Please enter the current name of the person whose name is to be changed:P1
Please enter the changed name:P11
P1has been renamedP11
Please select the action to be performed:F
P0
├─ P11
└─ P2
Please select the action to be performed:D
Please enter the current name of the person whose name is to be changed:P3
No such person in the family tree!
Please select the action to be performed:

3.6 总体系统的实现

3.61 总体系统流程图



3.6.2 总体系统核心代码

```
void solve() {
    tree genealogy;
    std::string str[8];
    str[0]="**        Family Tree Management System              **";
    str[1]="==========================================================
======";
    str[2]="**        Please select the action to be performed    **";
    str[3]="**         A-perfect family                          **";
    str[4]="**         B---Add family members                      **";
    str[5]="**         C---Dissolve local families                **";
    str[6]="**         D---Change the names of family members     **";
    str[7]="**         E---Exit the program                       **";
    for(int i=0;i<8;i++)
        std::cout<<str[i]<<'\n';
    genealogy.init();
    char ch;
    while (true) {
        std::cout << "Please select the action to be performed:";
        std::cin >> ch;
        while (std::cin.fail()) {
            std::cin.clear();
            std::cin.ignore(INT_MAX, '\n');
            std::cout << "Input error,please re-input!" << '\n';
        }
        if (ch == 'E')
            break;
        switch (ch) {
        case 'A': {
            genealogy.complete();
            break;
        }
        case 'B': {
            genealogy.add();
            break;
        }
        case 'C': {
            genealogy.erase();
            break;
        }
        case 'D': {
            genealogy.change();
            break;
        }
        case 'F': {
```

```
            std::vector<int> v;
            // v.push_back(0);
            genealogy.print(genealogy.get_root(), 0, v);
            std::cout << '\n' << '\n';
            break;
        }
        default: {
            std::cout << "Input error,please re-input!" << '\n' << '\n';
            std::cin.clear();
            std::cin.ignore(INT_MAX, '\n');
        }
        }
    }
}
```

3.6.3 总体系统截屏示例

```
**         Family Tree Management System                **
============================================================
**         Please select the action to be performed     **
**          A-perfect family                            **
**          B---Add family members                      **
**          C---Dissolve local families                 **
**          D---Change the names of family members      **
**          E---Exit the program                        **
First build a family tree!
Please enter the name of the ancestor:P0
The ancestors of this family tree are:P0

Please select the action to be performed:A
Please enter the name of the person who wants to start a family:P0
Please enter the number of P0's children:2
Please enter the names of P0's children:P1 P2
The first generation of P0's descendants are:P1  P2

Please select the action to be performed:B
Please enter the name of the person to add a son (or daughter):P2
Please enter the name ofP2 the newly added son (or daughter):P21
The first generation of P2 descendants are:P21

Please select the action to be performed:F
P0
 ├─ P1
 └─ P2
    └─ P21


Please select the action to be performed:C
Please enter the name of the person in the family to be dissolved：P2
The person who wants to dissolve the family is:P2
The first generation of P2 descendants are:P21

Please select the action to be performed:PF
Input error,please re-input!

Please select the action to be performed:F
P0
 ├─ P1
 └─ P2


Please select the action to be performed:D
Please enter the current name of the person whose name is to be changed:P1
Please enter the changed name:P11
P1has been renamedP11

Please select the action to be performed:F
P0
 ├─ P11
 └─ P2


Please select the action to be performed:
```