Project 3

家谱管理系统

班级：教务三班

彭高15331253

刘俊良 15331215

刘钟涛 15331222

## 「题目要求」

## 「数据结构与算法」

我们采用树状结构来模拟这个家谱模拟系统。我们定义了Person类来表示家谱中每一个人，定义Family类来表示家谱中不同家庭之间的关系。Person表示Family中的一个节点，Family类是一个二叉树。

最后，我们定义一个FamilyAgenda类来抽象整个管理系统，所有的操作都要通过FamilyAgenda来实现。

图1 Person类、Family类URL

#### Person类

1. 成员selfName表示Person的名字
2. 成员itself表示Person的性别
3. 指针partner表示他的配偶（可看作是二叉树中的左指针）
4. 指针child表示第一个儿子，或者是兄弟（可看作是二叉树的右指针）

#### Family类

1. 成员founder表示家族的建立者（可看作是二叉树的根节点）
2. 成员 familyName表示家族名
3. 成员members\_total表示家族总人数
4. 方法marry\_other用于模拟家族男性成员结婚
5. 方法born\_baby用于模拟家族生孩子
6. 方法remove\_person用于模拟家族中女性成员出嫁后被移出族谱
7. 方法find\_person为寻找名字为name的家族成员

#### FamilyAgenda类

1. 成员familyMap 为一个存放在当前系统中家族的键值队

2. 方法queryFamilyByName模拟寻找某个家族中名为name的人

3. 方法creatFamily 模拟家族创建

4. 方法removeFamily 模拟家族移除（本管理系统）

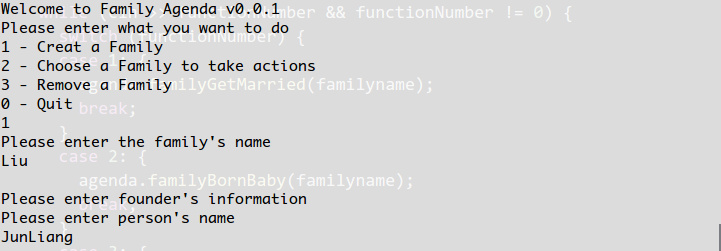
5. 方法familyBornBaby 模拟家族生孩子

#### 「数据测试、结果及分析」

### 家族创建

家族创建主要分为两个步骤，一个是Family对象的构造，一个是founder，即一个Person的构造。在这里我们把Person构造独立于Family的构造，为Person专门写了一部分交互，所以会出现“Please enter person’s name”的提示。

构造Person需要提供姓名和性别，这里默认家族创始人为男性，所以在代码这里构造Person的时候使用了一个默认参数。

图2 家族成功创建

构造Family需要提供founder和familyName。出于程序健壮性考虑，familyName输入之后会有一次重名判断。如果出现重名，则创建不成功，提示用户该家族已存在。

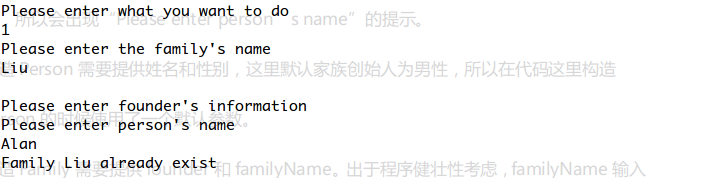


图3 家族创建失败

### 家族成员结婚

家族成员结婚需要提供男女双方信息。如果男方不存在，则提示用户，结婚失败；女方只能是新建对象，所以不会有判断。由于wife默认是女性，所以这里构造的时候也使用了默认构造参数。

同时，如果男方已经结婚，也是结婚失败，并提示用户不可重婚。

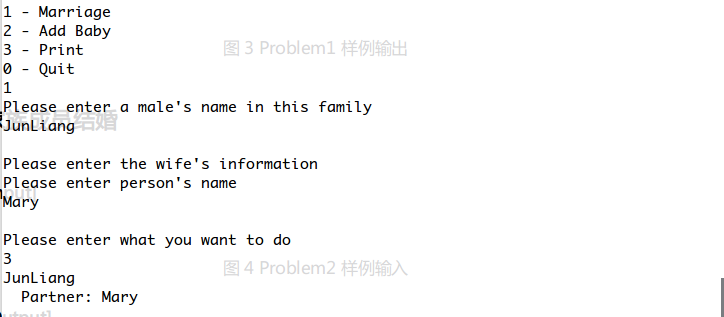


图4 结婚成功

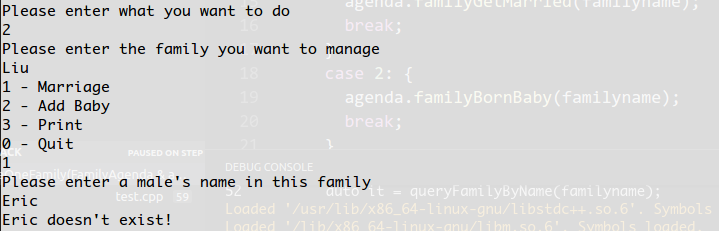


图5 结婚失败，男性不存在

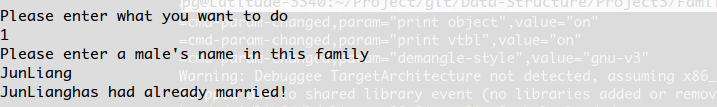
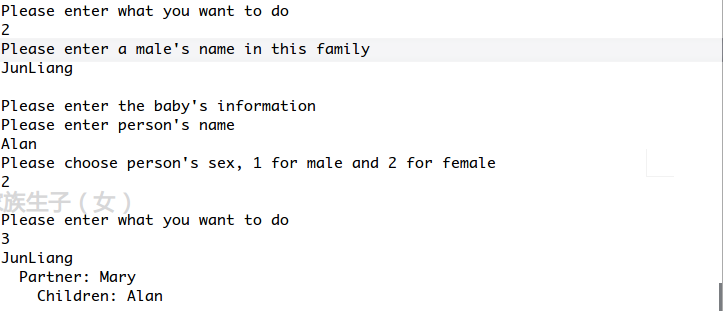


图6 结婚失败，不可重婚

### 家族生子（女）

家族中生子女只需要提供男方姓名。如果男方没有结婚则生子（女）失败，提示用户先结婚。生的孩子会在男方的右子树\*child（一个childChain）一直连下去。

图4 家族成员生子（女）

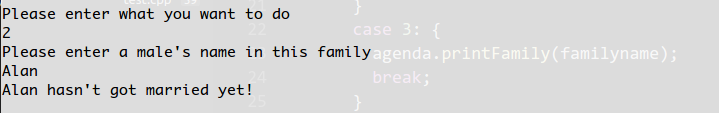


图4 生子女失败，男性未婚