**武汉大学计算机学院**

**本科生实验报告**

**《数据结构》课程设计**

**家谱信息管理系统**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：数据结构

课 程 教 师 ：文卫东

学生学号与姓名 ：2019302070035 刘涛榕

2019302080054 周熙宇

二○二○年十二月

**设　计　报　告　内　容　目　录**

1. 需求分析……………………………………………..….….….3
2. 项目设计……………………………………..…………….…..4
3. 测试报告…………………………….…………………….…...9

四、设计过程中遇到的问题及解决方法…………………….....…15

1. 尚未解决的问题以及应对的策略……………………….…..16

六、收获心得………………………………………………..…..….17

一、需求分析

1.1项目背景

家谱是人类生活中具有重要地位的内容，它记载着一个以血缘关系为主题的家族世代的绵延，记录着一个家族的成员以及血脉关系的载体，更是一个家族文化的延续的象征。家谱是中国特有的文化遗产，是中华民族的三大文献之一，属于珍贵的文化资料，对于历史学、民俗学、人口学、社会学和经济学的深入研究，都有着不同替代的独特作用。对于作为社会中各种家族的一份子的我们来说，家谱管理与我们的生活息息相关，因此我们需要将家谱管理与当代科技紧密的结合起来，以便于更好的将家谱文化保存下来以及发扬光大。

1.2 实现功能

（1）实现家庭成员信息存储：包括姓名，出生地，出生日期，死亡日期，性别，身高，职业等；

（2）家族关系存储：将各家庭成员之间的关系，存储在计算机中（可永久保存）；

（3）家谱数据的更新：修改、删除、加入；

（4）将家谱以较友好的格式输出（显示）；

（5）按基本信息查询成员，按亲戚关系查询；

（6）统计：平均寿命、平均身高、家庭平均人口等；

（7）屏幕显示家谱树形结构（类似Windows 目录）；

（8）支持鼠标操作。

二、项目设计

2.1总体设计

2.1.1设计思路

考虑到C++掌握了数据结构的相关知识但不会C++的Qt设计，但是python会Qt设计，而且有了C++的基础更易上手和学习，我们决定使用python语言实现家谱管理信息系统的基本功能，同时利用Qtdesigner和pyqt5来进行家谱管理系统的界面GUI设计与开发，利用Qt的信号与槽，将GUI界面和基本功能相结合，实现了家谱信息管理系统的成功运行。同时，我们将家谱信息存储在json文件中，相比于txt文件，json文件更易操作。

2.1.2 模块介绍

图2-1为我们的家谱管理系统的模块设计。ais用以保存家谱个人结点的基本信息以及相关函数操作。uis模块存放登录界面以及主页面的UI设计文件以及相应的信号功能。用familytreeapp.py文件将ais,uis中的主要功能进行封装。main.py文件里面进行familiytreeapp程序文件运行以及读写json文件功能。最后为了方便用户操作，我们创建bat文件进行程序的运行。

运用模块化的思想编写程序使得程序寻找bug维护显得较为轻松。

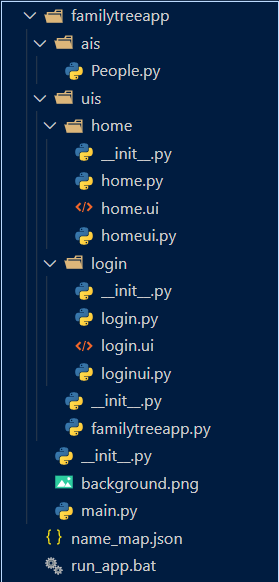


图2-1 项目模块图

2.1.3 UI介绍

考虑到家谱是中华传统文化宝库之一，本次系统UI设计采用中国传统水墨画为背景图，用书法作为字体，风格整洁统一。登录页面元素简洁，仅左侧竖排欢迎使用大字以及开始使用按钮，别无其他。



图2-2 登录界面UI

点击按钮进入主页面。菜单栏与分页区界限分明，易于用户识别重点内容进行操作分页区用户输入栏（lineEdit）、提示(label)、按钮(pushButton)排布间距适中，视觉体验舒适，主次分明。

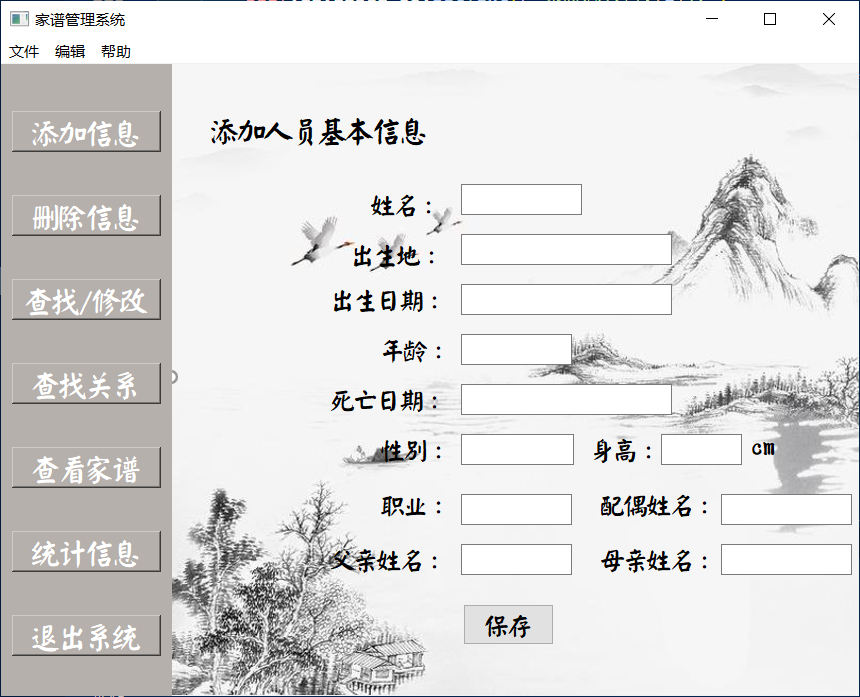


图2-3 主界面UI

2.2程序流程图

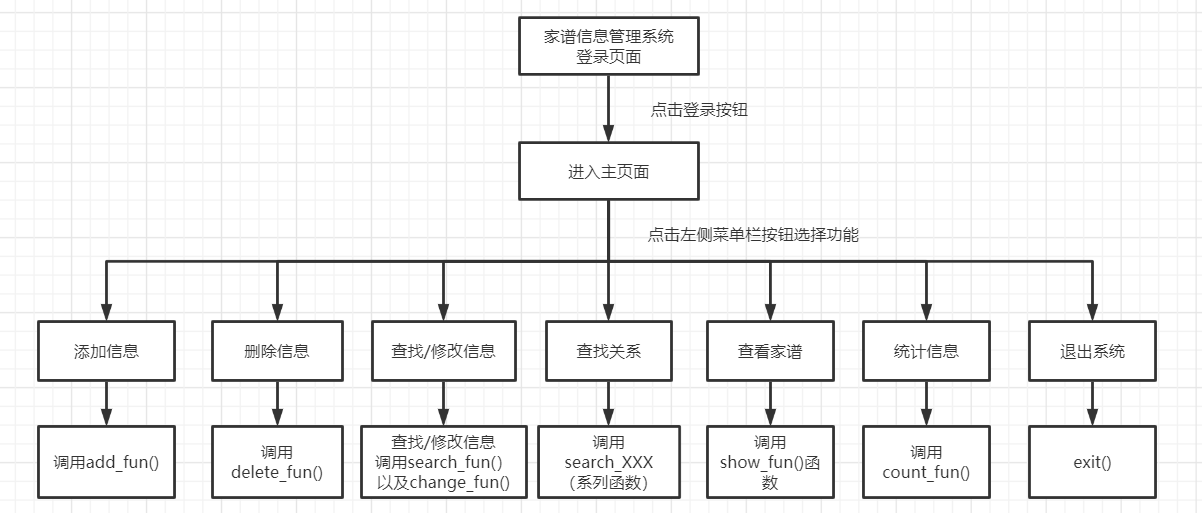


图2-4 程序流程图

2.3算法分析

在这次的程序设计中，我们小组大量地使用了循环、判断以及递归等结构。其中，循环和判断主要应用在判断主要应用在涉及到用户进行选择的功能实现上，递归用应用在成员的删除以及成员的显示功能上。

在众多具体功能的算法实现上，需要特别说明的主要是**和删除以及显示相关的函数算法。**

在删除函数delete中，传入参数“name”，delete函数会判断相应的结点有无孩子结点，如果有则对孩子列表中的每个“child\_name”调用delete函数。然后delete函数将会删除结点，并且判断该节点有无父母节点，如果有将会修改其孩子列表中的信息，确保在查看其父母结点具体信息是不会出现纰漏。删除函数如2-5所示。

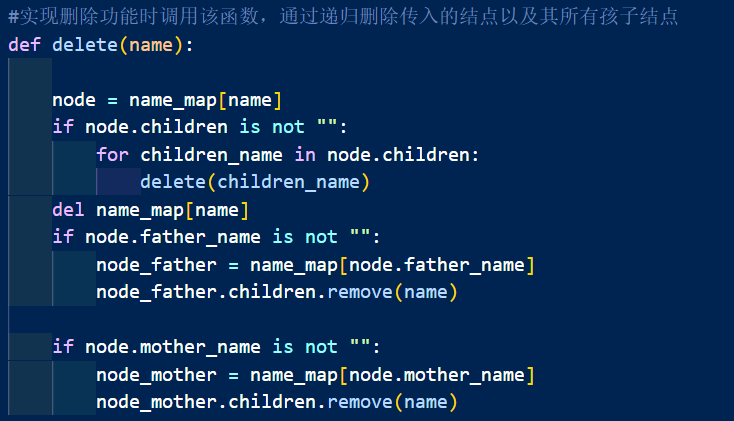


图2-5 删除函数展示

为了实现“将家谱以类似于Windows目录显示”，在和家族成员结点的显示相关的show\_fun和get\_map函数中，我们利用了pyqt中的QTreeWidget，为每个子孙结点建立一个与其姓名向对应的QTreeWidget对象，并将其上级目录设置为其父辈对应的对象。get\_map函数利用递归结构实现了主要的逻辑功能，该函数需要传递(node,root)两个参数，分别对应当前的成员结点以及对应的QTreeWidget对象。对于当前结点孩子列表中的每一个孩子，如果该孩子结点的孩子列表不为空，则为每个孩子结点创建一个QTreeWidget对象，其上级目录设为当前结点对应的QTReeWidget对象，然后再次调用get\_map函数；如果该孩子结点的孩子列表为空，则执行相同的步骤，但不在调用get\_map函数。get\_map（node,root）的具体算法如图2-6所示

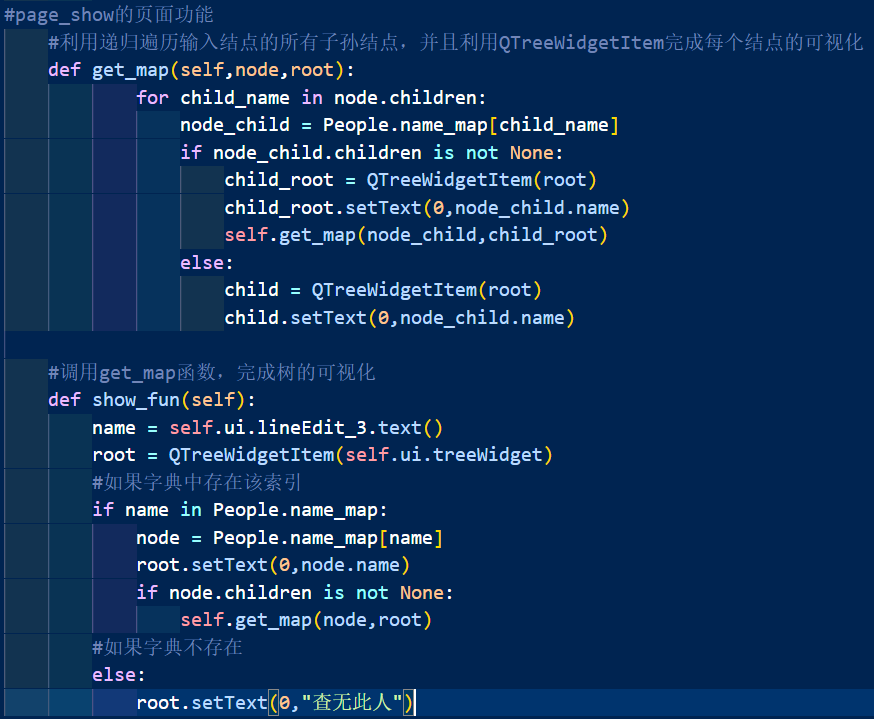


图2-6 get\_map和show\_fun的函数展示

2.4使用的数据结构

2.4.1 字典

在这次的大作业中，我们小组对整个家谱的存储和操作大多都是利用字典来完成的。字典是 Python 提供的一种常用的数据结构，它用于存放具有映射关系的数据。我们在People.py模块中定义了person类用于表示一个具体家庭成员，并且存储其所有的相关信息。对于每个家庭成员，我们将其具体的person类对象设置为字典name\_map中的value数据组；将该节点的姓名设置为key数据组。因此具体的每个成员结点就可以通过其姓名来进行访问。利用字典的该性质，我们实现了查找等诸多功能。通过字典中key到value一一对应的性质，我们完成了基本的亲缘关系的建立。例如为结点node添加母亲结点，只需要将node.mother\_name设置为其母亲的名字，即可通过字典name\_map访问其母亲结点。根据该性质我们完成了根据不同的亲戚关系查找结点的功能。

2.4.2 列表

考虑到实际情况，一对夫妻可能有多个孩子，因此我们放弃了最多设置两个孩子的方案，将孩子这一结点的属性设置为列表数据结构。由于列表是动态的，因此在添加或是删除孩子的时候非常方便，但是在具体处理的时候需要和结点其他的属性有所区别。例如在输入一个成员的名字，根据亲戚关系查找其所有孩子结点的信息时，就需要对列表额外使用for循环进行遍历，在寻找结点的兄弟姐妹时也需要用到类似的操作。

三、测试报告（以截图为主）

3.1实现从登录页面的跳转

操作：点击开始使用按钮，程序从登录界面跳转到主界面



图3-1 登陆界面



图3-2 跳转到主界面

3.2 实现添加家庭成员信息

操作：点击“添加信息”按钮进入界面，根据提示，输入添加的成员的信息（不填写的项默认为空），点击保存即可将该结点加入。



图3-3 添加家庭成员

3.3 实现信息的查找/修改

操作：点击“查找/修改”按钮进入界面，输入查找者姓名，如果没有这个人则会显示“查无此人”，如果有则会显示此人信息。选择需要修改的信息，点击“保存”进行信息修改。



图3-4修改张三年龄



图3-5 修改后

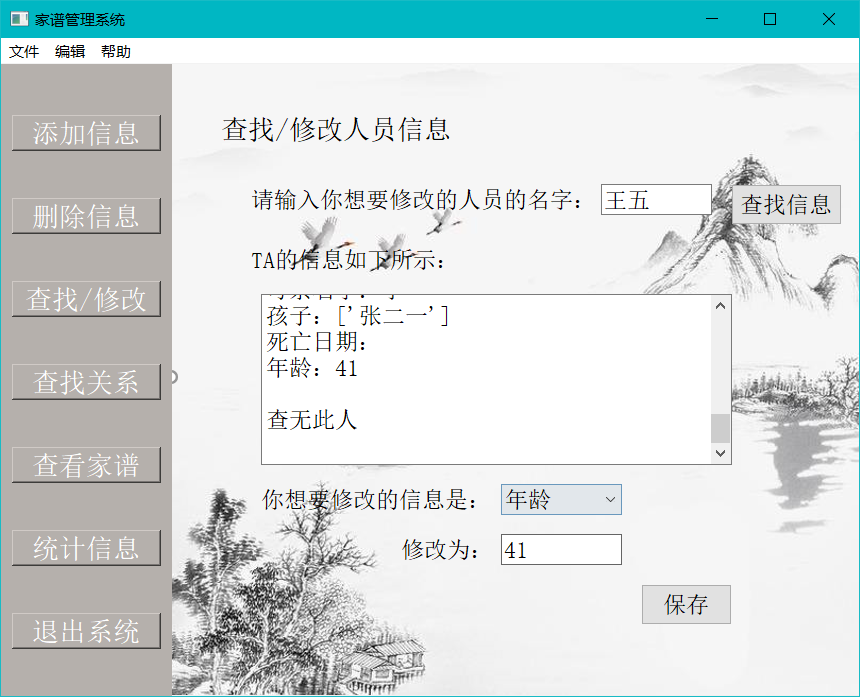


图3-6 查找王五显示“查无此人”

3.4 查找关系

操作：点击“查找关系”按钮进入界面，输入姓名，选择关系，即可看到目标的信息。如果搜索不到该节点，则显示“未录入xx信息”。

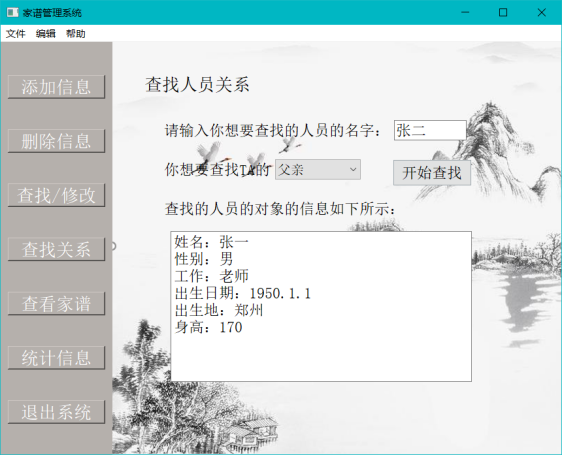


图3-7 查找张二父亲



图3-8 无法找到张二配偶

3.5 查看家谱

操作：点击“查看家谱”按钮进入界面，输入选择的祖先姓名，即可查看家谱。如果此人不存在则会显示“查无此人”。



图3-9 首先输入不存在的成员，然后查看张一家谱

3.6 统计信息

操作：点击“统计信息”按钮进入界面，即可看到平均的身高、年龄以及家庭总人数。



图3-10 显示统计信息

3.7 删除信息

操作：点击“删除信息”按钮进入界面，输入目标姓名，点击删除，完成操作。如果此人不存在，则显示“查无此人”。

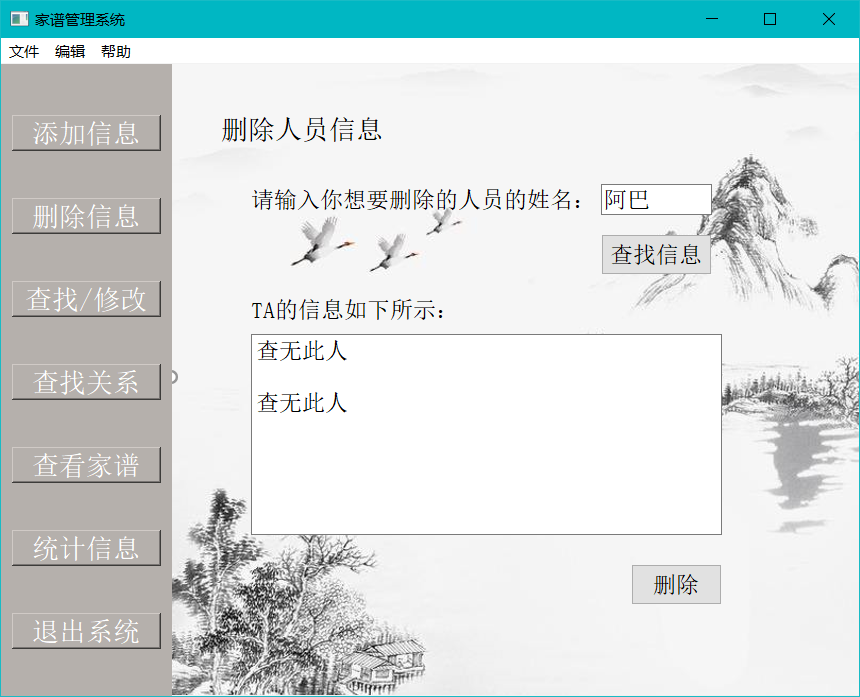


图3-11 试图删除不存在的人



图3-12删除“小张二一”

3.8 退出系统

操作：点击“退出系统”直接退出系统。

四、设计过程中遇到的问题及解决方法

4.1 UI设计遇到的问题

虽然在此前的实训课上已使用过Qtdesigner来进行GUI界面的设计，但是此次大作业要求多个功能的实现以及利用Windows目录的树状结构来显示家谱，这个操作确实很有难度。我们查询了CSDN博客上相关博文对于软件的介绍，最后采用了stackWidget类来进行主页面内分页面的跳转操作，利用treeWidget来显示家谱，这样符合Windows目录的树状结构。

在确定使用了合适的控件之后，如何使用控件的基本功能又成了一个问题。我们查询了Qt文档各个部分的函数，进行多次操作与实验成功解决了treeWidget如何显示家谱结构，各个页面按钮的信号跳转以及前后端的绑定。

附文档的链接：<https://doc.qt.io/qt-5/qtgui-module.html>

4.2家谱的显示

我们遇到的一个比较困难的问题就是如何显示家谱。由于我们使用的字典作为数据结构，因此相比之下不如和家谱本身更加接近的树那样方便展示。后来在pyqt中QTreeWidget模块的启发下，我们发现可以将通过结点之间的前后代关系关系，通过设置上层结点的方式，将字典中的结点以类似于数的形式展示出来。在具体的实现中，我们利用了递归结构，结合QTreeWidget模块，将祖先结点的所有子孙进行了遍历，并设置了其上层结点，完成了类似于WIndows目录的家谱显示。如图4-1所示。

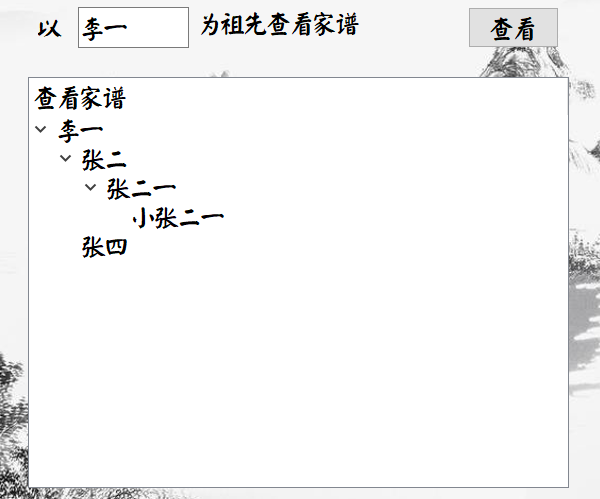


图4-1 最后的树状结构显示家谱

4.3 文件读取

由于我们采用的是json文件来存储家谱信息，所以如何利用python来读取json文件和将数据保存到json文件又是一个问题。我们同样查阅了相关博文以及询问了学长和同学们该怎么操作，最后解决了这个问题。

五、尚未解决的问题及考虑应对的策略

5.1 姓名的修改

在对整个程序进行测试的后期，我们在调试修改信息这一功能时，发现了和结点属性“name”的修改相关的一些待解决的问题。由于在这次的程序中，我们采用字典作为主要的数据结构，并且将name作为key数据组来进行结点的访问以及亲缘关系的建立，因此对于结点来说，name和其他的属性是有很大不同的。当用户需要修改一个结点的姓名，或者是其父母和配偶的姓名是，函数直接修改该结点相应的属性。但由于name还有索引的功能，而该索引实在结点被创建时生成的，直接修改姓名并没有修改索引，因此在直接或者间接查询用户的信息会出现一些问题。考虑到问题发现的比较晚，同时又涉及到个人、父母、配偶、孩子等多个关系的修改，因此我们不得不让修改功能局限在了“年龄、出生地、出生日期、死亡日期、身高、职业”这几项中。对此，我们提出的应对策略是，当用户尝试去修改和结点姓名相关的信息时，修改该节点在name\_map中的索引，同时判断该节点是否具有父母或者配偶、孩子等，然后利用函数来对具体的关系进行修改。希望今后我们能够将该功能进行完善。

5.2 UI的优化

相关组件操作需要完善。对于textBrowser控件和treeWidget控件在点击按钮，显示新的信息内容后，旧页面显示的信息无法清除和刷新。未来考虑将结点内容索引信息和显示的信息进行相关操作，能够进行页面刷新，让使用体验更完善。

用户体验要完善。比如在点击保存后应该也要弹出一个页面提醒用户保存成功，不然只靠终端的数字提示用户很难察觉操作成功。

5.3 菜单栏的优化

我们在最上方设有主菜单栏，包含文件、编辑与帮助。其中编辑菜单栏处的四个操作按钮都能跳转到相应的页面。我们希望未来能在文件中实现另存为，打开等功能（或许不是txt文件就没有这个必要？），在帮助里加入用户的使用说明手册

六、收获和心得

刘涛榕：

数据结构大作业可以说是我真正意义上完成的第一份大作业，经过了刷夜，无数的调参，查全英文的qt文档，递归，树的遍历与查找，仿佛都历历在目。对我个人而言，代码能力有了一定程度的提升。

大作业让我对python语言的运用有了更深的了解。比如python中字典的使用，键值对的遍历等，再比如python的Qt库的使用，Qtdesigner软件中相关控件的使用，可以说经历了实训和大作业，在某种意义上我成为了pyqt大师（误）。深夜和搭档一起debug，查找控件的用法会在我的大学生涯留下深刻的一笔。

其次，大作业让我对数据结构的相关知识进行了一个复习与巩固，比如字典的遍历，树的递归搜索等。我们在设计treeWidget时要展现Windows树状目录结构图，相应的get\_map函数设计了好久才终于实现，其中递归错误的经验值得吸取。

同时，本次大作业提升了我个人设计软件时的模块思维设计，将一个程序可以分为界面层、管理层与业务层，三个层面相结合调用，使得一个软件能够很好的运行，并且后期维护成本也不会太高。

周熙宇：

我觉得这次大作业是一次完全依靠我和小组搭档已有知识的运用以及未掌握知识的学习来实现的，因此我觉得还是很有成就感的。在这次作业中，我主要负责数据结构的逻辑实现，在编写函数的功能时，我对于之前在数据机构课上学到的递归等知识进行了回顾和应用，对这些知识点也有了更加深入的理解。除此之外，在和搭档一起思考如何实现界面和代码的对接的过程中，我也向他学习到了很多有关于pyqt5图形化界面的知识，了解到了Qt中许多组件以及一些其相关的操作函数，并且掌握了其应用。

在能力方面，我觉得自己的自学能力有了一定的提高，能够通过自学解决一些实际问题。

在心态方面，由于这次我们采用的时字典，因此出现了许多意想不到的bug。尽管多次被弄到非常崩溃，但是想要好好实现功能的决心还是使我们两个一次次修改，最终也收获了成就感。我觉得在今后会的工程中，我也应该秉持最终不屈不挠的心态，努力实现自己想达到的功能。