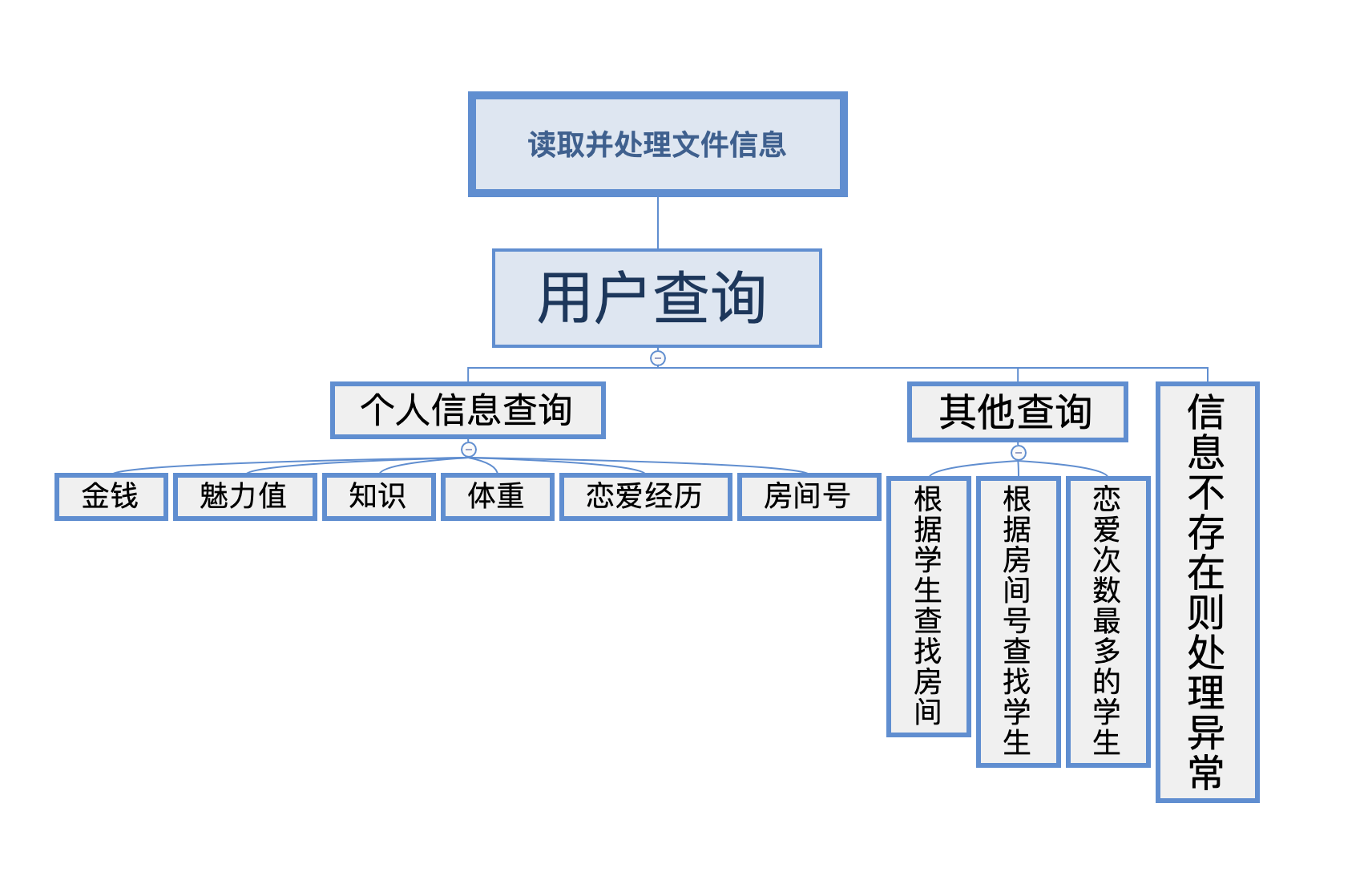
**课程项目设计总结**

1. **系统功能模板图**

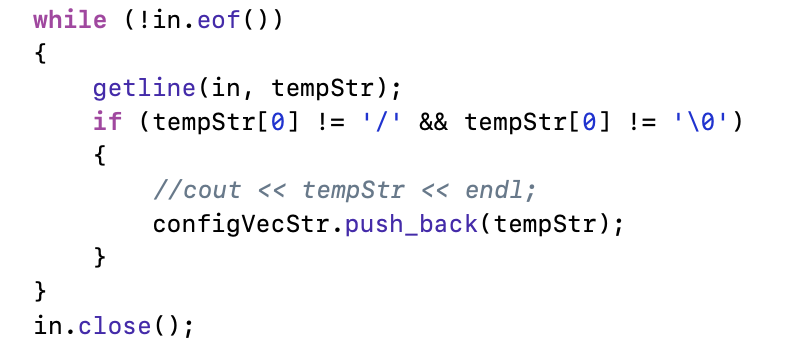


1. **程序实现思路**

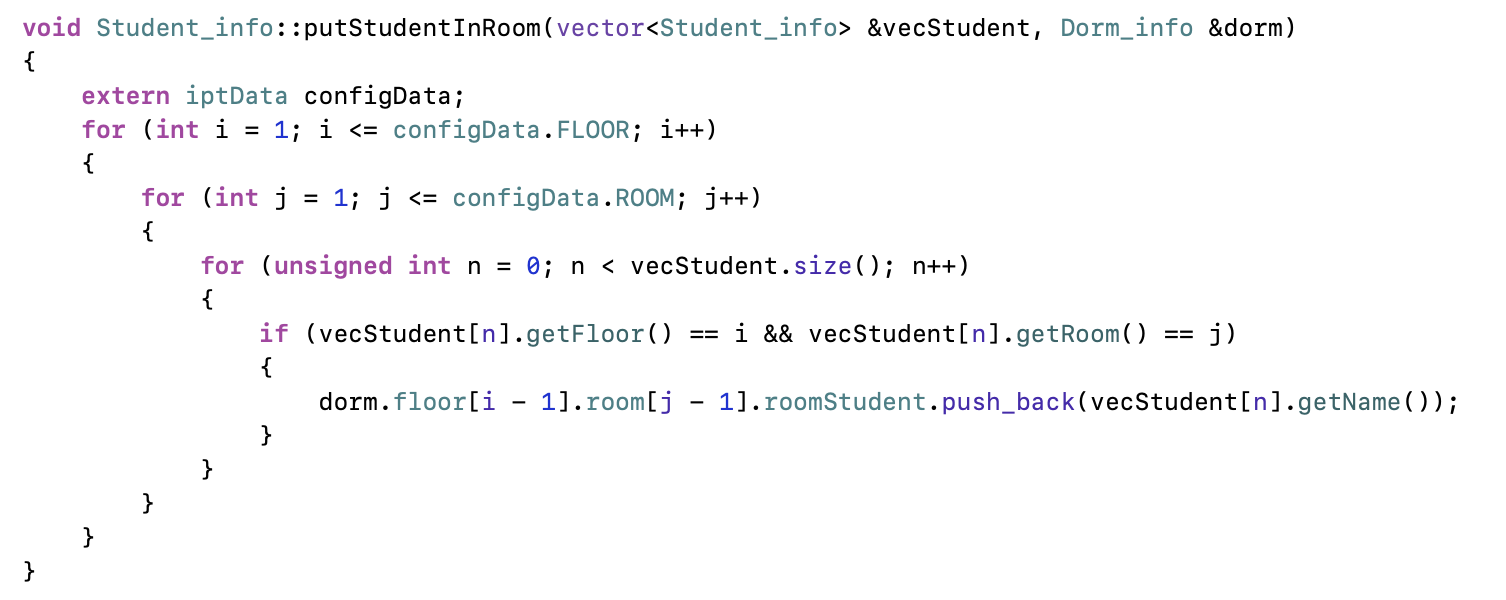
在第一部分的系统功能模板图中，已经大概展现出了这个学生养成系统的基本功能，程序的基本实现思路也如系统功能模板图中的步骤差不多。

\*\*\*\* 以下代码截图仅为功能实现的一部分代码 \*\*\*\*

1. 首先是读取文件信息，因为是 txt 文件，没有固定的格式，所以就需要根据里面给出的有效信息进行分割提取，无用的注释信息则忽略。



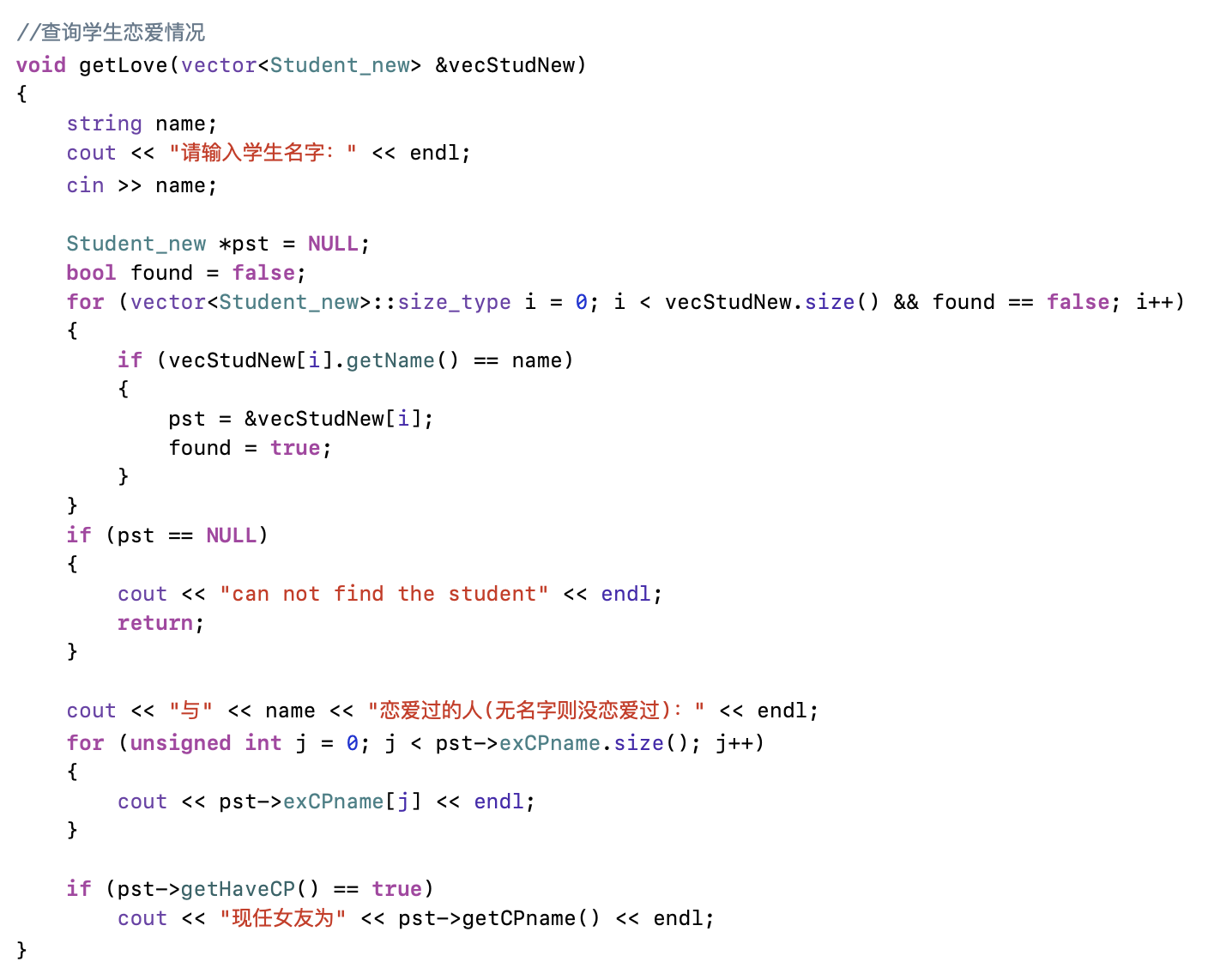
1. 信息提取后进行信息处理，，根据提取的信息里面的每一列所含的意义进行分别处理，例如把学生放进房间，学生之间恋爱，学生行为等等。





1. 学生的各种信息进行处理并存储后，就开始实现学生信息查询功能（search.cpp），这个查询分很多种，具体查询功能在系统功能模板图都有了展示，关键在于设计用户查询的函数能否复用重用，因为有些查询功能是比较类似的，因此如何抽象设计就成为了减少代码量的关键。





1. **设计说明**

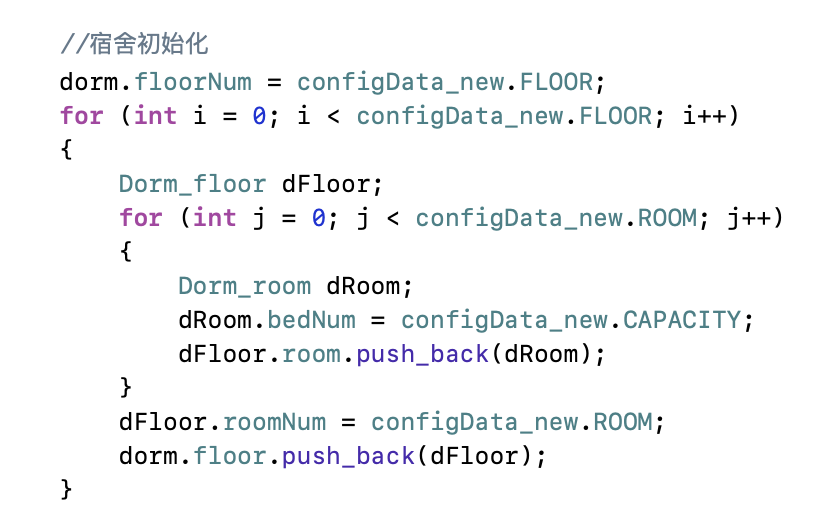
主要设计思路在第二部分的程序实现思路已有大概的说明，这个学生养成系统关键在于信息的提取处理以及查询函数的实现。

\*\*\* 以下代码截图仅为部分功能实现截图 \*\*\*\*

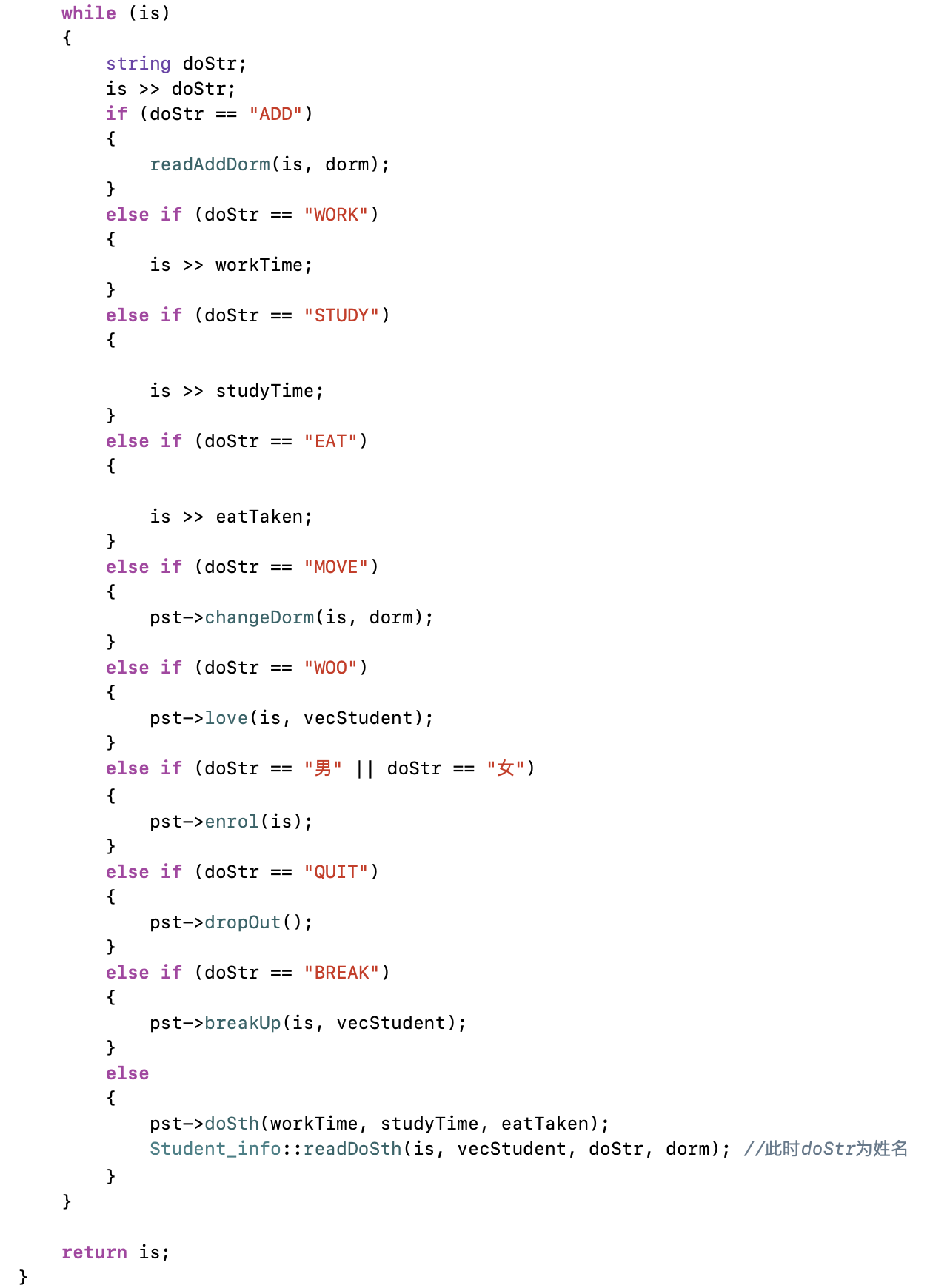
1. 提取处理信息的设计

程序首先对 config.txt里面的参数以及公式进行设计，对各项数据进行初始化，例如宿舍楼的初始层数以及初始房间数等等，以及对学生吃饭、学习、打工活动后的状态变化公式进行处理。





然后再对instruction.txt以及 dispatch.txt文件处理，把里面的每一项数据进行分割然后进行各种操作处理，以对应 confi.txt里面的一些公式数据。



1. 对查询功能的设计

在这个学生养成系统中，对信息的各种查询占了很大的比重，所以如何优雅地实现全部查询功能也是程序的关键，因此在这个项目中，我独立创建了一个专用于查询的.cpp源文件，因为需要查询的功能比较多，所以用 switch 语句来实现根据输入来查询是比用 if 语句更为不错的设计，在每个 case 语句中，都分别对应着一种查询功能，每中查询都单独用一种函数来实现，因此只需要一个功能调用一个对应的函数即可。



1. **设计心得**

这个项目确实对于初学C++的人来说是一个比较全面而有用的锻炼，首先是选题个人觉得很不错，贴近学生的实际生活，能引起我们学生的兴趣，然后就是里面需要用到的知识基本涵盖了C++基础知识的一大部分，能够真正起到锻炼的作用，最后就是项目难度从易到难，需求变更逐渐从简单到复杂，也让我们的思维逐步完善，对C++的体系构建也越来越齐全。从基础知识到知识进阶，这个项目都让我有了足够的了解，而且在完善项目的过程中，也用到了很多课本外的知识，我也通过网上查找相关知识以支撑完成项目，这让我学到了更多的知识，也让我学到了养成如何解决问题的习惯。

在项目的开始阶段，基本是想到什么就写什么，不会仔细去思考如何更有效率更优雅，后来深入写下去的时候，才发现以前写的错综复杂，难以维持下去，比如几乎把全部代码都直接写在了 main 函数里面。后来实在写得太乱，才决定重构了，把所有功能都单独封装在函数里面，而且主要运用了单一职责原则以及开闭原则等基本设计原则，同时使用了几种设计模式，例如单例模式、工厂模式以及外观模式等，最后还分为.cpp文件以及.hpp头文件，每个文件都负责相关的程序功能，互不干扰，又能互相调用，但即使是重构过一次了，还是有很多的不满意之处，即使是发现了问题所在但依然很难纠正过来，比如代码重复率太高了，很多相同的代码在程序内多处出现，这也反映了我对整个项目的大局观不够好，对负责一个项目的整体还是不够熟悉。

在这个项目中，我虽然能够实现需求中的大部分功能，但对C++的知识还是不够熟悉，很多都是需要通过网络搜索相关知识才能完成，基本是边查边调试边写，这有利也有弊，利在于让我可以不用很熟悉某种语言就可以通过已有知识和网络知识来迅速构建一种解决方案，这让我培养了一种先有解决方案后写代码的思维；弊在于知识构建不系统，编程效率低下。

同样地，这个项目也遇到很多很多的困难和问题，首先是编码问题，因为我一直习惯在 Linux 下面编程，在对文件进行信息提取的编码方面遇到了很大的问题，尝试了多种办法依然无法完美解决，依然有部分中文在程序输出中是显示乱码的，最后逼不得已开了个虚拟机用 Windows 系统，用上了 visual studio 才解决了编码问题，中文才得以正确输出；再者就是 debug 问题，刚开始对 vs的操作确实不熟悉，对调试也确实进行得很艰难，单步跟踪断点调试也还是很熟悉，对调试条件的设置也不是很好，这给编程带来了很大的麻烦，刚开始基本都是通过引入装饰器模式来打印 log 调试，后来才实现了两者共同实现，大大提高了编程效率。还有很多在C++知识方面的小错误更是让我体会到了深深的痛苦，但随后解决的成就感更是无与伦比，让我深深着迷，体会到了C++的魅力所在。

总之，通过这个课程项目，让我有了一次完整构建一个项目的经验，锻炼了如何在需求逐步变更中完善项目的思维，也体会了C++与其他编程语言的不同之处，感受到了它的魅力所在。做一个项目确实是最快熟悉一门编程语言的方法，但在C++只是体系中，我的认识还是很薄弱，如果想用C++真正做好一门能够商用的项目，还需要更多地看关于C++的书，逐步构建C++的思维体系，懂得利用 C++ 的长处所在。革命尚未成功，我仍需努力！

1. **关键功能代码**

在第二部分的程序设计思路以及第三部分的设计说明中，每一处都已经展示了不少的关键代码用以说明思路和设计，这里就不再多赘述。

