
华南农业大学数学与信息学院

综合性实验报告

课程：面向对象程序设计实验

实验题目：Java 源代码注释关键字
分析程序

学号：202014210104

姓名：古卓恒

专业：网络工程

班级：2020 级 1 班

指导教师：宋鸿陟

提交日期：2021 年 12 月 3 日

1. 题目介绍与分析

1.1 题目介绍

编写一个 Java 应用程序，对单个 Java 源程序文件及某个目录中的所有 Java 源程序文件（包括子目录）进行分析。

(1)Java 源程序文件个数，对目录分析进行分析时需要。

(2)Java 源程序中的字符个数，对目录分析时是其中所有源程序文件的字符个数总和。

(3)Java 源程序文件中的注释的个数，即源程序文件中共有多少个注释，包括：单行注释和多行注释。对目录分析时是其中所有源程序文件的总和。

(4)Java 源程序文件中的注释的字符个数，即源程序文件中所有注释的字符数之和。对目录分析时是其中所有源程序文件的总和。

(5)Java 源程序文件中关键字使用情况，即源程序文件各个关键字使用了多少次。对目录分析时是其中所有源程序文件的总和。

1.2 实验要求分析

(1) 程序运行时，首先要有一个主菜单。通过主菜单可以进入不同的子菜单进行操作或选择退出。

(2) 对于源程序文件（包括子目录）进行分析。

该程序首先提示并等待用户输入目标目录的路径。

1. 如果文件或目录不存在，提示用户该名称不是合法的目录名称，要求用户重新输入目录或退出当前界面或退出程序。

2. 如果文件或目录存在，则首先判断该路径是否为文件夹目录。

1)如果不是文件夹，则

2)如果是文件夹，

3. 分析完成后，提示用户“[目标路径文件] 分析完成.....”。

4. 最后将文件的信息输出到 data 的文件夹中，并保存到【文件名+.txt】文件中，详细的说明在（3）中说明。

(3) 对于将目录的详细信息输出到 data 文件夹中的详细功能分析。

由于将文件打包成 jar 包，而在 jar 包中不可以像读取文件夹的方式一样直接读取到 jar 包中的 data 文件夹。所以解决的方法是通过在运行 jar 包的目录下，寻找运行文件夹的目录下有无 data 目录，若无，则创建一个 data 文件夹，并把从源目录的分析结果保存到 data 文件夹下的【文件名+.txt】文件中去，同时，对于文件名也会进行特殊处理，将所有 .\ 替换成 _。以确保文件名不会与路径混淆和直观的看出分析文件位置。如果存在 data 文件夹，则直接获取 data 文件夹的实例化对象，将从源目录的分析结果保存到 data 文件夹下的【文件名+.txt】文件中去。如果已经存在同名的文件，则会更新文件的内容。

(4) 用户输入数字自动打开对应目录文件的分析结果。

自动打开文件，会因系统的不同而导致打开的方法不同。在此主要研究在 windows 系统下的打开方式。通过 `Runtime.getRuntime().exec("rundll32 url.dll FileProtocolHandler " + txt_name)`来打开 txt 文件，若无法打开，则会指引用户手动打开。

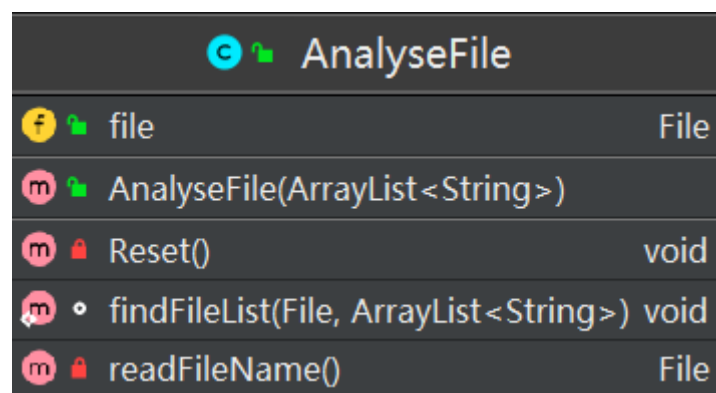
2. 程序静态结构

Main:程序主入口。在这里只有一行代码,即创建 Mainproject 实例化对象,以达到简洁代码的目的。

Mainproject:程序主任务。这里包含一些界面指令的处理和一些异常处理。

Interface:封装的程序可视化界面,及可视化界面各种操作方法的类。这个类中包含各种情况可能发生的状况而要弹出的界面,将所有界面和关于界面的操作集成到该类下,有几个优点:1. 使程序界面整体风格更加统一;2. 使代码管理更加方便;3. 其他类可更专注于其专业性的方面操作。

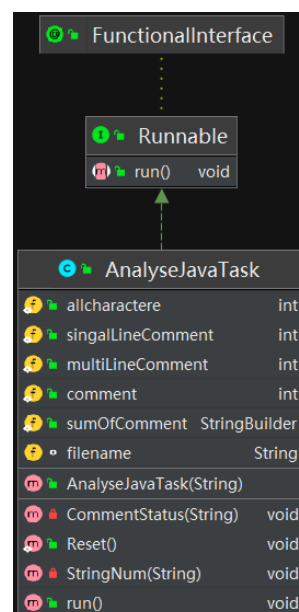
AnalyseFile:读目录的类。读入路径,若路径最后的后缀名为.java,则直接存入 list 中,否则,遍历目录,提取 java 文件到 list,若又发现子目录,则递归调用。最终得到的 list 列表记录了所有 Java 源文件绝对路径,最终得到所有 java 文件地址。



AnalyseJava: 对于 Analysefile 处理过后得到的 list 进行多线程操作,建立线程池。

TaskPool:封装了建立线程池操作的细节。

AnalyseJavaTask:对于单个 java 文件进行统计分析,得到该文件的字符个数、关键字使用情况、注释个数、注释字符数。

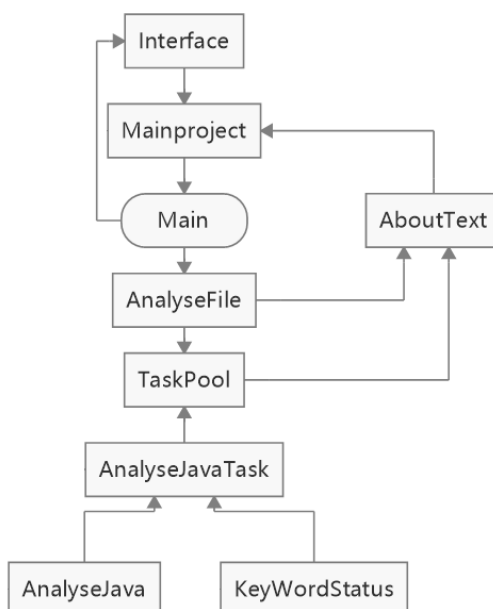


KeyWordStatus: 统计关键词使用情况。

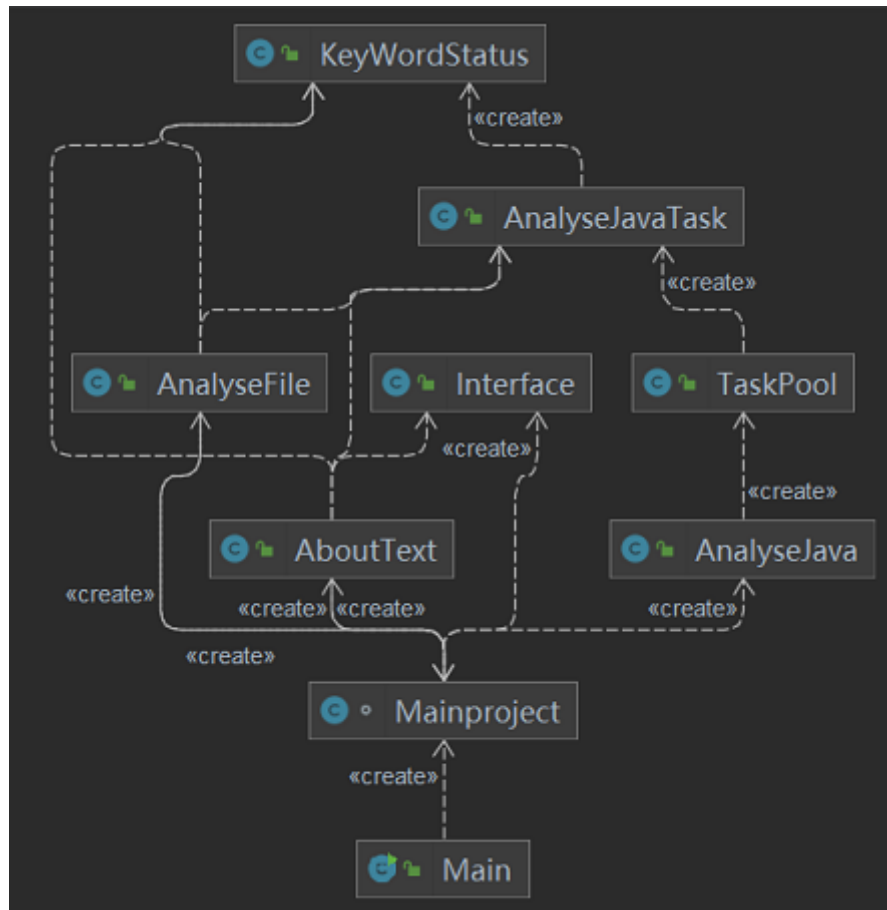
KeyWordStatus		
KEYWORDS		String[]
keywordsMap		Map<String, Integer>
KeyWordStatus(String)		
CleanJavaFile(String)		String
CountNumber()		void
InitializeMap()		Map<String, Integer>
Reset()		void
mapIntoListAndSort(Map<String, Integer>)		List<Entry<String, Integer>>
writelnToFile()		ArrayList<String>
content		String

AboutText:封装文件写入相关。

AboutText		
AboutText()		
AboutText(File, ArrayList<String>)		
OpenTxt(String)		int
ReadText()		void
SaveText(File, ArrayList<String>)		void



图表 1 初始规划类之间的关系

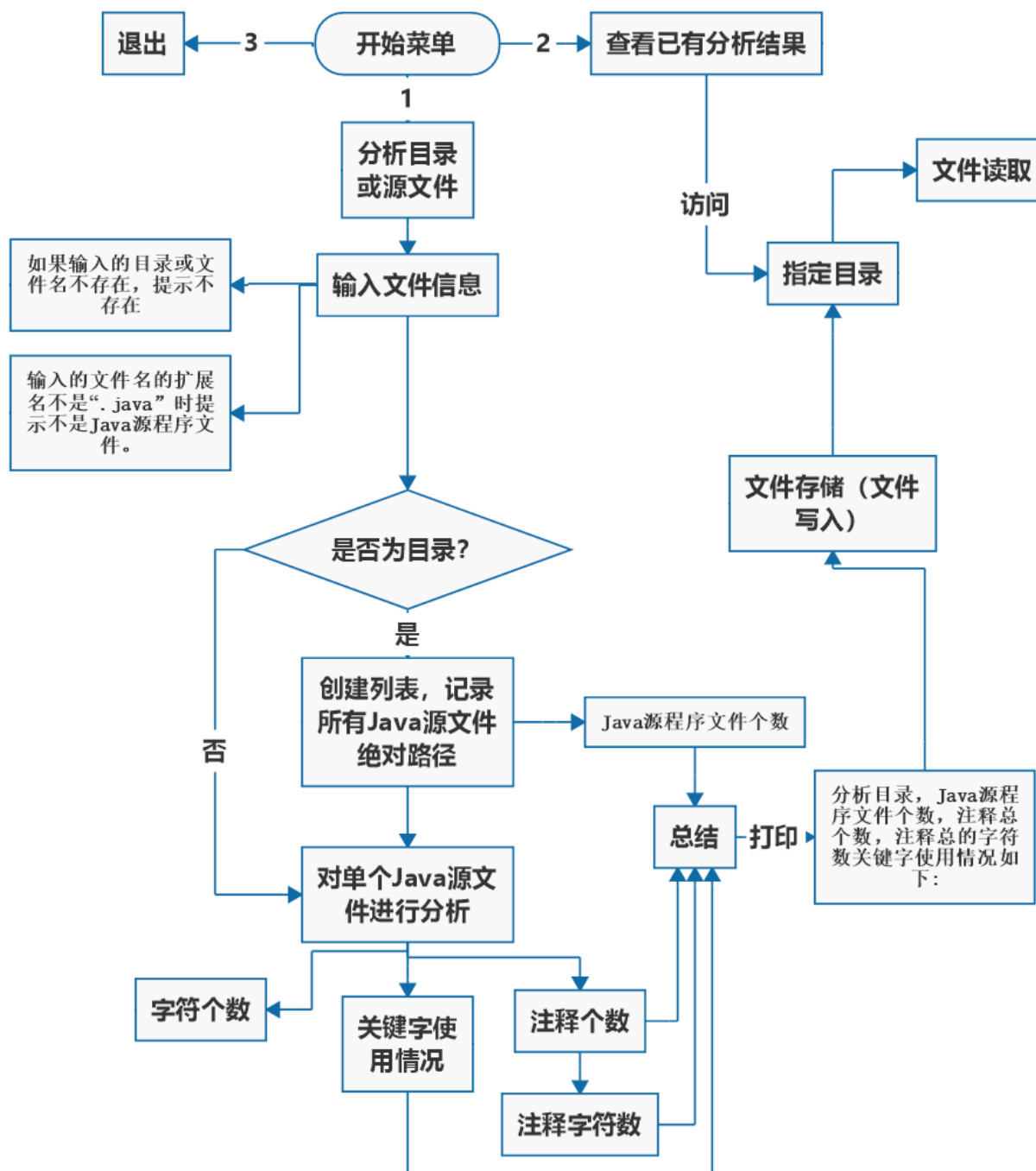


图表 2 具体实现

3. 程序主要功能

1. 分析目录或源文件程序。输入文件名后将分析结果存储到指定目录中。
 - 1) 在目录中筛选所有 Java 文件，并将绝对地址存入列表中。通过 equals 方法比较。
 - 2) 统计字符个数。逐行遍历，通过计算每行的长度可以得到总字符个数
 - 3) 注释个数。通过正则表达式
`Pattern.compile("((?!:)\s\s\s\s.*)|(\s\s\s\s*(\s\s|.)*?\s\s*\s\s)")` 和 `matcher.group()` 来判断是单行注释还是多行注释。
 - 4) 关键字使用情况统计。先用正则表达式除去所有注释、所有自字符串直接量，再逐个单词进行比对，将关键字使用情况存在字典里。
 - 5) 分析结果存储到指定目录中。通过字符串数组和 `BufferedWriter` 逐行写入。
2. 查看已有的分析结果。加载目录下结果文件名，通过数字选择打开。
 - 1) 加载目录。通过 `dir.list()` 存入文件名，并打印文件。

2) 打开文件。通过 `Runtime.getRuntime().exec("rundll32 url.dll FileProtocolHandler " + txt_name)` 打开。



程序整体流程图

4. 程序测试

注意：通过 cmd 运行 jar 包时，需要添加参数 `-Dfile.encoding=utf-8`

在 cmd 中输入命令行 `java -jar -Dfile.encoding=utf-8 Conservativeness.jar`

(1)以分析 `D:\IdeaProjects\javabaselearning\src` 为例，输入 1 后，再输入文件

目录的绝对路径，可以以极快的速度得到结果并存在 D_IdeaProjects_javabaselearning_src_result.txt 中，若检测到目录中已存在同名 txt 文件，则对该 txt 文件执行重新写入操作，以达到分析结果的更新。最后结果如图。

```
-----MENU-----  
1. 分析目录或源程序文件  
2. 查看已有的分析结果  
0. 退出  
-----  
请选择: 1  
请输入文件名(输入0退出程序,输入1返回上一层): |
```

文件分析结束，分析结果存放在文件[data\D_IdeaProjects_javabaselearning_src_result.txt]中
请输入文件名(输入0退出程序,输入1返回上一层):

此时，可选择继续分析或退到上一层。

若输入文件名错误或无法访问，则会要求重新输入。

```
-----MENU-----  
1. 分析目录或源程序文件  
2. 查看已有的分析结果  
0. 退出  
-----  
请选择: 1  
请输入文件名(输入0退出程序,输入1返回上一层): 000  
文件名错误，请重新输入！  
请输入文件名(输入0退出程序,输入1返回上一层):
```

(2) 以分析 F:\javaProjects\Conservativeness\src>AboutText.java 为例，结果类似(1)。

(3) 查看已有分析结果。在主界面输入 2，可查看已有分析结果。

```
-----MENU-----
1. 分析目录或源程序文件
2. 查看已有的分析结果
0. 退出
-----
请选择: 2
-----
1 -- D__IdeaProjects_javabaselearning_src_result.txt
2 -- F__javaProjects_Conservativeness_src_AboutText.java_result.txt
3 -- F__javaProjects_Conservativeness_src_Main.java_result.txt
4 -- F__javaProjects_Conservativeness_src_result.txt
-----
请输入要查看的文件编号:
```


输入想要查看的结果最前面的文件编号，即可自动打开该 txt 文件。

同时，输出框也会有跟进动作，提示文件已自动打开，同时输入 1 可返回上一层，输入 2 可继续查看其他文件，0 则可退出。

```
请输入要查看的文件编号: 2
文件: F__javaProjects_Conservativeness_src_AboutText.java_result.txt已打开
-----
1. 返回上一层
2. 查看其他文件
0. 退出
-----
请选择: |
```

输入数字后弹出对应 txt 文件

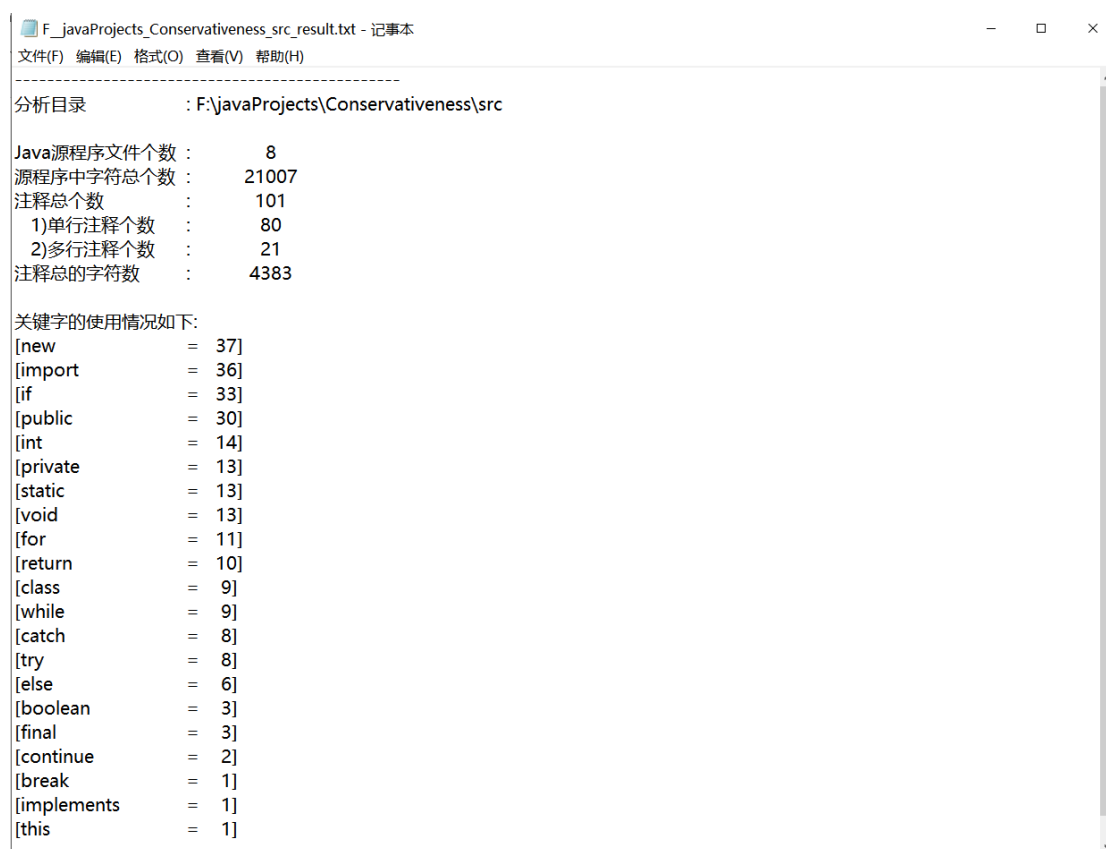
输入数字后自动打开，即弹出对应 txt 文件。



The screenshot shows a Notepad window titled "F:\javaProjects\Conservativeness_src>AboutText.java_result.txt - 记事本". The menu bar includes "文件(F)", "编辑(E)", "格式(O)", "查看(V)", and "帮助(H)". The text content is as follows:

```
-----  
分析文件      :F:\javaProjects\Conservativeness\src>AboutText.java  
  
源程序中字符总个数 :    6539  
注释总个数      :      7  
  1)单行注释个数   :      6  
  2)多行注释个数   :      1  
注释总的字符数   :    392  
关键字的使用情况如下:  
  
[if           = 11]  
[import       = 9]  
[new          = 9]  
[int          = 6]  
[for          = 5]  
[catch        = 4]  
[try          = 4]  
[else         = 3]  
[private      = 3]  
[public       = 3]  
[final        = 2]  
[void         = 2]  
[class        = 1]  
[continue     = 1]  
[return       = 1]  
[while        = 1]  
-----
```

分析单个 java 源程序



The screenshot shows a Notepad window titled "F:\javaProjects\Conservativeness_src_result.txt - 记事本". The menu bar includes "文件(F)", "编辑(E)", "格式(O)", "查看(V)", and "帮助(H)". The text content is as follows:

```
-----  
分析目录      : F:\javaProjects\Conservativeness\src  
  
Java源程序文件个数 :      8  
源程序中字符总个数 :   21007  
注释总个数      :    101  
  1)单行注释个数   :     80  
  2)多行注释个数   :     21  
注释总的字符数   :   4383  
关键字的使用情况如下:  
  
[new          = 37]  
[import       = 36]  
[if           = 33]  
[public       = 30]  
[int          = 14]  
[private      = 13]  
[static       = 13]  
[void         = 13]  
[for          = 11]  
[return       = 10]  
[class        = 9]  
[while        = 9]  
[catch        = 8]  
[try          = 8]  
[else         = 6]  
[boolean      = 3]  
[final        = 3]  
[continue     = 2]  
[break        = 1]  
[implements   = 1]  
[this         = 1]  
-----
```

分析目录

若输入错误数字，则会提示重新输入。

```
-----
1 -- D__IdeaProjects_javabaselearning_src_result.txt
2 -- F__javaProjects_Conservativeness_src_AboutText.java_result.txt
3 -- F__javaProjects_Conservativeness_src_Main.java_result.txt
4 -- F__javaProjects_Conservativeness_src_result.txt
-----

请输入要查看的文件编号: 5
数字输入错误, 请重新输入
```

5. 实验总结

在这一次综合性实验中, 我个人的 Java 技能得到了实际的应用, 对于 Java 的各类知识和概念有了更深层次的认识。多功能程序的设计、面对对象的思想要求我们有抽象观、全局观。除此之外, 为了满足实验要求, 我还自学了一些知识, 例如 Java 正则表达式、封装、聚合、组合、继承、IO 等知识。在这一过程中拓展了面对对象程序设计的知识图谱。

在写代码的过程中, 我有充分的认识, 就是每个类都是流水线上面的个体, 我要做的就是让每个类只操心自己的工作, 而不需要关注数据从哪里来、是什么数据结构, 我也非常清楚每个类的功能, 每个类需要去实现的具体的功能, 需要将数据处理到什么样的效果。

在程序测试部分, 发现 bug、解决 bug 的过程中, 需要细心和耐心, 还有查阅资料的能力, 在这些方面我都得到了经验和提升。

众所周知, 实验报告是一个实验成果的结果的集中展示和靓丽门面, 也是实验者总结自身、反思自我和提升自己的必由路径。对于产品来说, 再好的产品没有介绍说明书, 可能会导致使用门槛的增高; 对于一个程序软件来说, 没有必要的说明和介绍, 使用者也难以顺利的使用; 对于科研结果来说, 如果没有字面上的、图片上的解释和说明, 其价值也很难被发现。我一开始没有认识到这点, 我在 2021 年 11 月 10 几号完成代码部分后, 便去做了其他事而没有撰写实验报告, 导致最后实验报告过于简单和肤浅, 在实验报告中, 我没有完全展现出打代码过程中的一些理解和思考, 在最后甚至发现了一些 bug, 以至于我在 2021 年 12 月 3 日凌晨 1 点还在完善我的实验, 而且已经过了 deadline 几小时了。在此, 感谢学委的通融, 这也给我上了极为宝贵的一课: 程序员不仅要注重代码的质量, 还要注重文档的撰写, 这不仅帮自己厘清了自己的思路, 也帮他人可以更了解自己的产出。

在这一次综合性实验中, 我的逻辑思维能力、自学能力、分类管理能力、探究问题能力都有了提升。君子曰: 学不可以已”, 我也会通过不断的学习, 改进自己写代码的质量。完善整个设计的流程, 掌握面向对象开发程序的真正内涵。最后, 感谢老师的指导。