

**2019年春季学期  
计算机学院《软件构造》课程**

**Lab 4实验报告**

|  |  |
| --- | --- |
| 姓名 | 闫锋 |
| 学号 | 1170300614 |
| 班号 | 1703006 |
| 电子邮件 | [1502833622@qq.com](mailto:1502833622@qq.com) |
| 手机号码 | 18800410681 |

**目录**

[1 实验目标概述 1](#_Toc3922818)

[2 实验环境配置 1](#_Toc3922819)

[3 实验过程 1](#_Toc3922820)

[3.1 Error and Exception Handling 2](#_Toc3922821)

[3.2 Assertion and Defensive Programming 2](#_Toc3922822)

[3.2.1 checkRep()检查invariants 2](#_Toc3922823)

[3.2.2 Assertion保障pre-/post-condition 2](#_Toc3922824)

[3.3 Logging 2](#_Toc3922825)

[3.3.1 写日志 2](#_Toc3922826)

[3.3.2 日志查询 2](#_Toc3922827)

[3.4 Testing for Robustness and Correctness 2](#_Toc3922828)

[3.4.1 Testing strategy 2](#_Toc3922829)

[3.4.2 测试用例设计 2](#_Toc3922830)

[3.4.3 测试运行结果与EclEmma覆盖度报告 2](#_Toc3922831)

[3.5 SpotBugs tool 2](#_Toc3922832)

[3.6 Debugging 3](#_Toc3922833)

[3.6.1 理解待调试程序的代码思想 3](#_Toc3922834)

[3.6.2 发现并定位错误的过程 3](#_Toc3922835)

[3.6.3 如何修正错误 3](#_Toc3922836)

[3.6.4 结果 3](#_Toc3922837)

[4 实验进度记录 3](#_Toc3922838)

[5 实验过程中遇到的困难与解决途径 3](#_Toc3922839)

[6 实验过程中收获的经验、教训、感想 4](#_Toc3922840)

[6.1 实验过程中收获的经验和教训 4](#_Toc3922841)

[6.2 针对以下方面的感受 4](#_Toc3922842)

# 实验目标概述

# 本次实验重点训练学生面向健壮性和正确性的编程技能，利用错误和异常处 理、断言与防御式编程技术、日志/断点等调试技术、黑盒测试编程技术，使程序 可在不同的健壮性/正确性需求下能恰当的处理各种例外与错误情况，在出错后 可优雅的退出或继续执行，发现错误之后可有效的定位错误并做出修改。 实验针对 Lab 3 中写好的 ADT 代码和基于该 ADT 的三个应用的代码，使用 以下技术进行改造，提高其健壮性和正确性： ⚫ 错误处理 ⚫ 异常处理 ⚫ Assertion 和防御式编程 ⚫ 日志 ⚫ 调试技术 ⚫ 黑盒测试及代码覆盖度

# 实验环境配置

|  |  |
| --- | --- |
|  | 配置 |
| URL | <https://github.com/ComputerScienceHIT/Lab4-1170300614> |
| 系统 | Win10 |
| JDK版本 | 1.8.0\_16 |
| JAVA\_PATH | C:\Program Files (x86)\Common Files\Oracle\Java\javapath |

# 实验过程

请仔细对照实验手册，针对每一项任务，在下面各节中记录你的实验过程、阐述你的设计思路和问题求解思路，可辅之以示意图或关键源代码加以说明（但千万不要把你的源代码全部粘贴过来！）。

## Error and Exception Handling

1.FileException

文件的打开错误，以及各种存在错误。

2.FileFormatException

文件的类型还有格式错误。以及作为suer类申明别的异常。

3.EqualLabelException

重复标签异常

4.illegelinputexception

非法输入异常

5.noappexception

空app系统的异常即空输入的异常

6.orbitexception

轨道异常

7.sentenceexception

判定异常

8.timeexception

时间错误

9.trackexception

轨道异常

10.userageexception

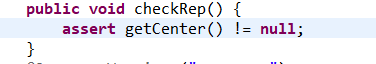
错误的用户以及时间异常

## Assertion and Defensive Programming

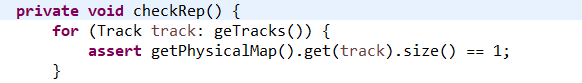
1. atomstructure

### checkRep()检查invariants

1. atomstructure



1. stellarsystem

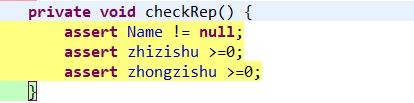


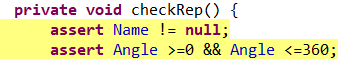
1. appsystem

C:\Users\Don YAN\AppData\Roaming\Tencent\Users\1502833622\TIM\WinTemp\RichOle\A%@)`CI(AC8ZYL@$U4ZGI}3.png

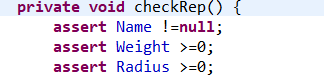
### Assertion保障pre-/post-condition

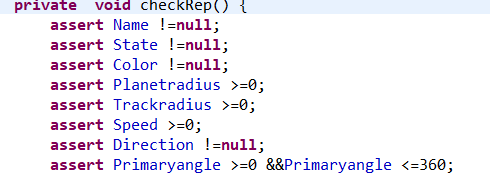
1. atomstructure





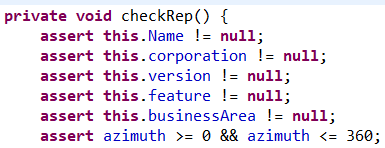
1. stellarsystem





1. app system

C:\Users\Don YAN\AppData\Roaming\Tencent\Users\1502833622\TIM\WinTemp\RichOle\017[UBQOJA9V$}P13MUKXLS.png

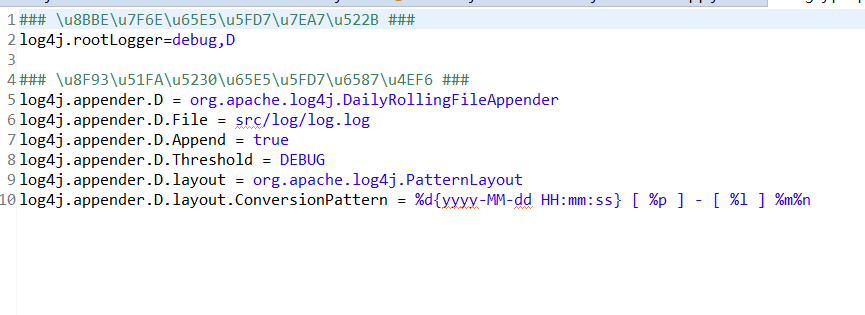


## Logging

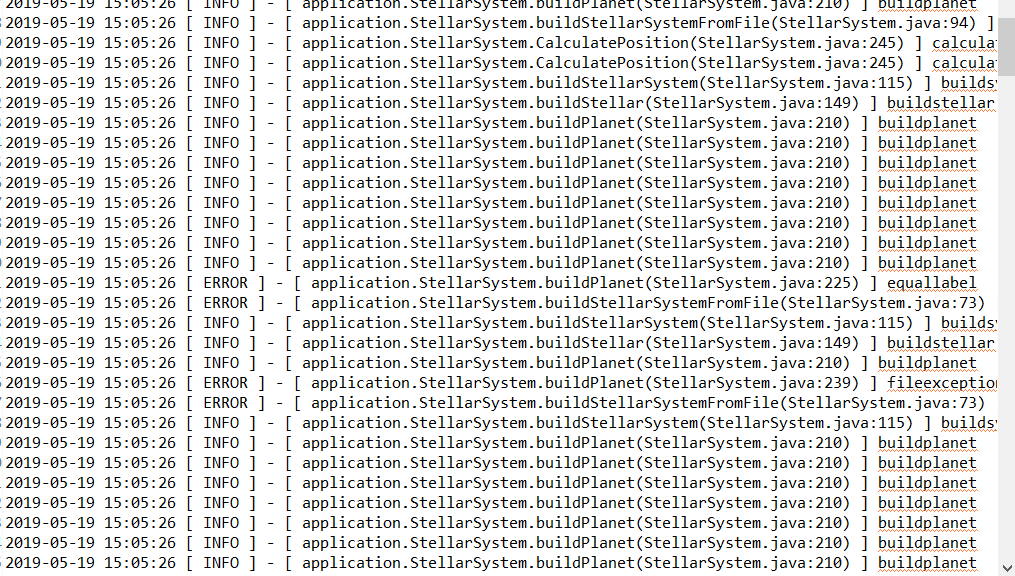
### 写日志

考虑到各种情况我使用log4j.

配置文件情况



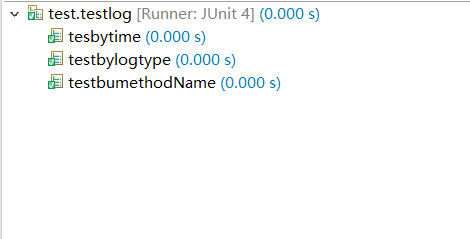
Log的记录情况



### 日志查询

考虑使用多种查询方式，我建立一个查询类。

分别实现了按时间段的查询还有按日志类型，和方法名称查询



## Testing for Robustness and Correctness

### Testing strategy

选择对底层method进行全面测试。实现多类型和多方面的测试。

### 测试用例设计

1. atom structure

我选择了多种异常情况。比如原子核错误还有电子跃迁错误。当然还有输入错误等等。

1. stellarsystem

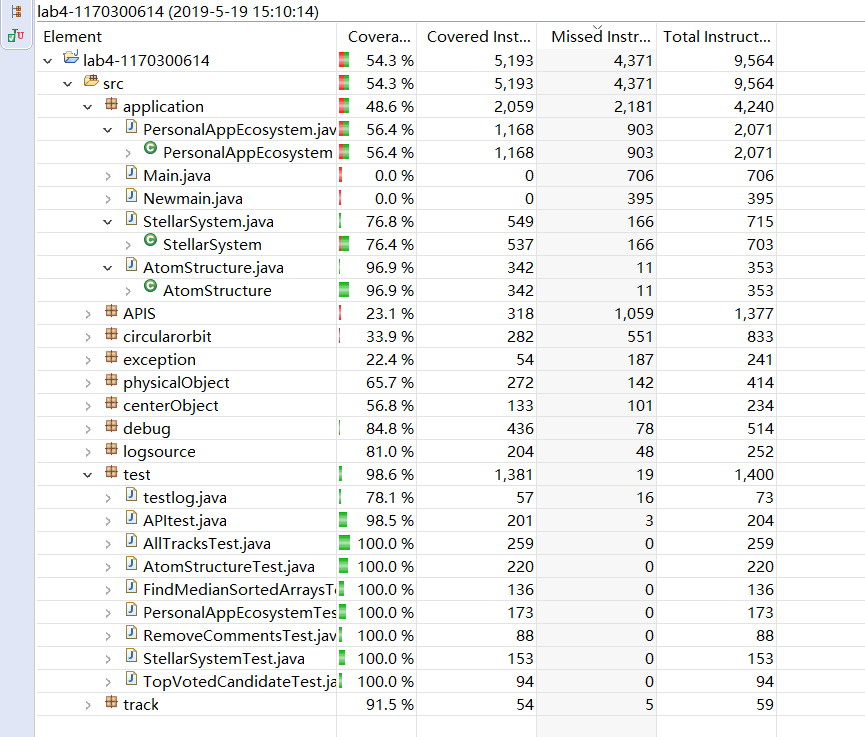
测试情况有缺少恒星，缺少行星。轨道的错误，相同行星的错误

3app system

测试等多的地方。例如用户的各种错误，还有app的使用的错误，还有各种的app的关系错误，文件读入的错误。安装卸载错误等等

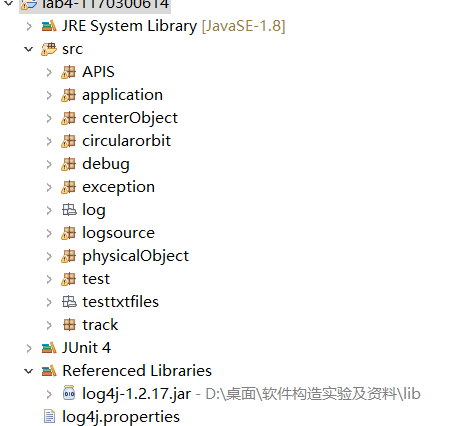
### 测试运行结果与EclEmma覆盖度报告

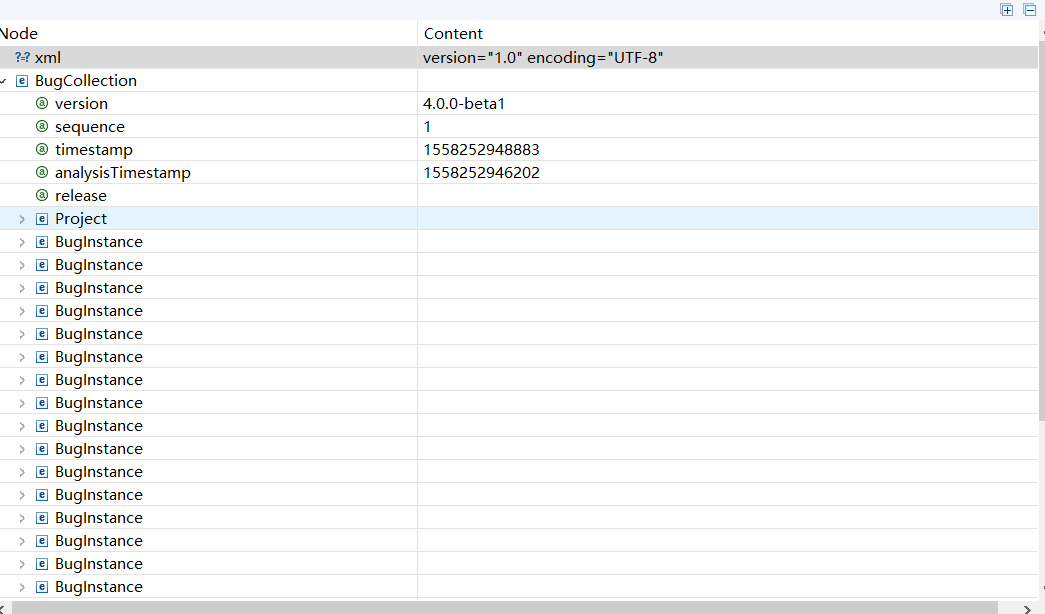
截图如下



## SpotBugs tool

使用了spotbug发现并且修改了所有的bug





## Debugging

### 理解待调试程序的代码思想

1FindMedianSortedArrays

求两个数组的中位数

2.RemoveComments

显然要求我们去除注释

3. TopVotedCandidate

求选举的人数

### 发现并定位错误的过程

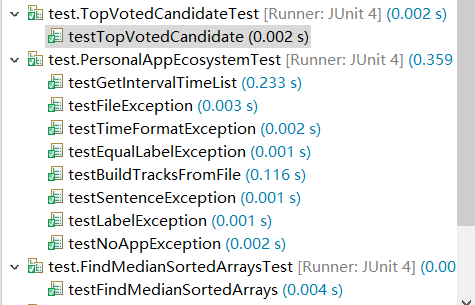
通过debug一步一步的理清程序的内容

### 如何修正错误

通过实验的要求还有debug结果来改正错误

### 结果

Test结果如下：



# 实验进度记录

请使用表格方式记录你的进度情况，以超过半小时的连续编程时间为一行。

每次结束编程时，请向该表格中增加一行。不要事后胡乱填写。

不要嫌烦，该表格可帮助你汇总你在每个任务上付出的时间和精力，发现自己不擅长的任务，后续有意识的弥补。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 计划任务 | 实际完成情况 |
| 2019.5，6-2019.5.8 | All day | 实现了3.6的debug部分 | 基本完成 |
| 2019.5.9-2019.5.13 | Free time | 实现3.1的大部分 | 实现 |
| 2019.5.16-2019.5.16 | All day | 实现3.2 | 基本实现 |
| 2019.5.17 | Na | 实现log工作 | 基本实现 |
| 2019.5.18 | All day | 实现了加强健壮性 | 基本实现 |
| 2019.5.18 | All day | 实现了robust的最后操作 | 实现 |
| 2019.5.19 | Free day | 实现spot bugs | 基本实现 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的难点 | 解决途径 |
| 关于防御性编程理解不够深入 | 深入学习防御性编程 |
| 使用log4j. | 学习log4j |
|  |  |

# 实验过程中收获的经验、教训、感想

## 实验过程中收获的经验和教训

## 针对以下方面的感受

1. 健壮性和正确性，二者对编程中程序员的思路有什么不同的影响？

答：健壮性比较适用于程序成熟和使用要求比较高；正确性是要求实现程序的基本操作

1. 为了应对1%可能出现的错误或异常，需要增加很多行的代码，这是否划算？（考虑这个反例：民航飞机上为何不安装降落伞？）

答：不划算

1. “让自己的程序能应对更多的异常情况”和“让客户端/程序的用户承担确保正确性的职责”，二者有什么差异？你在哪些编程场景下会考虑遵循前者、在哪些场景下考虑遵循后者？

答：一个是程序要有自己的防御性。一个是实现用户的自己操作。

1. 过分谨慎的“防御”（excessively defensive）真的有必要吗？如果你在完成Lab5的时候发现Lab5追求的是I/O大文件时的性能（时间/空间），你是否会回过头来修改你在Lab3和本实验里所做的各类defensive措施？如何在二者之间取得平衡？

答：有必要，

1. 通过调试发现并定位错误，你自己的编程经历中有总结出一些有效的方法吗？请分享之。Assertion和log技术是否会帮助你更有效的定位错误？

答：准确使用debug很重要

1. 怎么才是“充分的测试”？代码覆盖度100%是否就意味着100%充分的测试？

答：对底层的测试是很有意义的

1. Debug一个错误的程序，有乐趣吗？

答：是的

1. 关于本实验的工作量、难度、deadline。

答：可以接受

1. 到目前为止你对《软件构造》课程的评价和建议。

答：增加课时

1. 期末考试临近，你对占成绩60%的闭卷考试有什么期望或建议？//请严肃的提出，杜绝开玩笑，教师会认真考虑你们的建议。

答：希望能够增加一些习题课时