

**2018年春季学期  
计算机学院大二软件构造课程**

**Lab 4实验报告**

|  |  |
| --- | --- |
| 姓名 | 于沐潇 |
| 学号 | 1160301028 |
| 班号 | 1603010 |
| 电子邮件 | yumuxiao@163.com |
| 手机号码 | 13796016081 |

**目录**

[1 实验目标概述 1](#_Toc511890111)

[2 实验环境配置 1](#_Toc511890112)

[3 实验过程 1](#_Toc511890113)

[3.1 Error and Exception Handling 1](#_Toc511890114)

[3.1.1 针对输入文本文件的异常/错误处理 1](#_Toc511890115)

[3.1.2 针对输入图操作指令的异常/错误处理（可选） 1](#_Toc511890116)

[3.2 Assertion and Defensive Programming 2](#_Toc511890117)

[3.2.1 checkRep()检查invariants 2](#_Toc511890118)

[3.2.2 Assertion保障pre-/post-condition 2](#_Toc511890119)

[3.3 Logging 2](#_Toc511890120)

[3.3.1 写日志 2](#_Toc511890121)

[3.3.2 日志查询 2](#_Toc511890122)

[3.4 Testing for Robustness and Correctness 2](#_Toc511890123)

[3.4.1 Testing strategy 2](#_Toc511890124)

[3.4.2 测试用例设计 2](#_Toc511890125)

[3.4.3 测试运行结果与覆盖度报告 2](#_Toc511890126)

[3.5 FindBugs tool（可选） 2](#_Toc511890127)

[3.6 Debugging 3](#_Toc511890128)

[3.6.1 待调试程序 3](#_Toc511890129)

[3.6.2 理解待调试程序的过程 3](#_Toc511890130)

[3.6.3 发现并定位错误的过程 3](#_Toc511890131)

[3.6.4 如何修正错误 3](#_Toc511890132)

[3.6.5 结果 3](#_Toc511890133)

[4 实验进度记录 3](#_Toc511890134)

[5 实验过程中遇到的困难与解决途径 3](#_Toc511890135)

[6 实验过程中收获的经验、教训、感想 3](#_Toc511890136)

# 实验目标概述

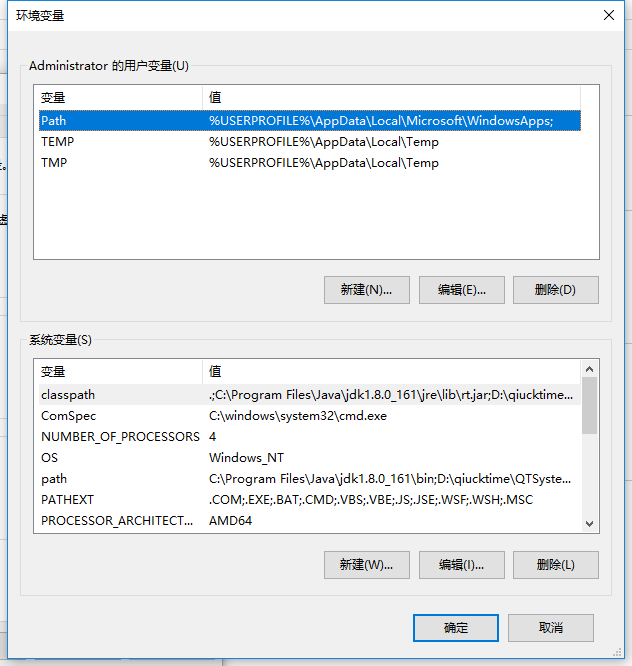
本次实验重点训练学生面向健壮性和正确性的编程技能，利用错误和异常处理、断言与防御式编程技术、日志/断点等调试技术、黑盒测试编程技术，使程序可在不同的健壮性/正确性需求下能恰当的处理各种例外与错误情况，在出错后可优雅的退出或继续执行，发现错误之后可有效的定位错误并做出修改。

实验针对 Lab 3 中写好的 ADT（Graph<L,E>）代码和基于该 ADT 的四个应用（GraphPoet、社交网络、计算机网络拓扑、电影网络）的代码，使用以下技术进行改造，提高其健壮性和正确性：

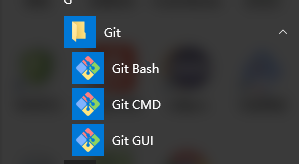
* + 错误处理
  + 异常处理
  + Assertion 和防御式编程
  + 日志
  + 调试技术
  + 黑盒测试及代码覆盖度

# 实验环境配置

Java是很早以前配置好的



Git从网上下载的



在这里给出你的GitHub Lab3仓库的URL地址（Lab3-学号）。在这里给出你的GitHub Lab4仓库的URL地址（Lab4-学号）。

https://github.com/ComputerScienceHIT/Lab4-1160301028.git

# 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

### 针对输入文本文件的异常/错误处理

覆盖实验手册3.1节(1)中列出的各项任务。

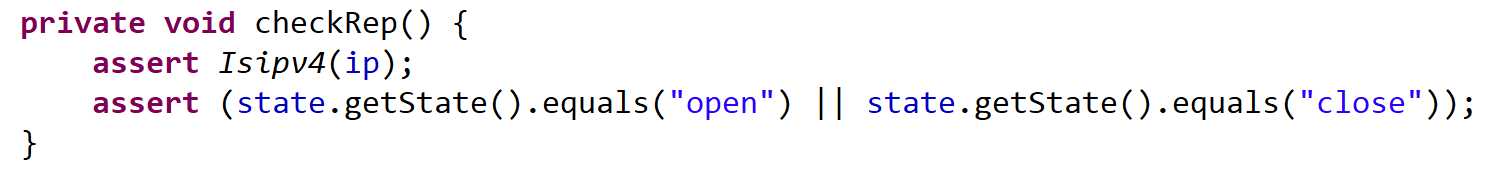
### 针对输入图操作指令的异常/错误处理（可选）

覆盖实验手册3.1节中(2)列出的各项任务。

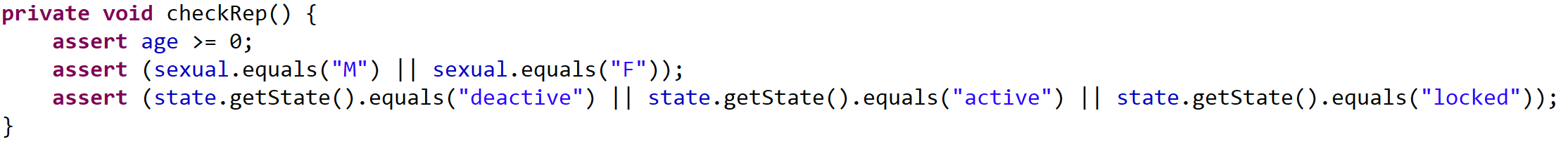
## Assertion and Defensive Programming

### checkRep()检查invariants

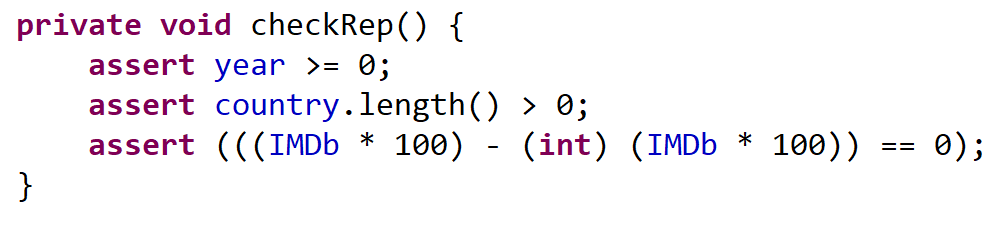
Computer的checkRep(), Router/WirelessRouter/Server的checkRep与Computer相似



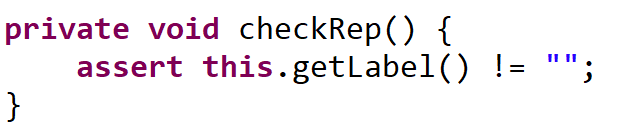
Person的checkRep, Actor/Director的与其类似



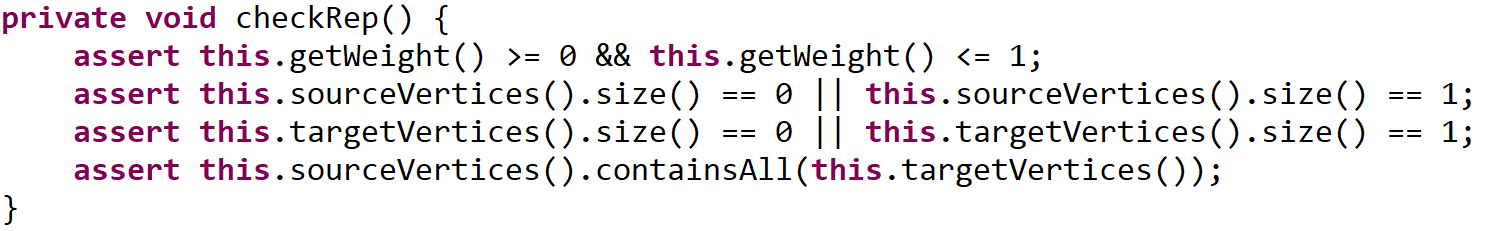
Movie的checkRep



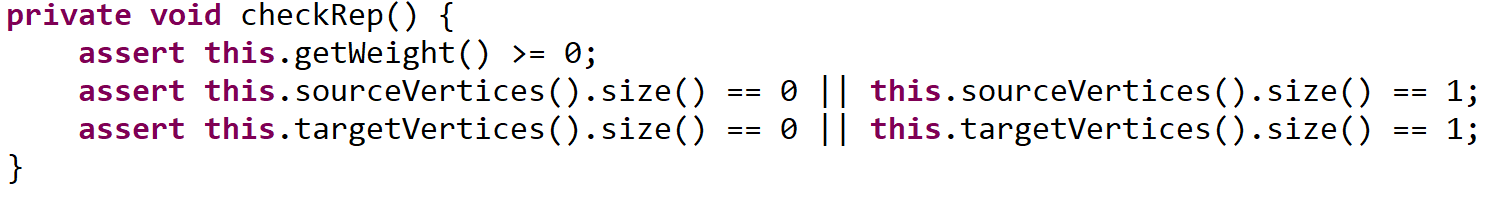
Word的checkRep



FirendConnection的checkRep

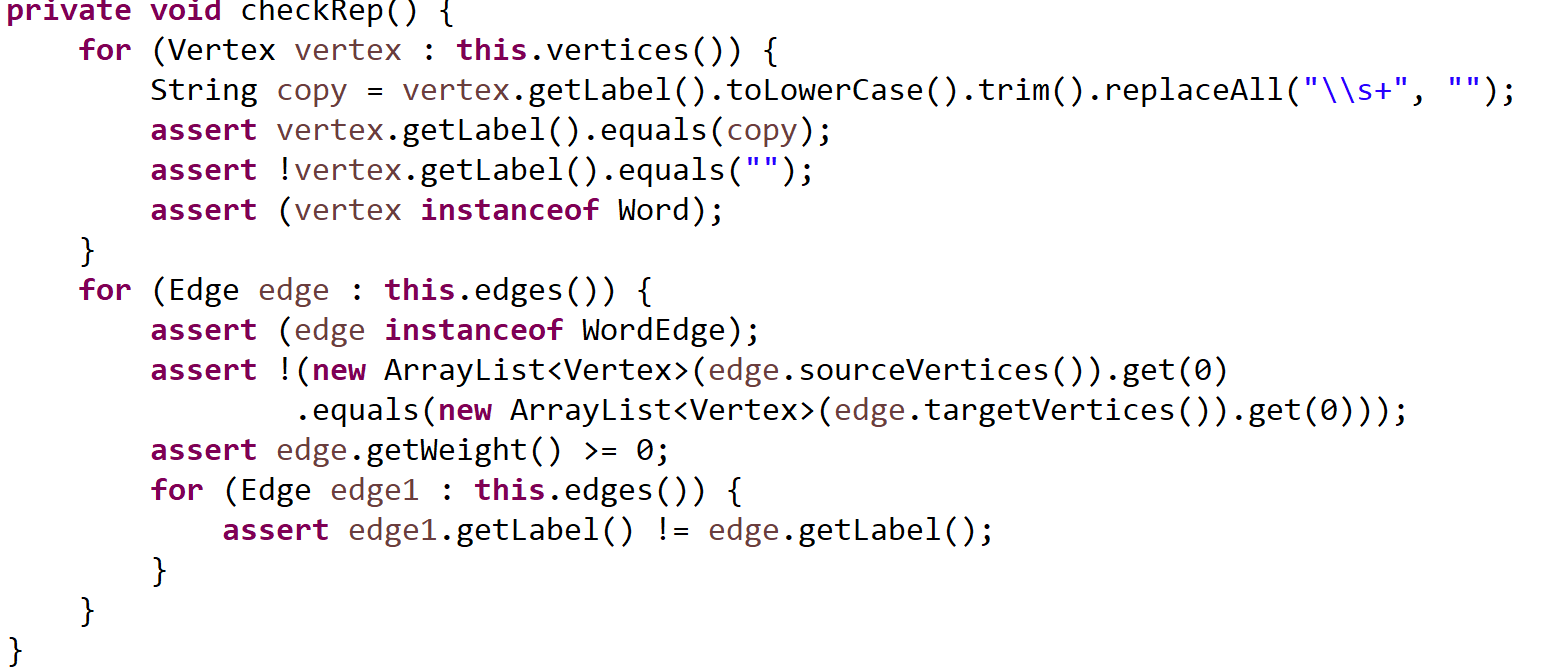


WordEdge的checkRep

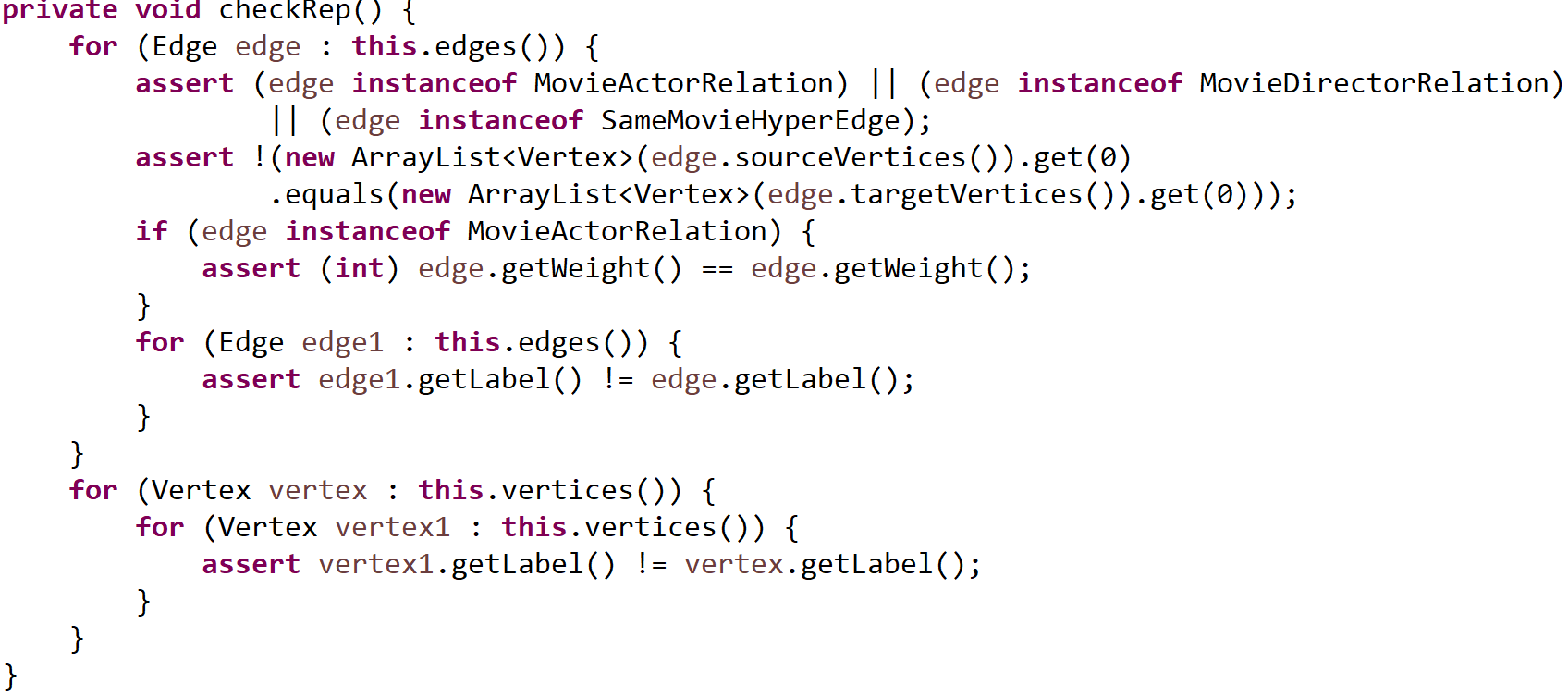


其他表示边的类的checkRep与上面两个相似

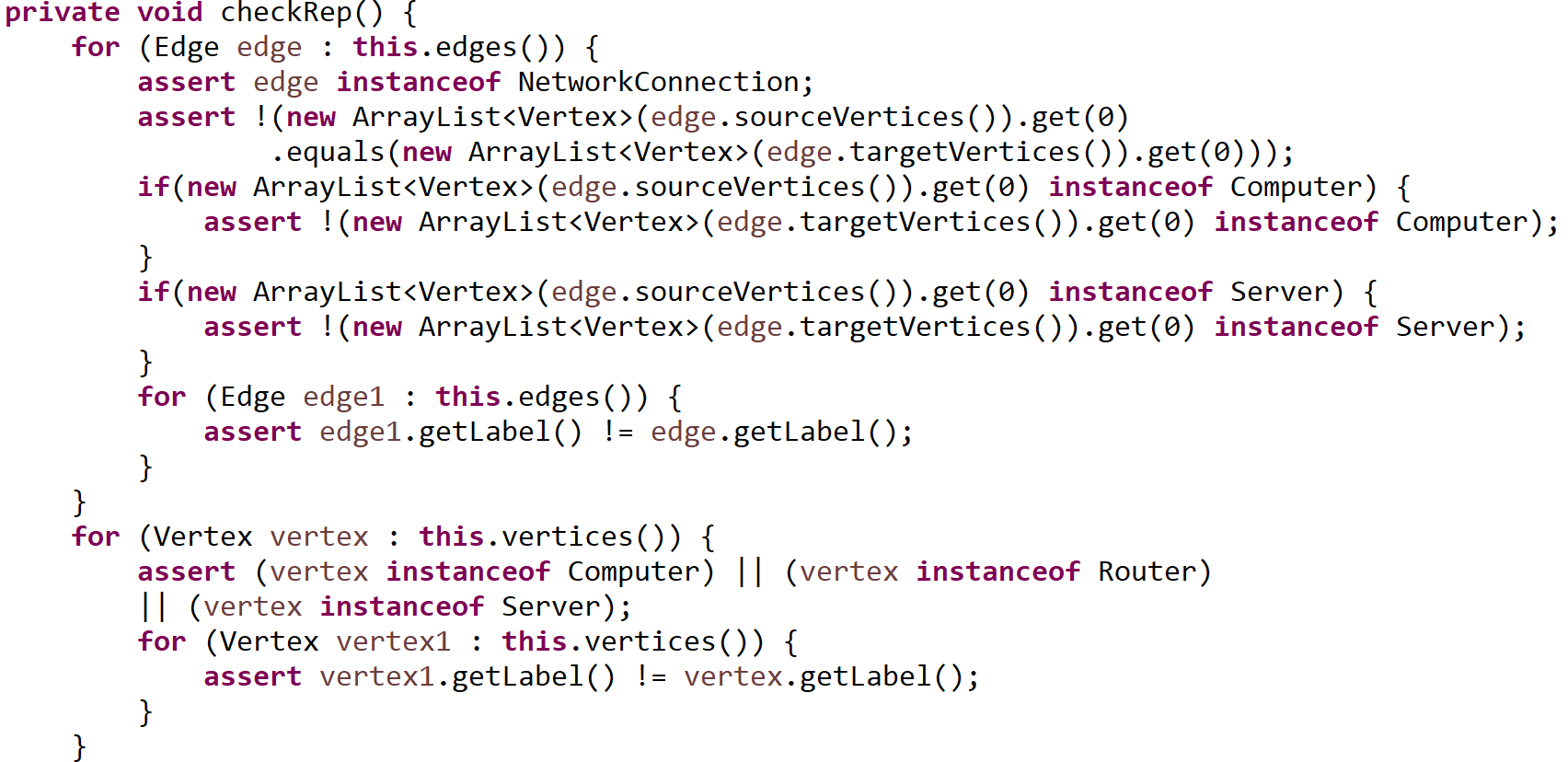
GraphPoet的checkRep



MovieGraph的checkRep



NetworkTopology的checkRep



### Assertion保障pre-/post-condition

## Logging

### 写日志

### 日志查询

## Testing for Robustness and Correctness

### Testing strategy

### 测试用例设计

### 测试运行结果与覆盖度报告

## FindBugs tool（可选）

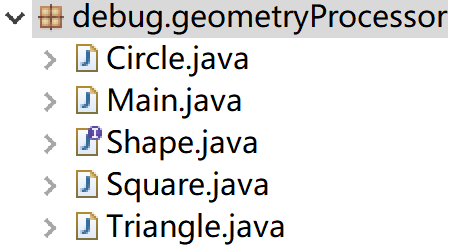
发现了哪些错误，每种错误代表什么不良的编程习惯

对代码修改，消除这些错误。

## Debugging

### 待调试程序

选择geometryProcessor作为待调试程序



### 理解待调试程序的过程

该程序功能非常简单，主要就是创建Shape类和继承自Shape类的其他图形类，建立图形对象，并打印出对象的各个参数

### 发现并定位错误的过程

首先最明显的错误是IDE自动检测出的错误，有错误提示

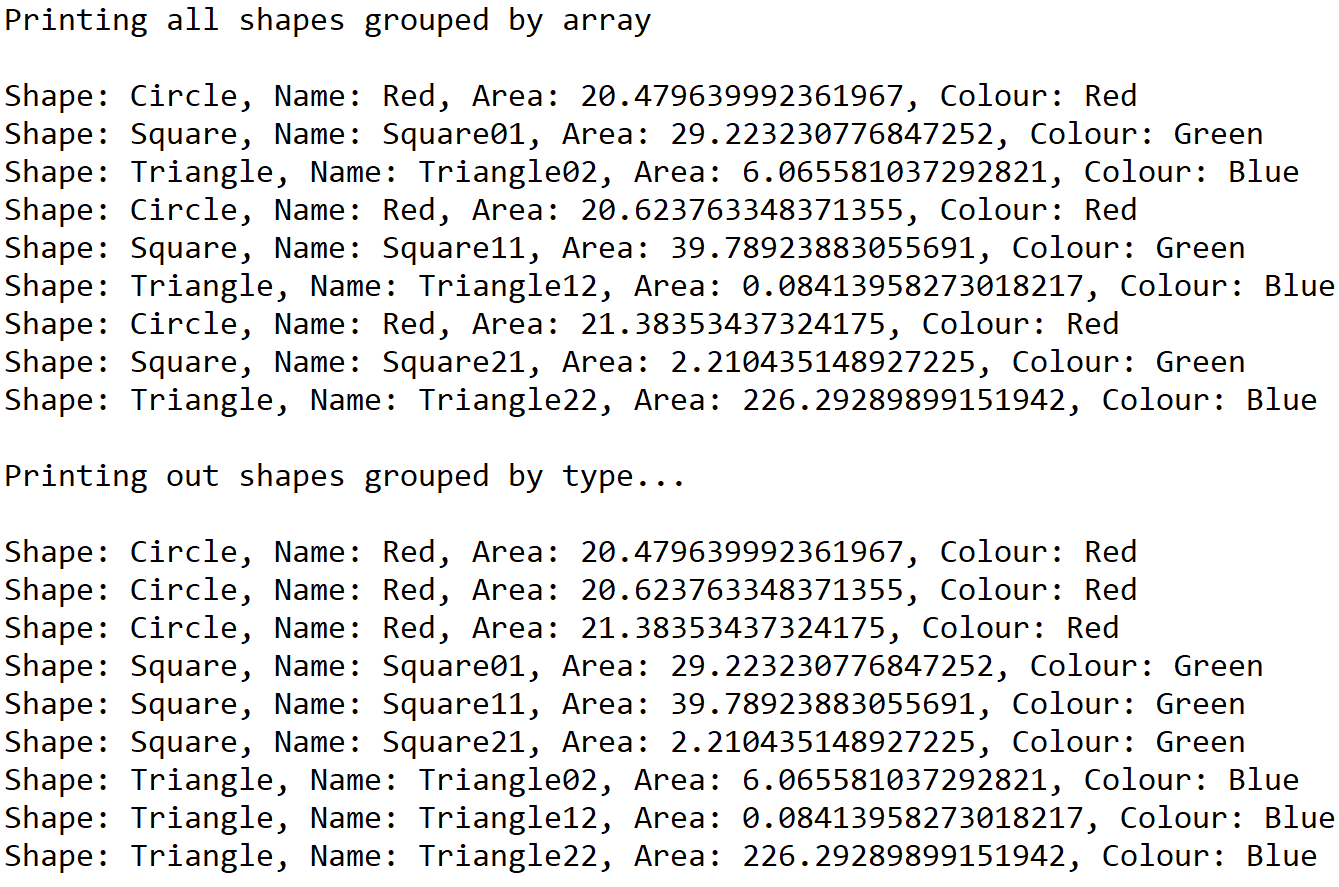
其次在Main函数中还有一些会造成数组越界的错误，会在运行时向控制台抛出错误，根据错误提示可以定位错误位置

### 如何修正错误

在Square和Circle类里分别有一个错误，错误类型是一致的，都是没有实现接口中的某一个方法，只需要把该方法填上就可以修正错误，而修正的方法也很简单，只需要直接返回对象的成员变量即可。

在Main函数中的错误有两处，第一处实在3\*3的数组定义时，可以直接将矩阵大小定义为3\*3减少不必要的麻烦。第二处实在14、15行的二层嵌套for循环中，第二层循环应改为n<shape[i].length这样逻辑正确，避免了数组越界的异常。

### 结果



# 实验进度记录

请尽可能详细的记录你的进度情况。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 计划任务 | 实际完成情况 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 实验过程中遇到的困难与解决途径

# 实验过程中收获的经验、教训、感想

本节除了总结你在实验过程中收获的经验和教训，也可就以下方面谈谈你的感受（非必须）：

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？
2. 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？
3. “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？
4. 过分谨慎的“防御”（excessively defensive）真的有必要吗？
5. 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？
6. 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？
7. 关于本实验的工作量、难度、deadline。
8. 到目前为止你对《软件构造》课程的评价和建议。