haoz

**2024年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 |  |
| 学号 | 2022112831 |
| 班号 | 2237101 |
| 电子邮件 | 1162030037@qq.com |
| 手机号码 |  |

**目录**

[1 实验目标概述 1](#_Toc838645221)

[2 实验环境配置 1](#_Toc1175054086)

[3 实验过程 3](#_Toc874405590)

[3.1 Magic Squares 3](#_Toc904154709)

[3.1.1 isLegalMagicSquare() 4](#_Toc533907991)

[3.1.2 generateMagicSquare() 6](#_Toc1204927571)

[3.2 Turtle Graphics 8](#_Toc446894587)

[3.2.1 Problem 1: Clone and import 8](#_Toc1207010150)

[3.2.2 Problem 3: Turtle graphics and drawSquare 8](#_Toc1089061488)

[3.2.3 Problem 5: Drawing polygons 9](#_Toc853305435)

[3.2.4 Problem 6: Calculating Bearings 10](#_Toc608651379)

[3.2.5 Problem 7: Convex Hulls 10](#_Toc1139116192)

[3.2.6 Problem 8: Personal art 12](#_Toc309125939)

[3.2.7 Submitting 13](#_Toc716714680)

[3.3 Social Network 13](#_Toc587850737)

[3.3.1 设计/实现FriendshipGraph类 14](#_Toc1582560559)

[3.3.2 设计/实现Person类 16](#_Toc1510347018)

[3.3.3 设计/实现客户端代码main() 16](#_Toc1145623986)

[3.3.4 设计/实现测试用例 17](#_Toc163953700)

[4 实验进度记录 20](#_Toc348316799)

[5 实验过程中遇到的困难与解决途径 20](#_Toc120019071)

[6 实验过程中收获的经验、教训、感想 20](#_Toc673381764)

[6.1 实验过程中收获的经验和教训（必答） 20](#_Toc288487858)

[6.2 针对以下方面的感受（必答） 21](#_Toc1744838127)

# 实验目标概述

本次实验通过求解三个问题，训练基本Java编程技能，能够利用Java 面向对象开发基本的功能模块，能够阅读理解已有代码框架并根据功能需求补全代码，能够为所开发的代码编写基本的测试程序并完成测试，初步保证所开发代码的正确性。另一方面，利用Git作为代码版本管理的工具，学会Git的基本使用方法。

其中可以总结为以下四点

基本的Java 面向对象编程

基于Intellj Idea IDE进行Java编程

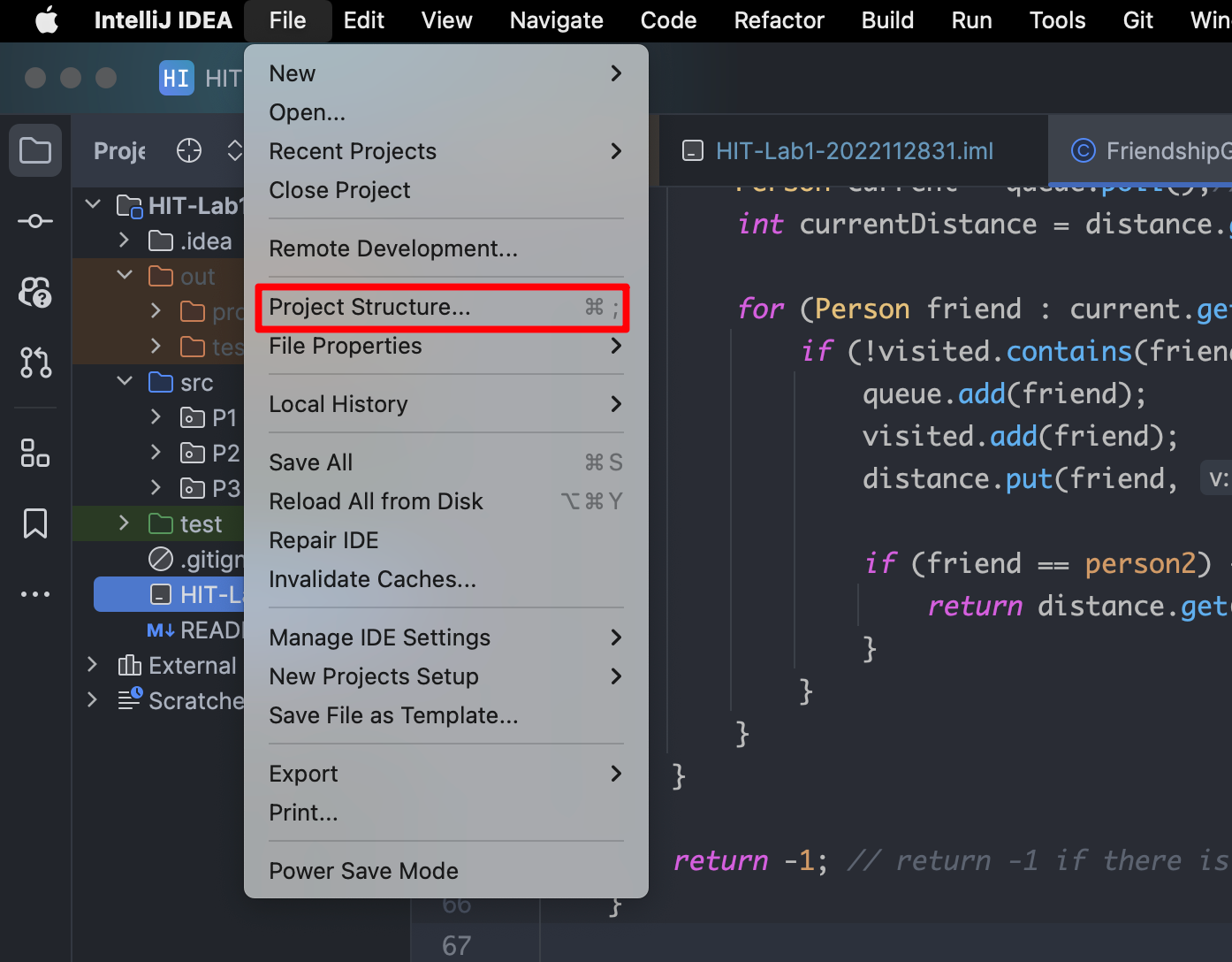
基于JUnit4的测试

基于Git的代码版本管理

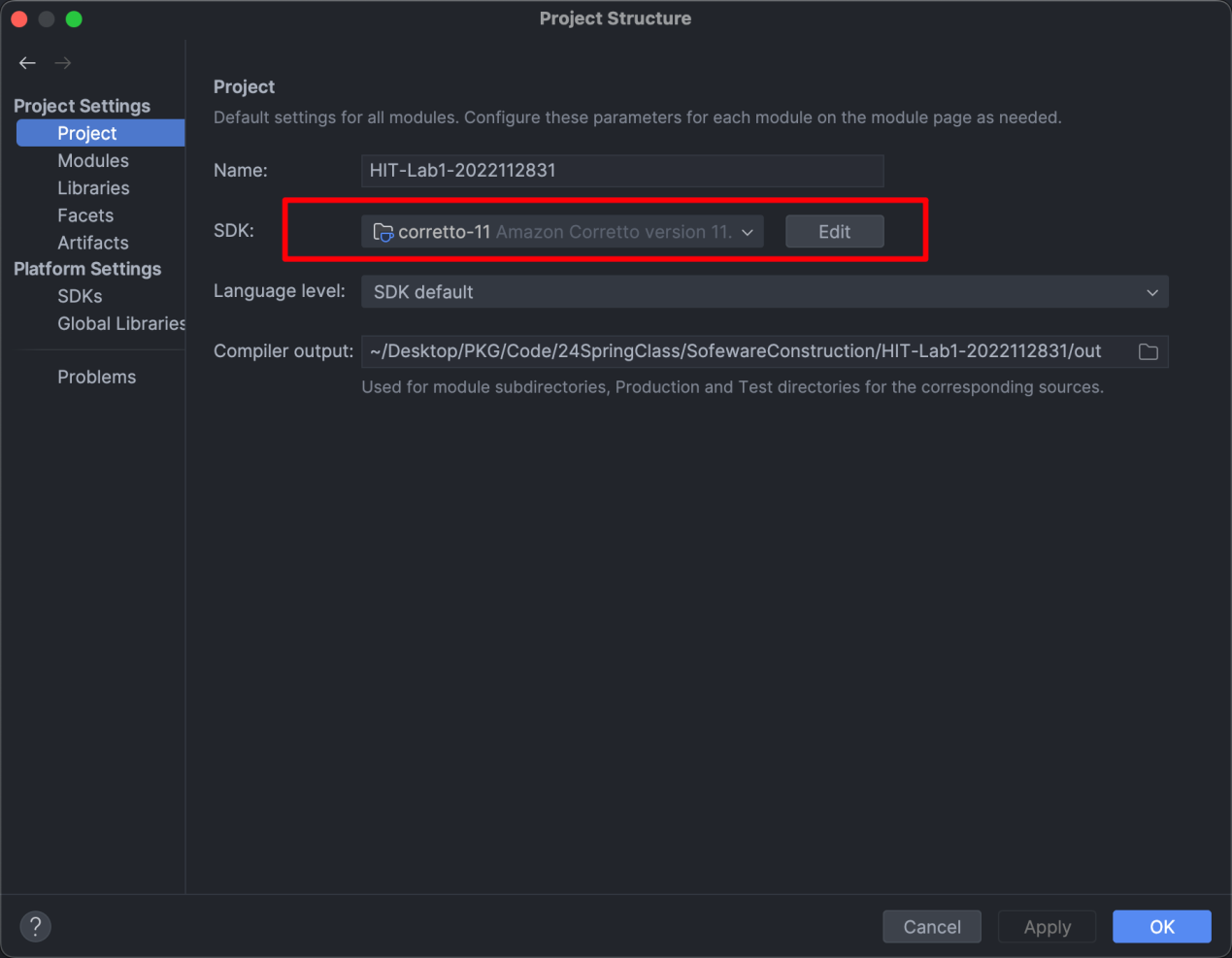
# 实验环境配置

实验设备: MacbookPro2018，在 Jetbrains 公司官网下载 Intellj Idea，安装后在本地更改 maven 仓库的镜像源，在本地的/Applications/IntelliJ IDEA.app/Contents/plugins/maven/lib/maven3/conf/settings.xml文件中修改 mirror 配置，添加腾讯云的镜像源。

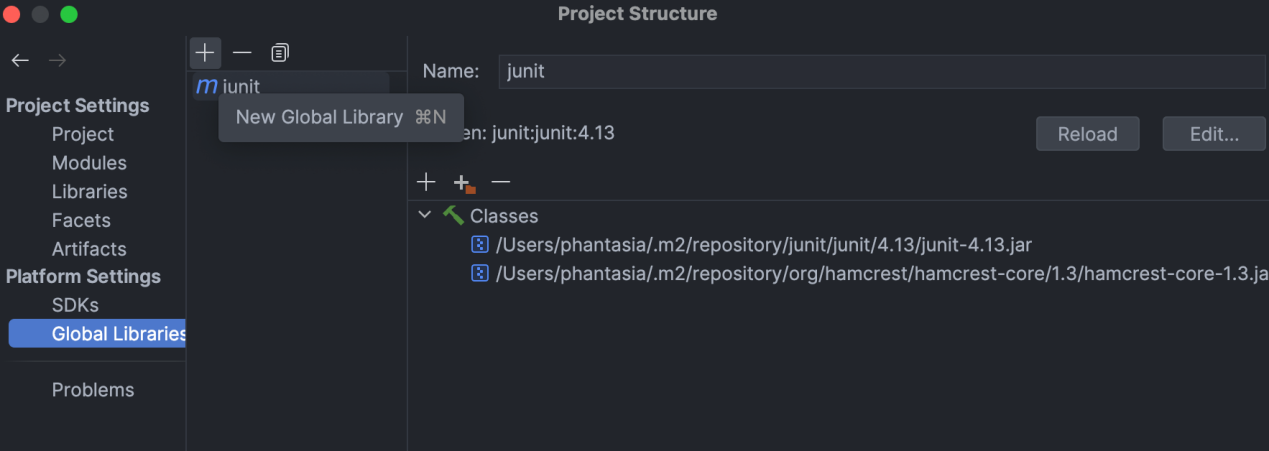
随后从 github上 clone 实验仓库，并且在仓库中新建一个 idea 项目，并且准备好实验的项目结构，在 Porject Structure 设置下，



配置使用 correto11（jdk11 的亚马逊发行版），



并且在 Global Libraries 中添加 JUnit4（确保 Idea 的 Plugins 中已经安装了 JUnit 插件）。到此为止，开发环境已经配置完成



终端使用 github 开源项目<https://github.com/ohmyzsh/ohmyzsh>。

git 使用macos 下的包管理工具 brew 安装，并且使用--cask 参数保证 git 版本最新。

配置过程中并未遇到明显的困难，大多数情况下百度就可以解决问题。

Github Lab1 仓库地址<https://github.com/ComputerScienceHIT/HIT-Lab1-2022112831>

# 实验过程

请仔细对照实验手册，针对三个问题中的每一项任务，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（但无需把你的源代码全部粘贴过来！）。

为了条理清晰，可根据需要在各节增加三级标题。

## Magic Squares

// 在这里简要概述你对该任务的理解。

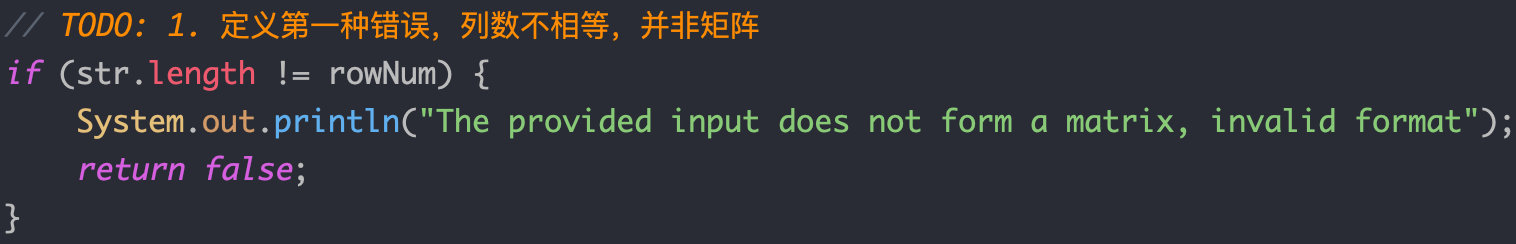
实验要求我们对一个基础图形“MagicSquare”有所认识，所谓幻方就是，一个 n\*n 矩阵，每一行每一列以及每一个对角线的数字的和相等。在 MagicSquare.java函数中的 main 函数中我们需要读入预先准备的 5 个 txt 文本文件，判断不同条件下幻方时候成立，不成立的时候需要输出错误原因。并且需要对实验手册中给出的生成幻方的代码进行测试，并且在可能出错的地方抛出异常或者结束函数执行，将生成的幻方写入文件 6.txt 中。最后绘制流程的执行图。

### isLegalMagicSquare()

// 按步骤给出你的设计和实现思路/过程/结果。

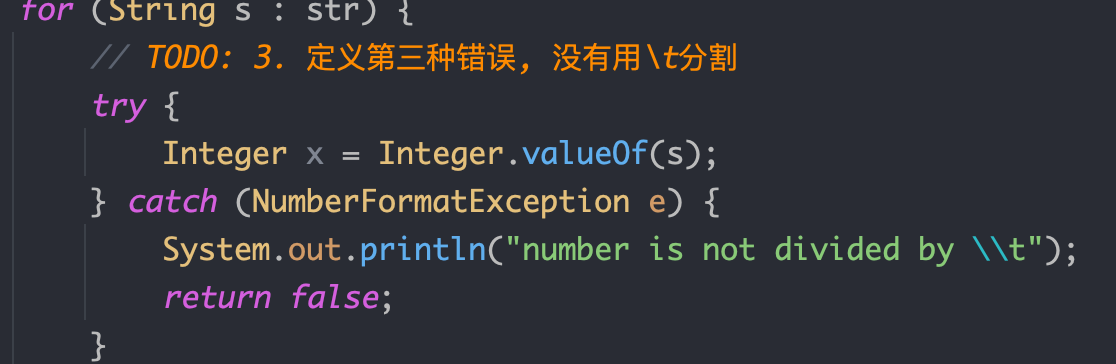
#### 首先判断矩阵的合法性

设计思路是由于是对矩阵进行操作，所可以尽可能在一些地方通过增加耦合度的方式来减少反复遍历的次数，从而达到代码执行效率的提高。

通过分别判断矩阵的列行数是否想等，以及每一行的长度是否相等来判断矩阵的合法性。



再通过valueOf() 的抛出的异常来判断是否由 Tab 分隔符进行分割。

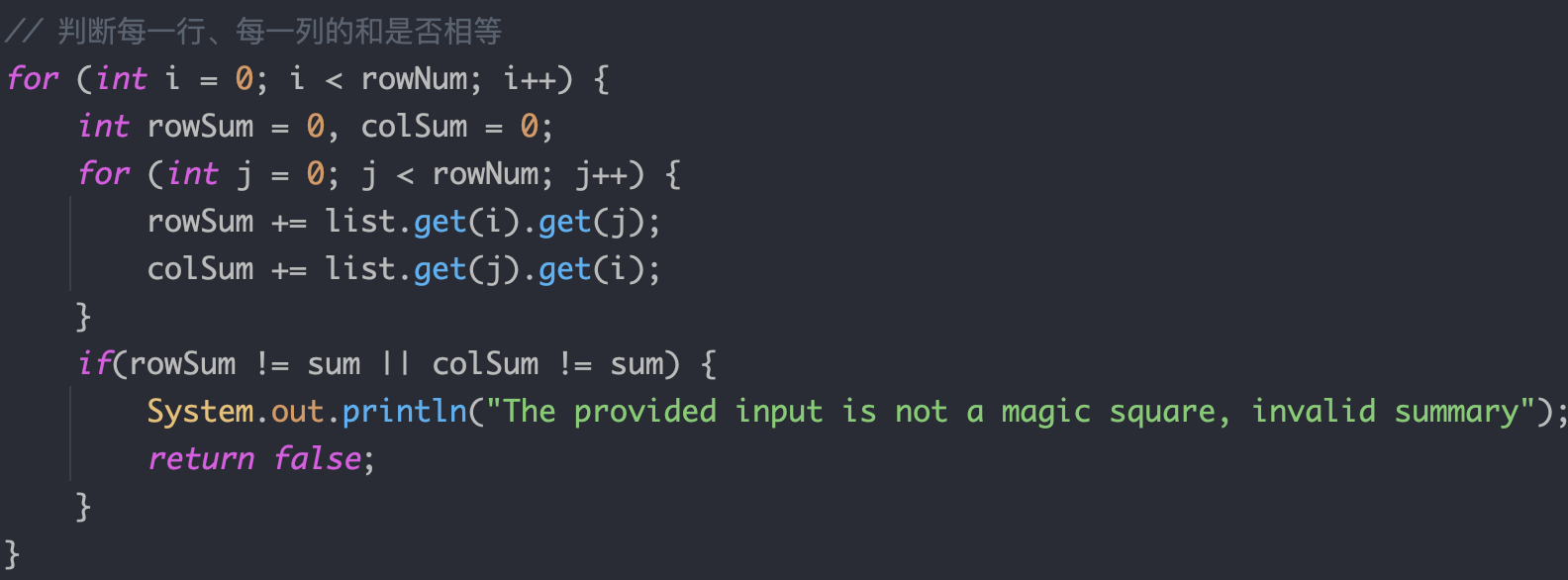


再通过字符串的 contains 函数来判断是否含有‘-’、‘。’等特殊字符，来判断小数和负数的存在。

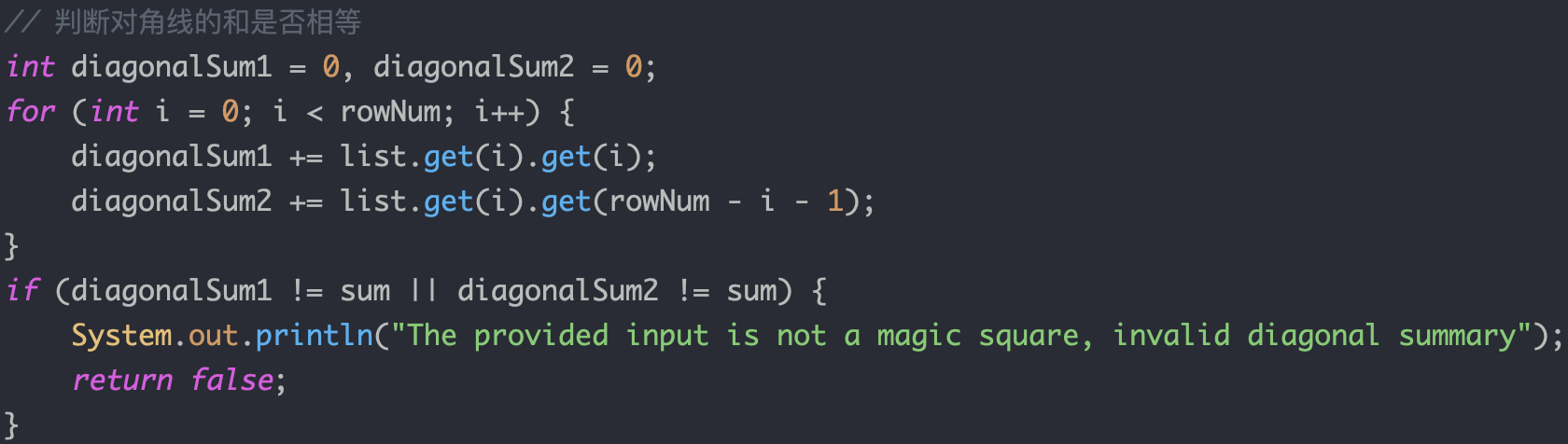


#### 幻方合法性判断成功后的验证

* 先判断每一行每一列是否相等：

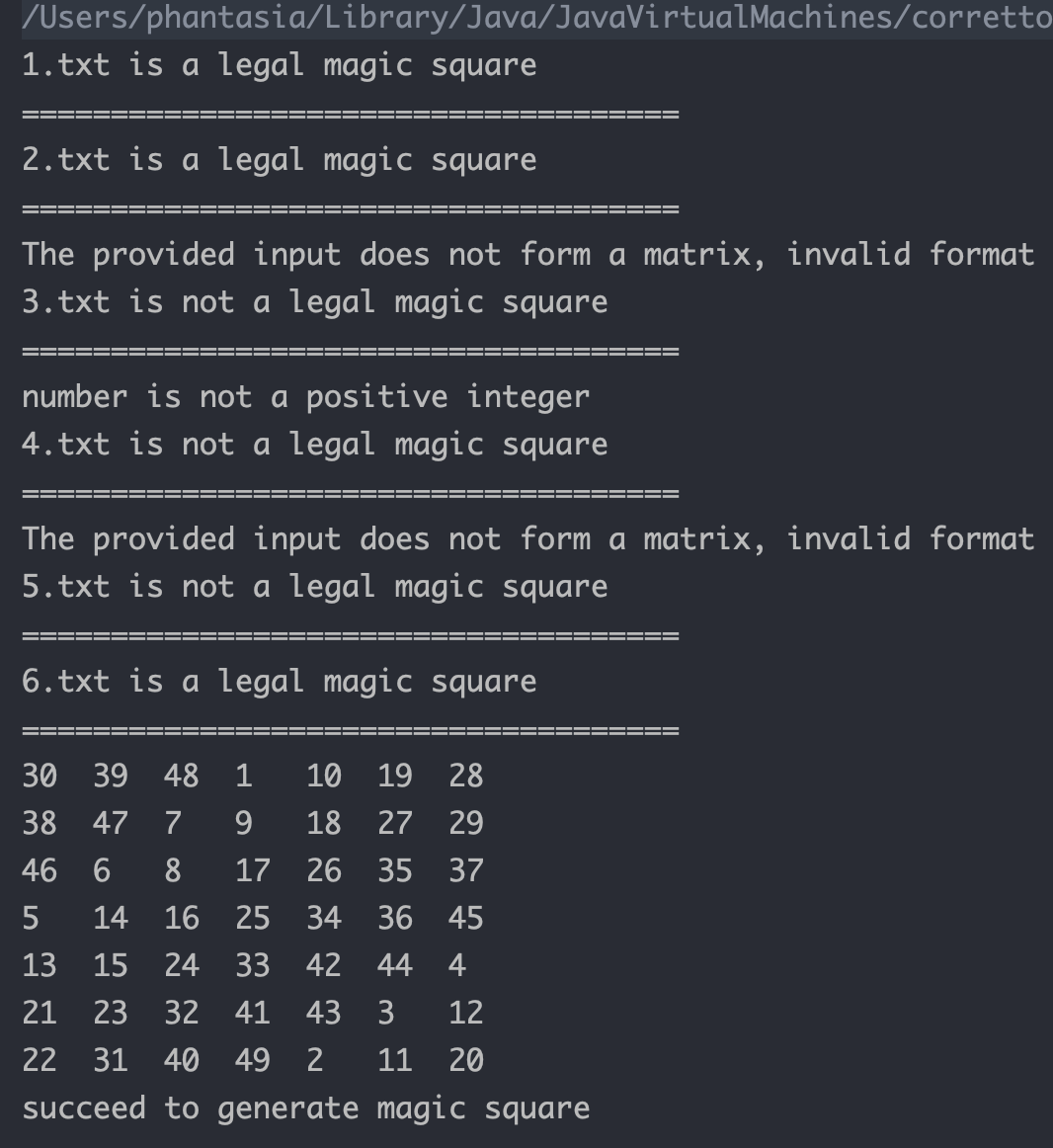


* 再判断对角线是否相等



通过上述所有的判断和验证后就可以得到一个合法的幻方。

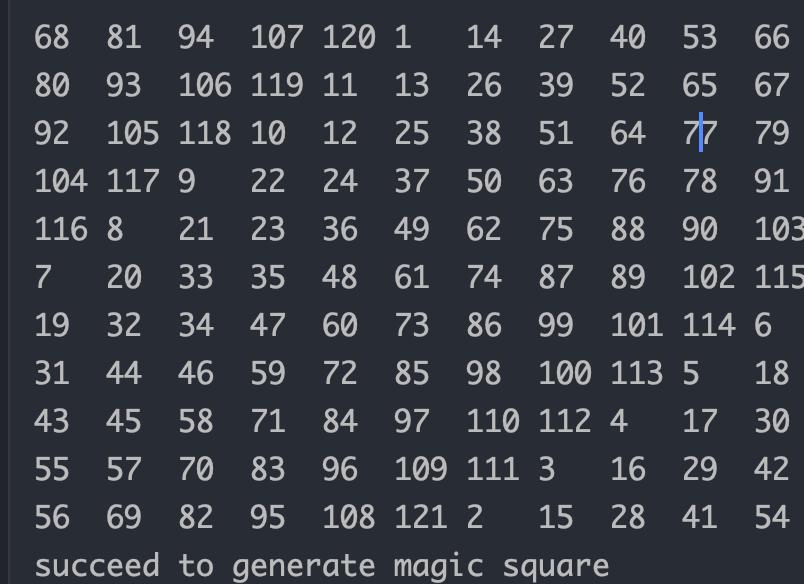
JUnit 进行 Test 的结果

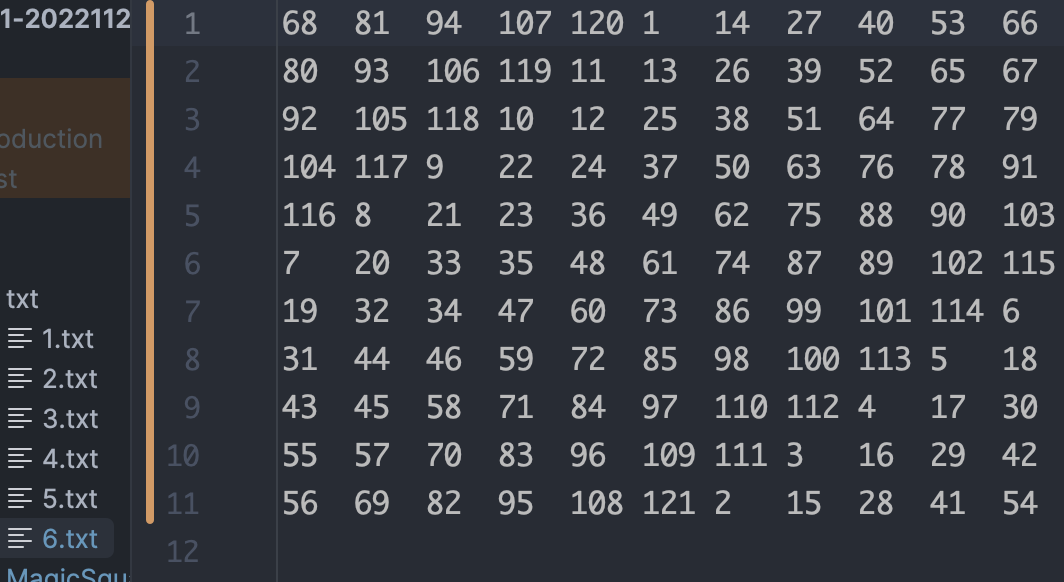


### generateMagicSquare()

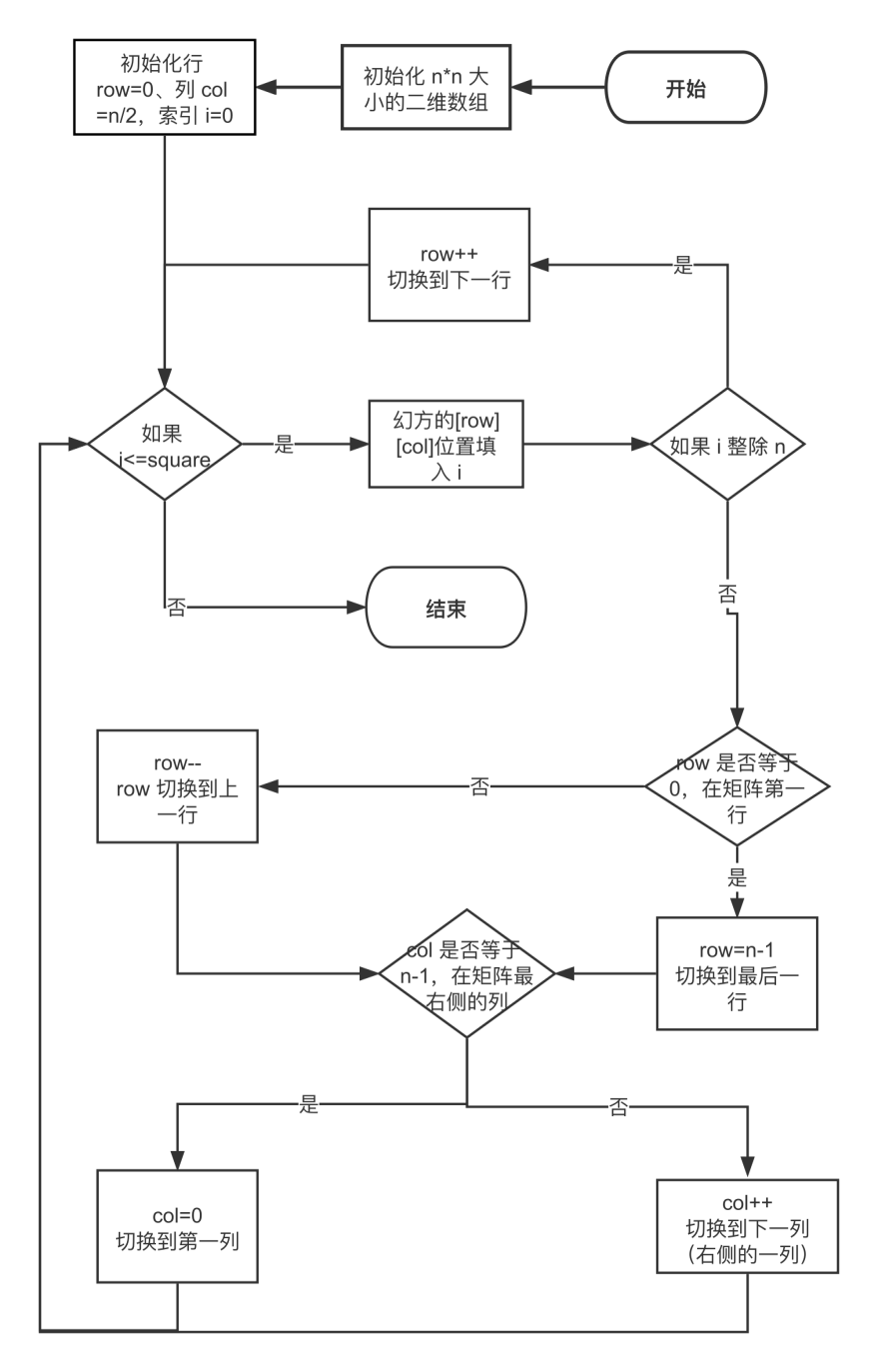
实际上是La Loubere 法奇数幻方构造，通过首先在第一行中间放入 1，后通过判断i 是否整除 n 来将row移动到下一行，否则将行上移，列右移，得到下一个应该插入数字i的为止。

下图是函数执行后的矩阵。

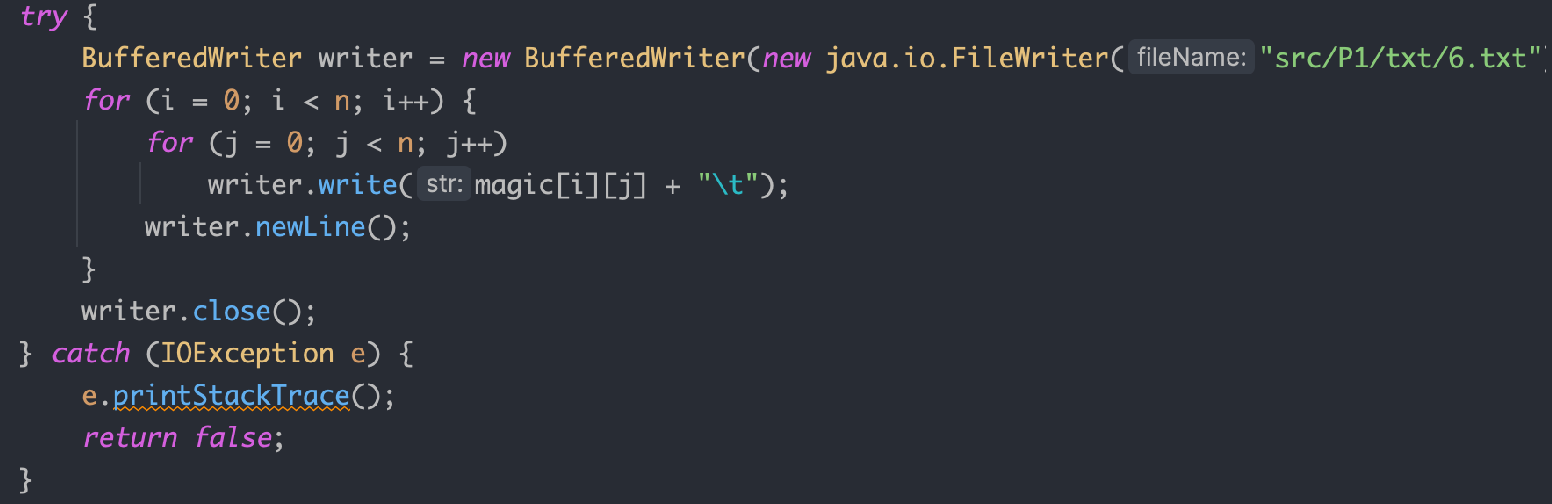




下图是函数执行的流程图。



幻方生成完后将幻方写入文件，使用 BufferedWriter，使用\t分割。



关于实验手册中提到的两个异常问题，我写了一篇博客对这两个问题做个一个简单的了解<https://www.cnblogs.com/Phantasia/p/18087086>。

## Turtle Graphics

// 在这里简要概述你对该任务的理解。

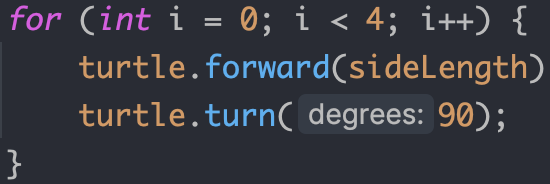
实验需要实现一系列函数，包括实现画正方形、绘制多边形、计算方位角、计算凸包的函数。文件所需要的基本代码需要从 github 仓库导入，当实现所需的函数后，可以绘制出一些图形，这类似于 python 的 turtle 包。

### Problem 1: Clone and import

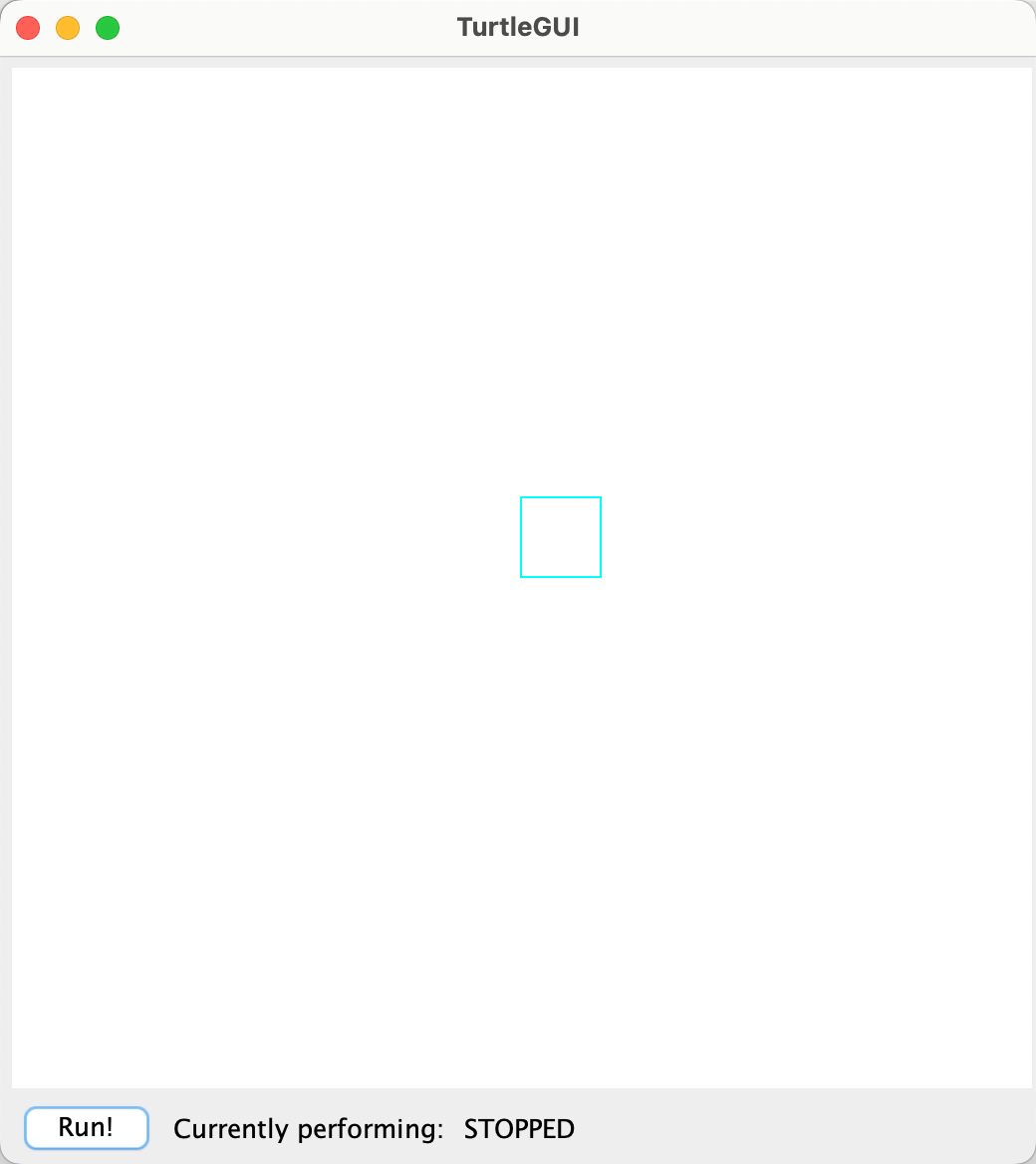
打开终端（本机为 zsh），进入 src/P2 目录下，使用 git clone 命令克隆对应的 P2 仓库，并将文件放入对应的目录下，调整每个 java 文件的目录结构，使其能在本地项目中进行开发。

### Problem 3: Turtle graphics and drawSquare

完成一个简单的 Turtle 函数的使用，绘制了一个简单的正方形。



绘制出的图形如下

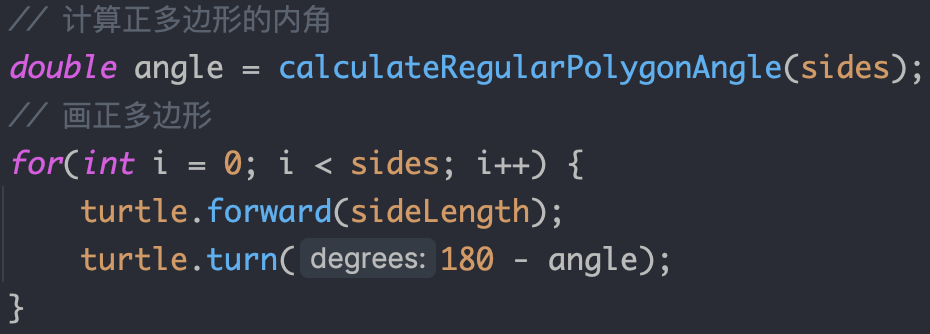


### Problem 5: Drawing polygons

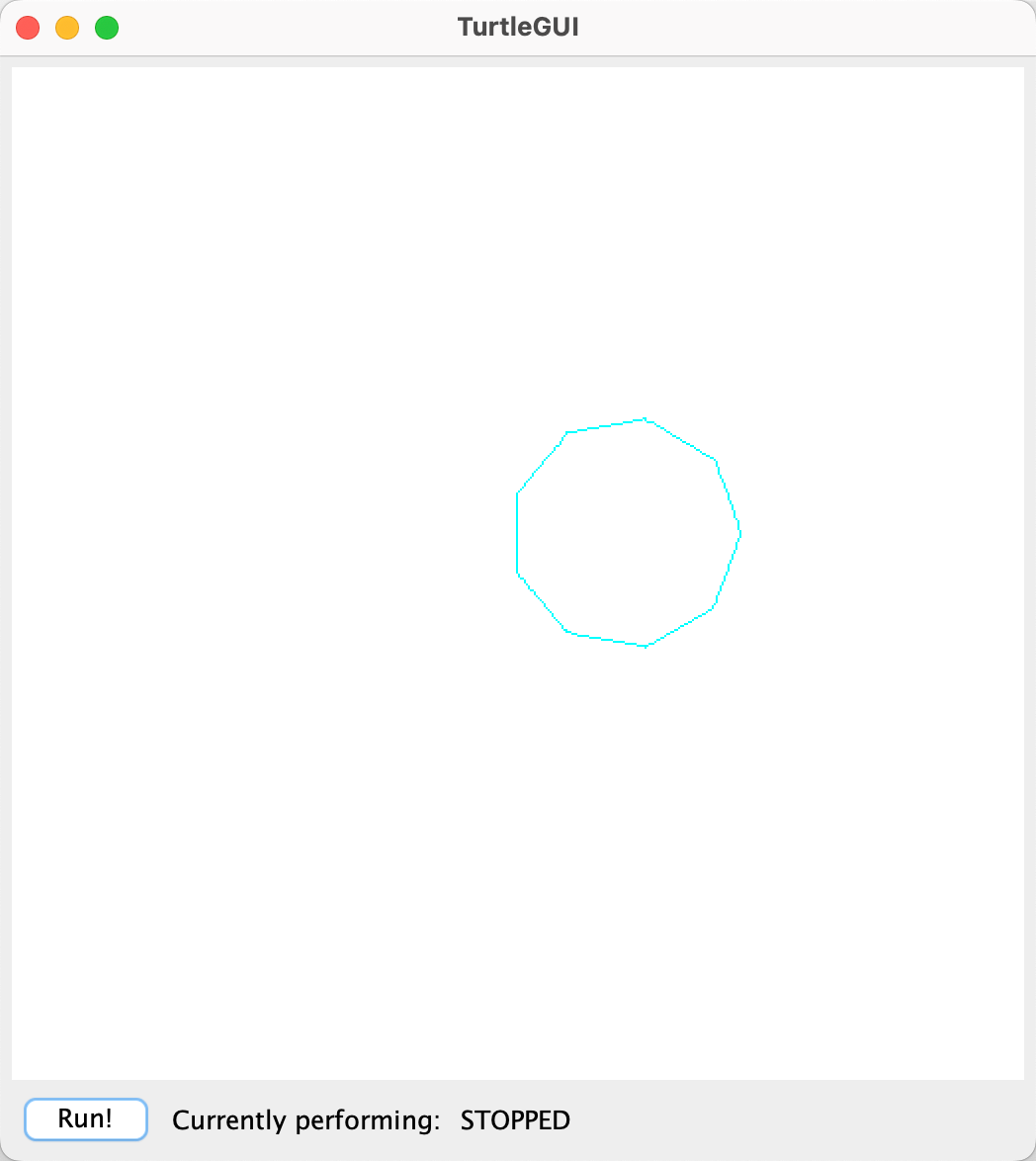
需要实现计算多边形内角的函数，从而计算绘制多边形的时候需要的转角。关键是返回一个正确的转角值。



使用设计的calculateRegularPolygonAngle()函数计算正多边形的内角，并且在绘图函数中调用，在画完对应的边后进行 turn() 函数进行角度的切换，在 drawRegularPolygon() 函数中书写具体画图函数

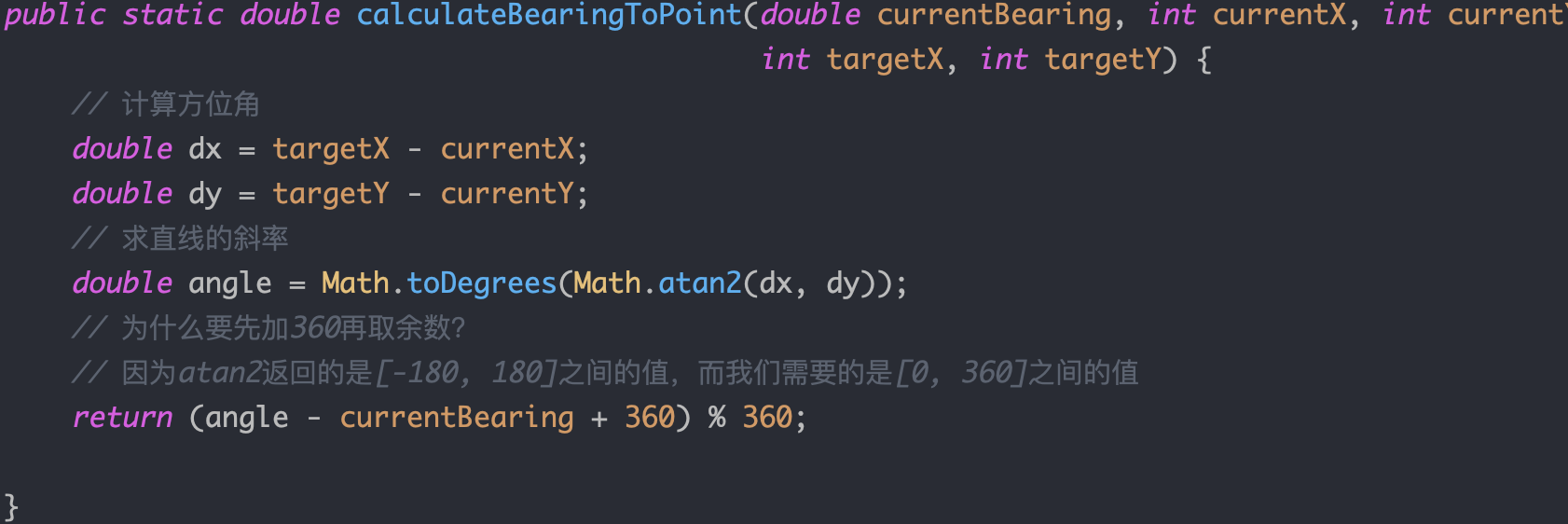


绘图结果如下，以九边形为例



### Problem 6: Calculating Bearings

Problem6 需要设计一个计算方向角的函数calculateBearingToPoint()，currentBearing 参数是当前的基准角度，currentX 和 currentY 是当前的基准点，targetX 和 targetY 是目标点，需要计算目标点和基准点连线相对于基准角的角度，通过 dx 和 dy 计算 x 和 y 的差值，通过 Math 库下的 atan2 函数计算斜率，并通过 toDegrees 函数将斜率转换为角度，因为转换的角度是从-180°～+180 ° 的范围，所以将角度+360 再模 360，取得正确的方向角。



### Problem 7: Convex Hulls

凸包问题是本次实验中最关键的问题，主要涉及一些简单算法的使用，本次实验我使用 Graham 算法来完成。

Graham 算法的基本实现原理是

1. 找到所有点中纵坐标y最小的点，也就是这些点中最下面的点，记为p0。

2. 然后计算其余点与该点的连线与x轴之间夹角的余弦值，将这些点按其对于最低点的正弦值从大到小排序，排序好的点记为p1, p2, p3, ......

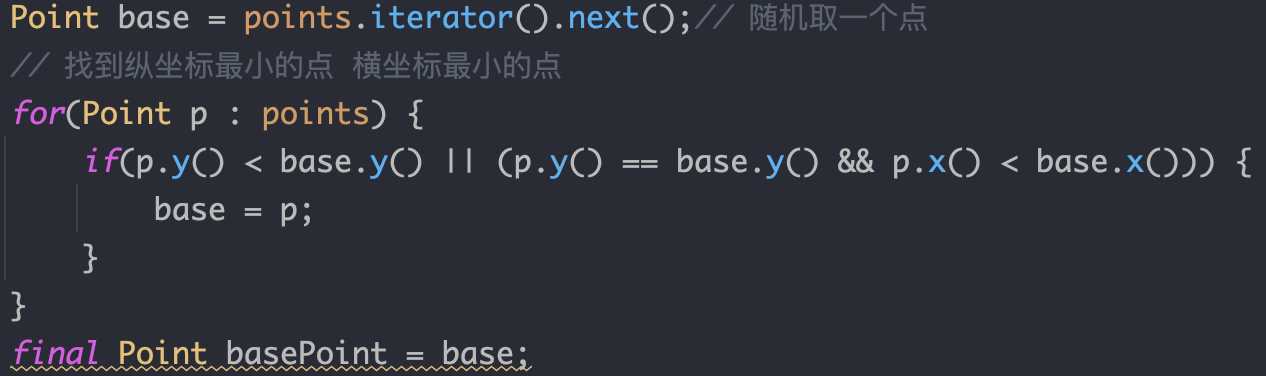
3. 将最低点p0和排序好的点中的第一个点p1压入栈中，然后从p2开始计算，计算栈顶两个点与该点三点向量是否是逆时针转动，若是，则将该点压入栈中；若否，则将栈顶元素推出，此时再选取栈顶2个元素与当前点进行比较，如果为逆时针转动，压入栈中，如果否，则将栈顶元素推出。如此往复。

下面是部分代码实现

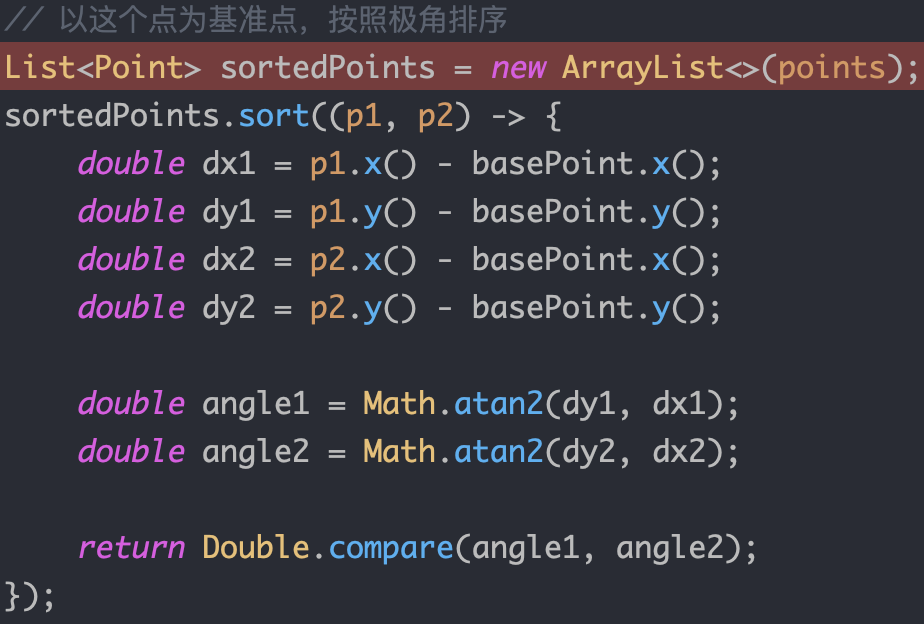
1. 边缘情况处理



1. 找到基准点



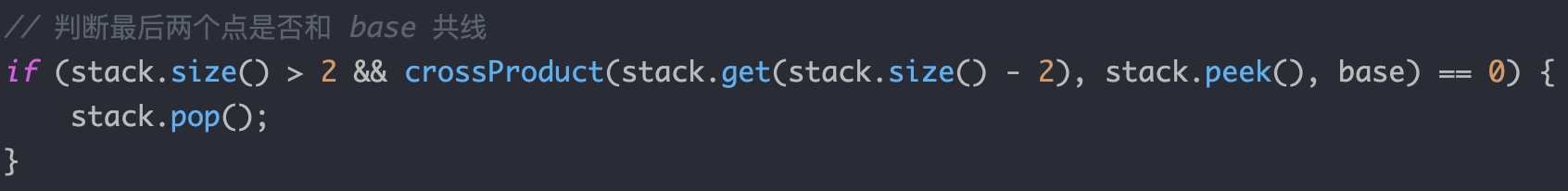
1. 使用Lambda 表达式对数组中的点按照极角大小顺序排序



1. Graham 算法主体实现



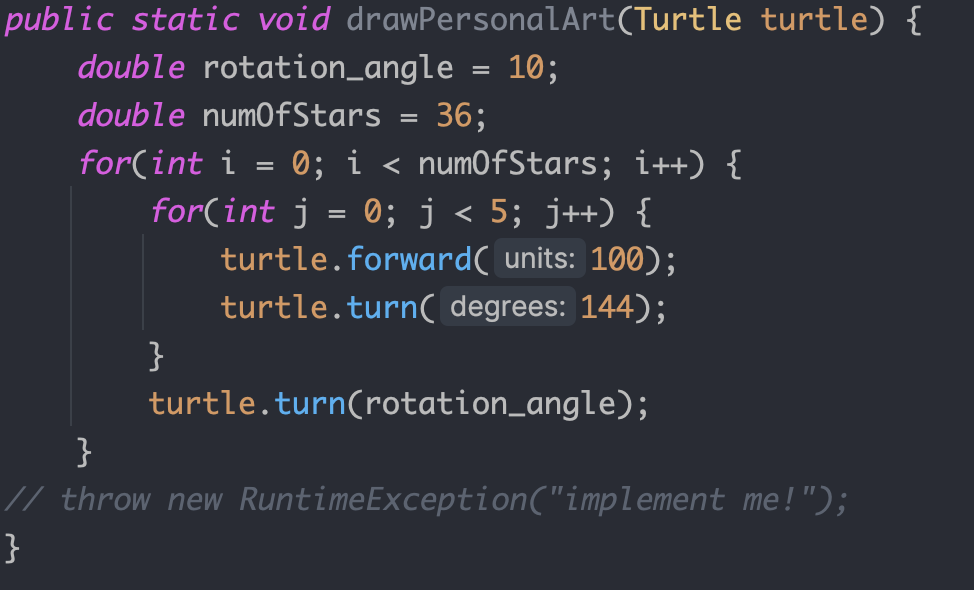
1. 处理最后两个点和基准点共线的情况



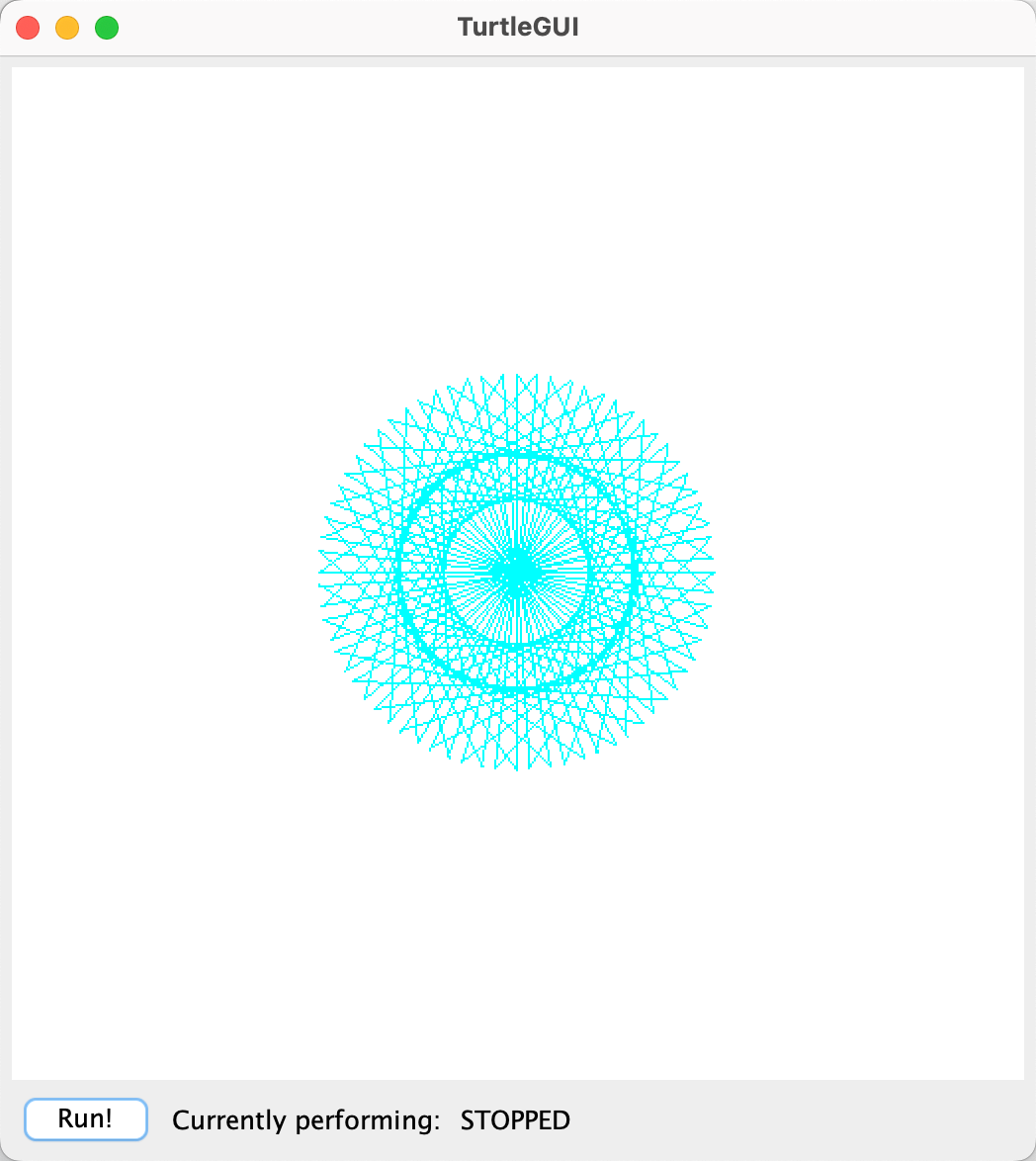
在算法全部结束后，返回一个 HashSet 用栈 stack 初始化，得到的就是闭包中的所有点。

### Problem 8: Personal art

画了一个很多五角星旋转的图形，代码实现如下



效果如下



### Submitting

在 .gitignore文件中指定好忽略上传的文件、文件夹后，由于剩下的文件都需要进行提交，分别使用命令 git add . | git commit -m “P2 finished” | git push 就可以将代码上传到远程仓库中。

## Social Network

该任务涉及到创建一个QuestionshipGraph类来模拟社交网络中的人际关系，并计算图中两个人之间的最短距离。还需要一个辅助Person类。QuestionshipGraph类应该有一个getDistance方法，该方法接受两个Person对象作为参数，并以整数形式返回它们之间的最短距离。如果两个人之间没有链接（意味着从第一个人到第二个人之间不存在路径），则应该返回-1。

代码中的 Rep 均定义为私有变量，以保证程序运行的安全。

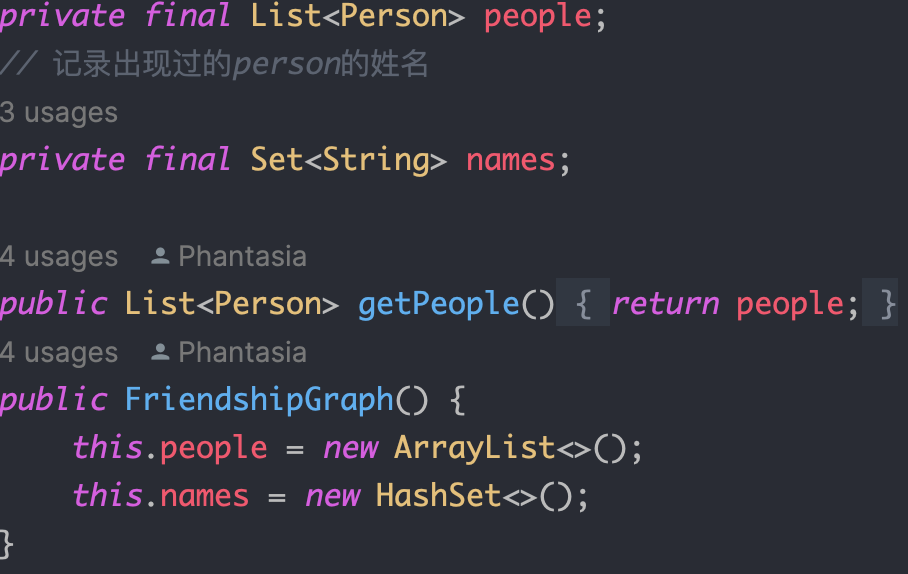
### 设计/实现FriendshipGraph类

3.3.1.1 设计：

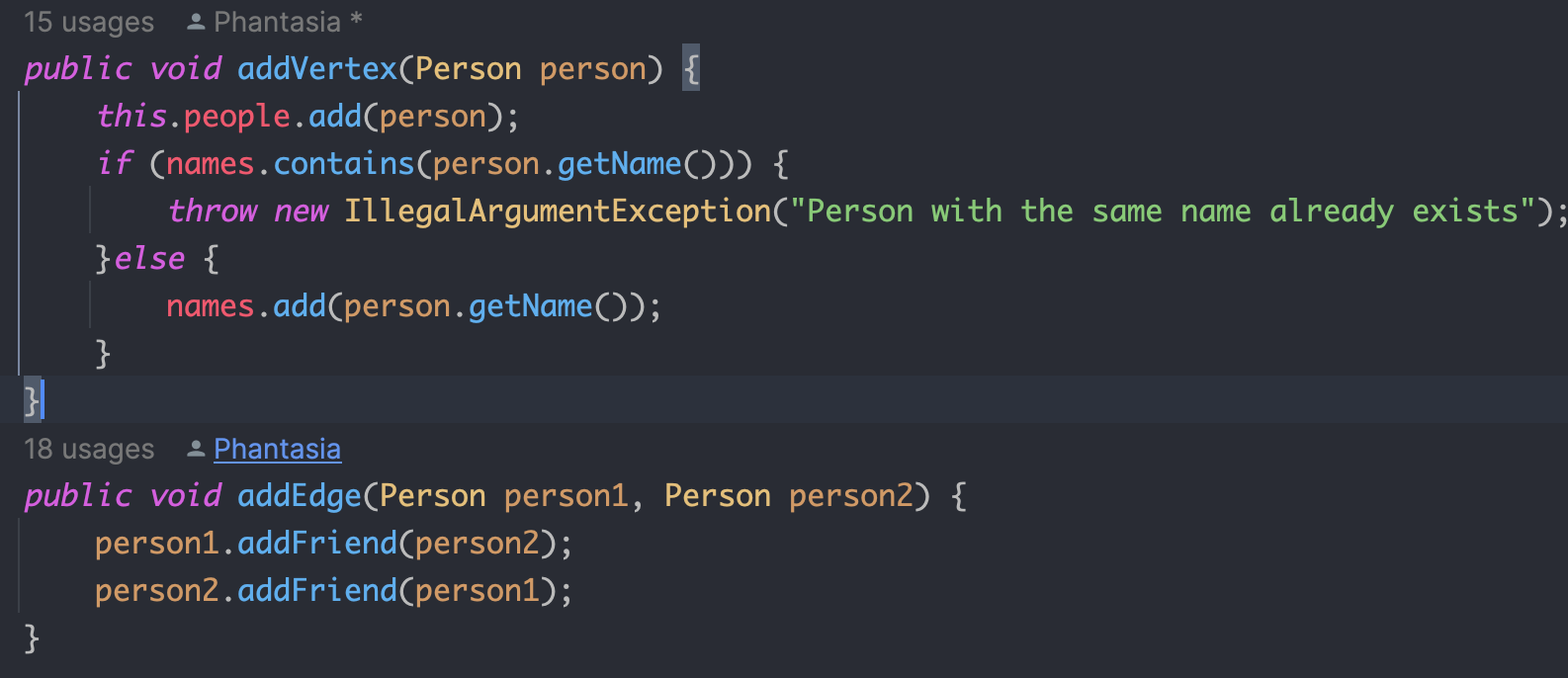
* 使用List存储加入表中的人（Vertex）
* 使用 Set 确认表中出现的名字的唯一性
* 提供一个构造器初始化 FriendshipGraph 类，并且初始化 List 和 Set
* 提供一个方法向表中添加 Vertex
* 提供一个方法向表中添加 Edge
* 提供一个方法获取任意两个节点之间的距离

3.3.1.2 实现：

使用 List 存储加入表中的节点（Person），使用了较为特殊的设计，没有设置一个数组来存储边，但是每一个 Person 类中有一个List 数组，存储了当前 Person 的朋友，这样可以通过BFS来进行搜索。下图是 FriendshipGraph 类中的两个 Rep 以及 constructor 和一个基础的 javabean 函数（get 方法）。



下图是添加 Person（addVertex 函数）和添加朋友关系（addEdge函数）两个函数的实现，在addVertex 函数中判断当前 Person 的名称是否出现过，如果已经出现则抛出异常。在 addEdge 函数中，由于朋友关系是双向的，并且图总体上来说是无向图，所以需要在两个 Person 下分别调用 addFridend 函数。



getDistance 代码主要实现思路是 BFS广度搜索，具体实现如下图所示，如果输入的两个 Person 相同则返回 0，否则进入深入广度搜索过程，distance 负责存储当前 Person 1 和索引到的 Person 的距离，queue 是用于 BFS队列，visited 是用户广度搜索中标志遍历过的辅助数组。



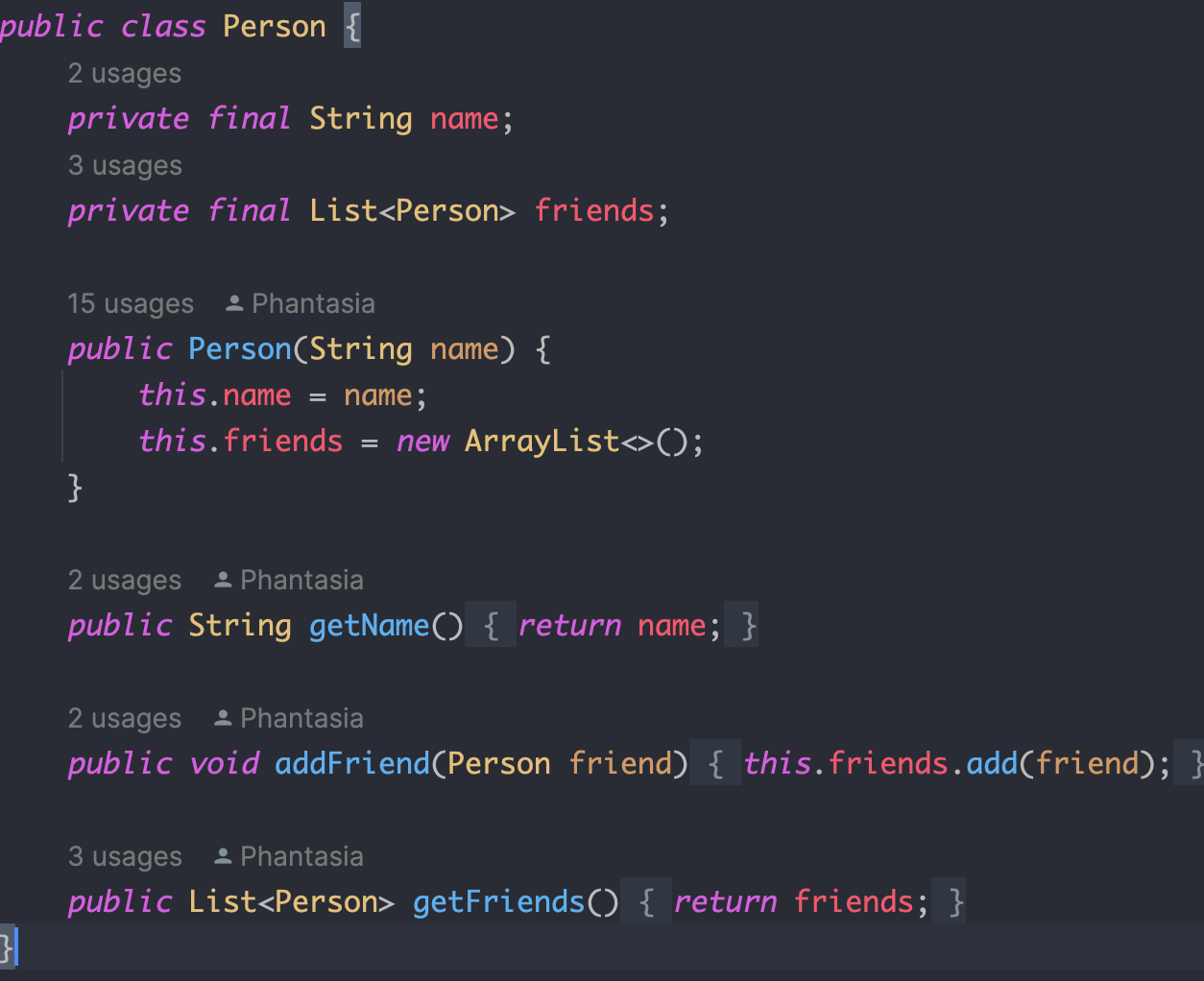
### 设计/实现Person类

#### 设计

* Person 类有 name 属性，标志当前 Person 的姓名
* Person 类有friends 的 List 数组，标志着当前 Person 的朋友们。
* Person 类有一个构造器，传入参数 name，并且初始化 List 数组
* Person 类有一个 public 方法 addFriend，用于为当前 Person 添加朋友关系。

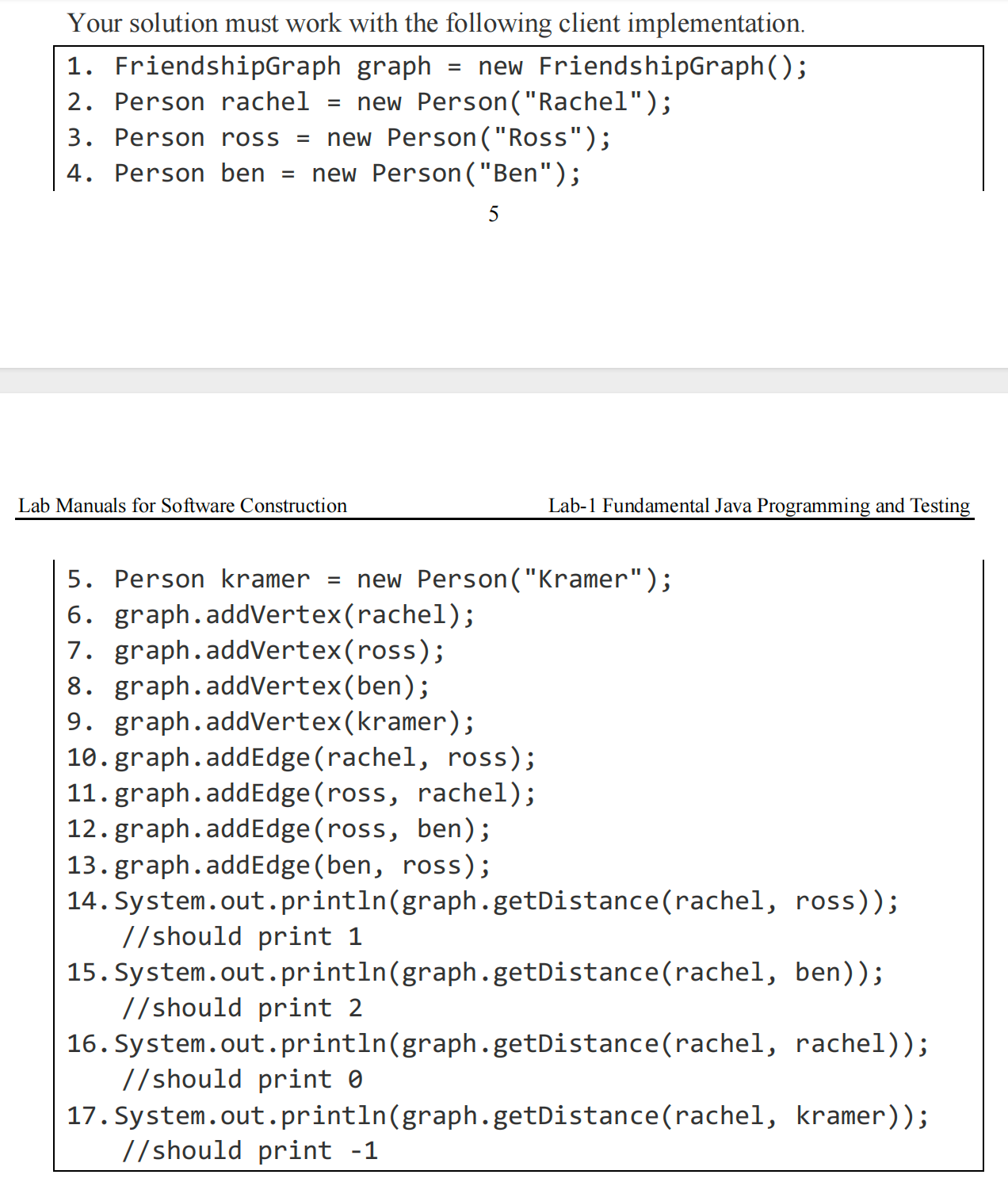
#### 实现

具体实现较为简单，如下图所示



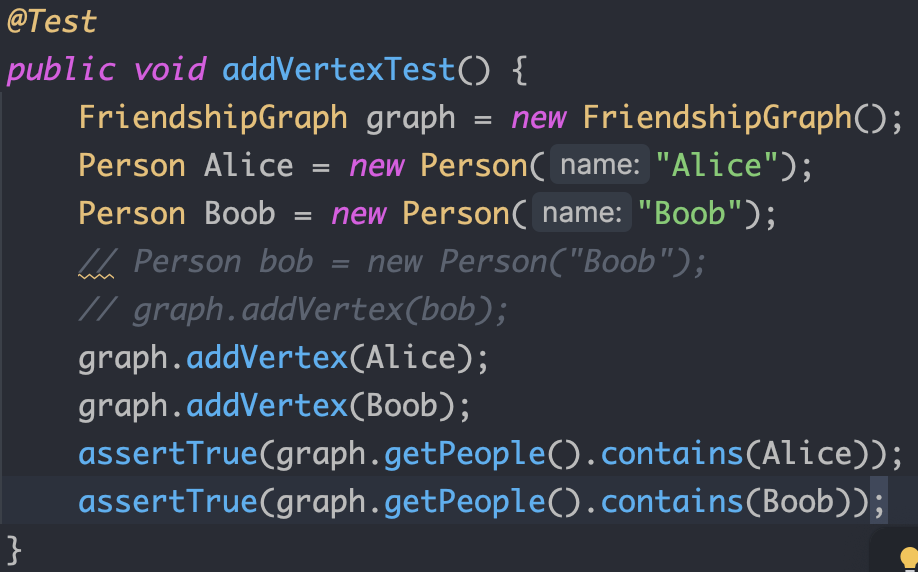
### 设计/实现客户端代码main()

设计和实现均参考实验手册中内容



### 设计/实现测试用例

addVertex函数的 test，如果加入相同的名称的 Person 会抛出异常。测试点是否添加到 Graph 中。



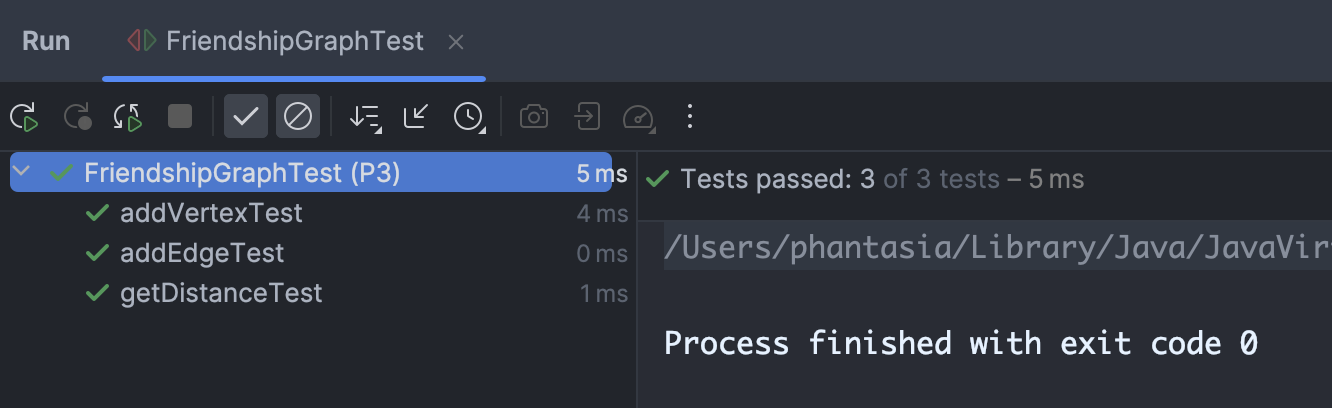
addEdge 函数的测试，测试实际存在的边



getDistance 函数的测试，通过添加进去一系列测试点，测试不同角色之间、同一角色的距离返回是否正确。代码实现如下。



测试结果如下



# 实验进度记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 2024-03-21 | 17:30-19:30 | 完成 P1 并且进行测试 | 按计划完成 |
| 2024-03-21 | 19:30-21:30 | 编写问题 3 并进行测试 | 按计划完成 |
| 2024-03-23 | 17:30-18:30 | 编写问题二除闭包问题其他代码 | 延迟一小时完成 |
| 2024-03-23 | 19:30-21:30 | 问题 二闭包问题完成 | 按计划完成 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的困难 | 解决途径 |
| 闭包算法的实现 | 通过网络上搜索学习了 Graham 算法，reference：<https://blog.csdn.net/u013066730/article/details/106661030> |
| 测试过程中闭包数组结果比应当出现的结果多出一个节点 | 经过自己在纸上模拟了函数的执行发现了问题所在，由于函数本身没有错误，所以问题出现在最后基准点和最终状态下栈顶和次栈顶的两个元素是否共线，如果共线应当删除栈顶的节点。原来的函数不可能在索引完所有节点后再继续执行下去索引基准点，所以需要在函数执行完毕后添加一段代码判断共线。 |

# 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（必答）

总得来说，写测试用例的确是一个费脑子的事情，在没有源码的时候先写 spec 再写测试用例（原来这就是测试优先），其实在写实验的过程中能感觉到国外的计算机教育真的十分成熟，完善的实验平台和作业平台，以及对实验十分细致的讲解，原本没开始写 turtle 的时候觉得纯英的 P2 部分应当有些难完成，事实上当自己沉下心去看 MIT6031 的作业内容时，发现每一个问题都解释的十分清楚，相对来说其实很容易完成，也希望老师可以继续将软件构造这门课发展成类似于 MIT这种教育模式的课程，真的会在实验过程中遇到问题解决问题，并且获得很多收获。

## 针对以下方面的感受（必答）

1. Java编程语言是否对你的口味？与你熟悉的其他编程语言相比，Java有何优势和不足？

因为之前学习过 Java，能感觉到 Java 作为一门 OO 编程语言，拥有极强的语言规范性，这为项目开发提供了十分优秀的漏洞控制，当然不足之处也正是如此，这导致了 Java 代码大多冗长且设计十分复杂，学习成本也是很高的。

1. 关于Eclipse或IntelliJ IDEA，它们作为IDE的优势和不足；

Eclipse 在实验课下课的时候听罗家乐学长吐槽了，Eclipse 的项目结构奇怪的异于常人，想要正确的去运行一个 Eclipse 项目的话配置还是比较复杂的。Idea 好在它的更新速度快，并且作为现代的 JavaIDE，它具有美观的 UI，人性化的项目管理，这点十分受人喜爱，当然 Idea 的问题就在于它的稳定性有所欠缺，这或许也是所有 Jetbrains 公司IDE 的一个缺点，会出现偶发性的无法关闭项目的情况，只可以强制关闭后台。

1. 关于Git和GitHub，是否感受到了它在版本控制方面的价值；

Git 是这个时代项目版本控制软件的奇迹，可以说 Linus 是一位真正的天才，Git 极大的方便了对项目代码的控制和开发管理，无论是个人还是企业都都是不可或缺的，Github 作为“老生常谈”基于 Git 的代码仓库，其作为全世界最大的开源社区其含金量不言而喻，丰富的代码资源注定了他是学习 coding 的第一选择。

1. 关于CMU和MIT的作业，你有何感受；

在前面的收获当中提到了，真的觉得国内外的CS教育差距体现的很明显，CMU 和 MIT作为传统的CS强校，他们对作业的难度设置合理，并且尤其是对作业的描述尤为清晰，可以很方便学生按照要求完成作业。

1. 关于本实验的工作量、难度、deadline；

工作量其实相对而言不是很高，唯一需要花点时间的地方应该是凸包算法的学习上，难度较为基础，deadline 还是很宽松的，甚至写报告的时间可能比代码的时间还要长。

1. 关于初接触“软件构造”课程；

初次接触，我更喜欢在实验课中对课上学习内容进行实践，因为课上的内容我还是不太理解，在实验课中能够淋漓尽致的体现出来。很喜欢这门课，希望老师可以在这门课上进行更多的实验内容的介绍。