

**2019年春季学期  
计算机学院《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 卿嘉诚 |
| 学号 | 120L022107 |
| 班号 | 202012 |
| 电子邮件 | qjc38643000@126.com |
| 手机号码 | 18007733101 |

**目 录**

[1 实验目标概述 1](#_Toc30493)

[2 实验环境配置 1](#_Toc24226)

[3 实验过程 1](#_Toc25940)

[3.1 Magic Squares 2](#_Toc19246)

[3.1.1 isLegalMagicSquare() 2](#_Toc22213)

[3.1.2 generateMagicSquare() 3](#_Toc15292)

[3.2 Turtle Graphics 5](#_Toc30735)

[3.2.1 Problem 1: Clone and import 5](#_Toc14884)

[3.2.2 Problem 3: Turtle graphics and drawSquare 6](#_Toc9390)

[3.2.3 Problem 5: Drawing polygons 7](#_Toc20706)

[3.2.4 Problem 6: Calculating Bearings 9](#_Toc20030)

[3.2.5 Problem 7: Convex Hulls 11](#_Toc16368)

[3.2.6 Problem 8: Personal art 12](#_Toc7796)

[3.2.7 Submitting 13](#_Toc21837)

[3.3 Social Network 14](#_Toc12806)

[3.3.1 设计/实现FriendshipGraph类 15](#_Toc28347)

[3.3.2 设计/实现Person类 16](#_Toc31728)

[3.3.3 设计/实现客户端代码main() 16](#_Toc5761)

[3.3.4 设计/实现测试用例 17](#_Toc7087)

[4 实验进度记录 19](#_Toc31541)

[5 实验过程中遇到的困难与解决途径 20](#_Toc1631)

[6 实验过程中收获的经验、教训、感想 20](#_Toc17114)

[6.1 实验过程中收获的经验和教训 20](#_Toc8688)

[6.2 针对以下方面的感受 20](#_Toc31695)

# 实验目标概述

本次实验通过求解四个问题，训练基本Java编程技能，能够利用Java OO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作为代码配置管理的工具，学会Git的基本使用方法。

* 基本的Java OO编程
* 基于Eclipse IDE进行Java编程
* 基于JUnit的测试
* 基于Git的代码配置管理

# 实验环境配置

简要陈述你配置本次实验所需开发、测试、运行环境的过程，必要时可以给出屏幕截图。

特别是要记录配置过程中遇到的问题和困难，以及如何解决的。

本次实验使用Eclipse、JDK作为开发环境。参照了<http://web.mit.edu/6.031/www/fa18/getting-started/>的指南。

在配置中遇到了新建项目时出现不存在相应版本JDK的问题，结果发现是JDK没有下载指定版本，重新下载了才解决的。

在这里给出你的GitHub Lab1仓库的URL地址（Lab1-学号）。

https://github.com/ComputerScienceHIT/Lab1-120L022107.git

# 实验过程

请仔细对照实验手册，针对四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。

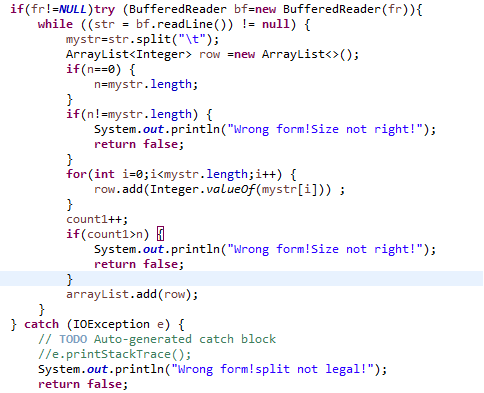
这个任务是实现判断txt文件内存放的矩阵是否是幻方的程序。主要问题集中在文件的读入和处理，遇到非法输入和报错时的反馈以及对对应矩阵的判断。

### isLegalMagicSquare()

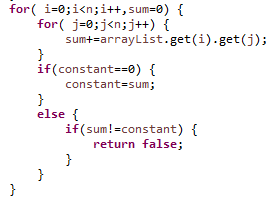
按步骤给出你的设计和实现思路/过程/结果。

首先将文件根据文件名以字符串形式按行读入，再用Integer.valueOf转换成整形，使用ArrayList<Integer>存放一整行的数字；整个矩阵用ArrayList<ArrayList<Integer>>表示。

在逐行读入的时候进行非法输入的检查和矩阵形状是否合法

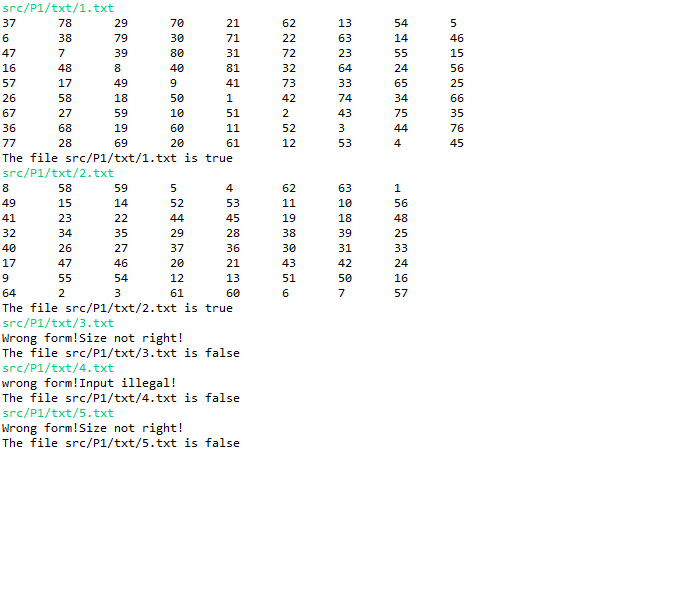


同时利用try catch对含有小数、不用\t分隔的矩阵进行排除。



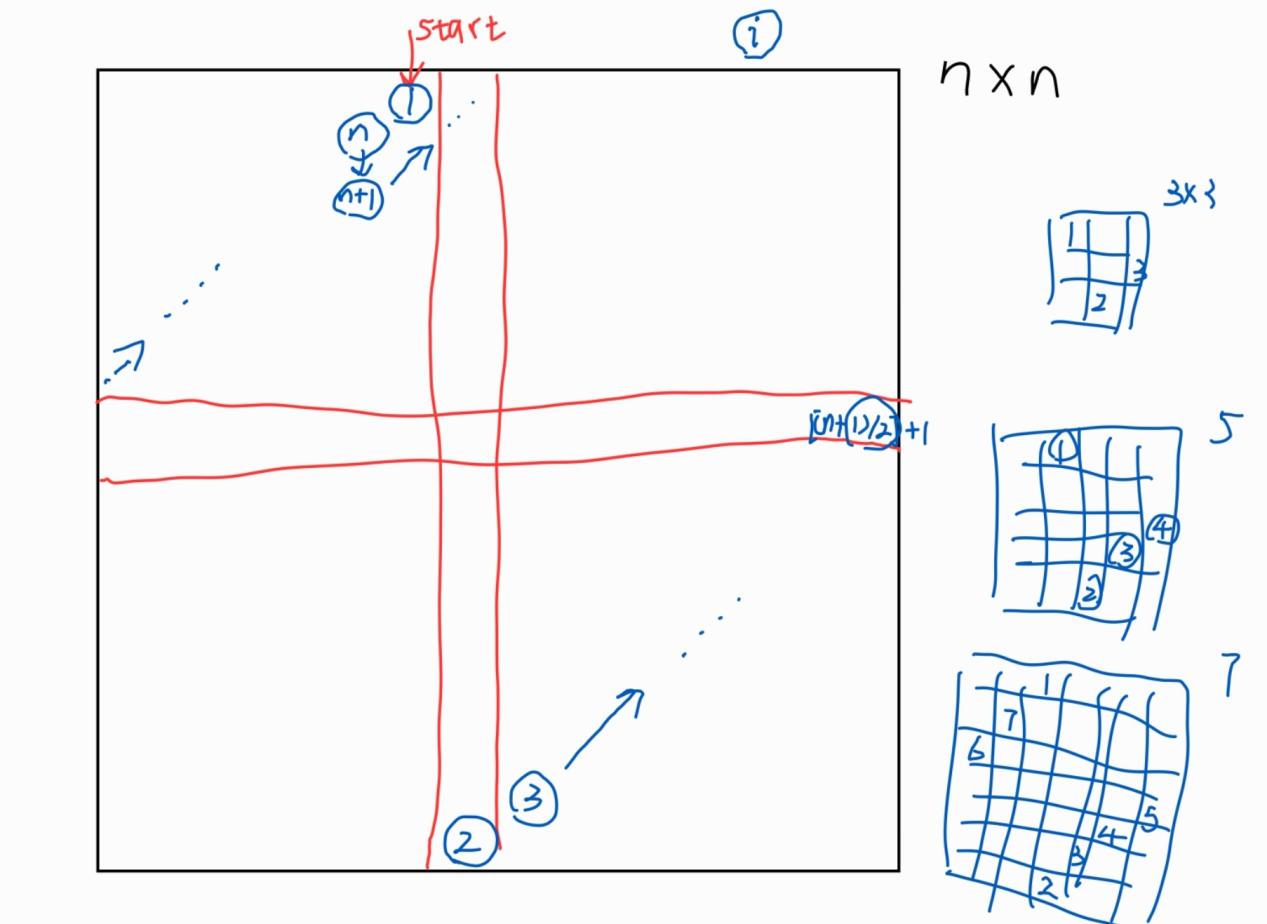
逐行检查每行数字之和是否相等，逐列和主次对角线上同理，先求和再比较，如图访问【i】【j】位置的数。

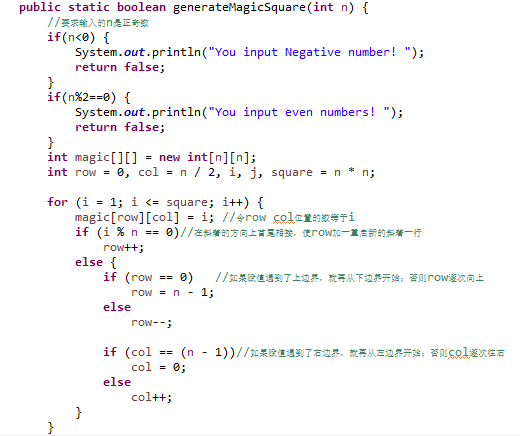
通过检查最后才返回true。



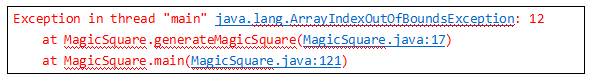
### generateMagicSquare()

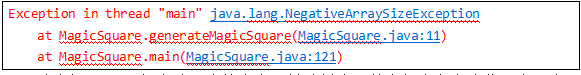
按步骤给出你的设计和实现思路/过程/结果。





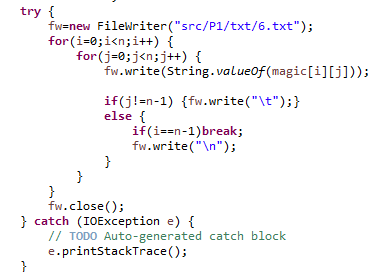
根据输入参数n，从1~n的中位数-1位置（图圆圈1）开始，斜着向右上填入从1开始的递增数列，并在遇到已经填入数的位置时往下移动。这样子每一行或每一列就在分成n份的nxn个数中的每一份都按这个规律取了一个数作为自己的行（列）元素，并有一定偏移使得累加和相等。这样子就得到了一个nxn的幻方。





当输入偶数时，在向斜右上方逐个输入时会遇到输入到最右下角并且i可以被n整除的情况，此时当row再加一数组就越界了。输入奇数就不会出现这种情况。

当输入负数时，在new一个数组的时候就会因为输入不合法而报错。



使用FileWriter逐个写入6.txt，用\t分割，每行最后直接换行，并在最末尾处直接结束。

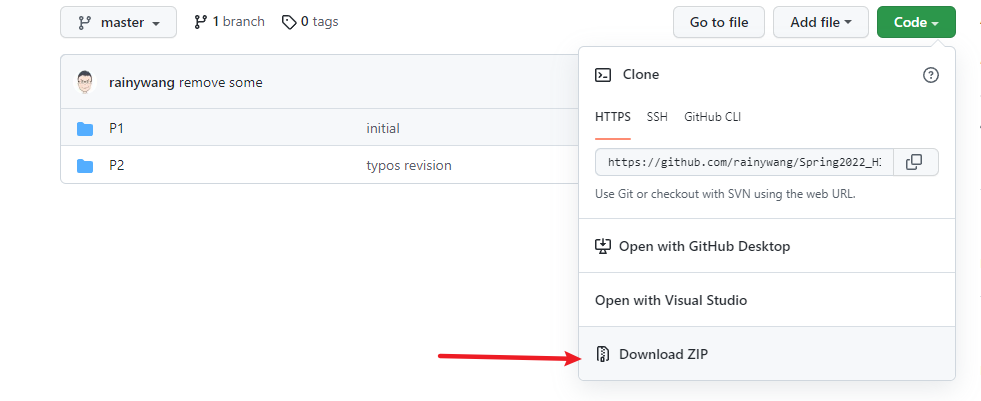
## Turtle Graphics

在这里简要概述你对该任务的理解。

### Problem 1: Clone and import

如何从GitHub获取该任务的代码、在本地创建git仓库、使用git管理本地开发。

<https://github.com/rainywang/Spring2019_HITCS_SC_Lab1/tree/master/P2>

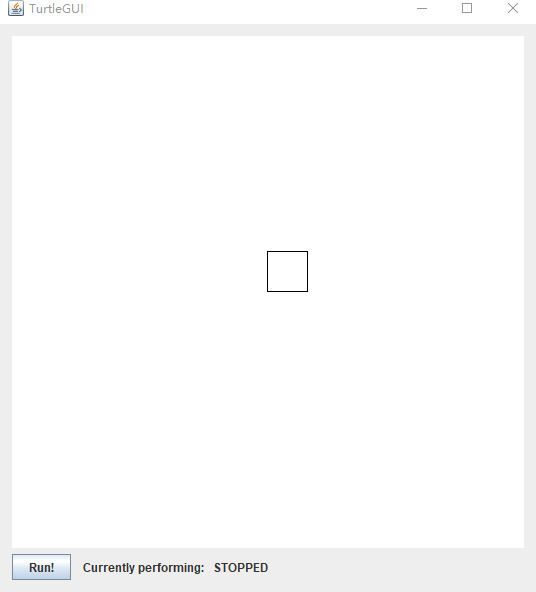


从这个地址获取含有P2的zip文件，并拷贝到本地src文件夹。

### Problem 3: Turtle graphics and drawSquare

使用turtle类的方法forward和turn来画一个正方形。





在点击Run！后可以看到画笔根据我们的代码画了一个正方形。这里简单地直接画-转-画-转-画-转-画在原地画了一个指定尺寸的正方形。

### Problem 5: Drawing polygons

这个问题要求绘制多边形，要求只编辑TurtleSoup.java。还可以添加测试TurtleSoupTest。

1. 实施calculateRegularPolygonAngle

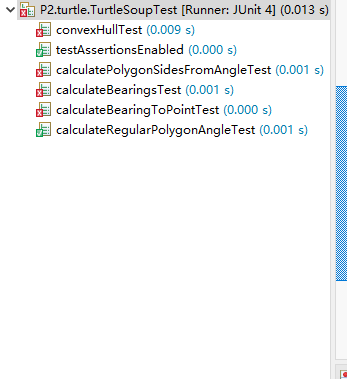
推导正多边形的内角应该是多少：

设正多边形边数为n，内角为r。因为正多边形外角和都是360度，所以有

1. r）\*n=360

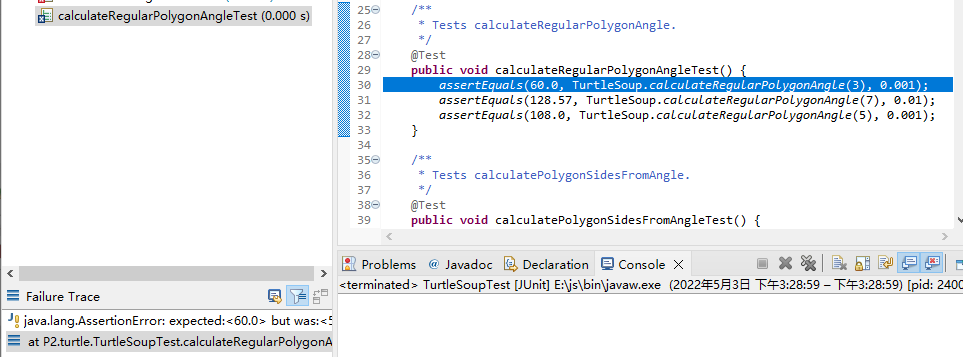
则 r=180-（360/n）

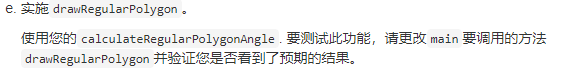
1. 运行JUnit测试程序测试刚刚实现的calculateRegularPolygonAngle方法是否正确



可以看到是正确的。

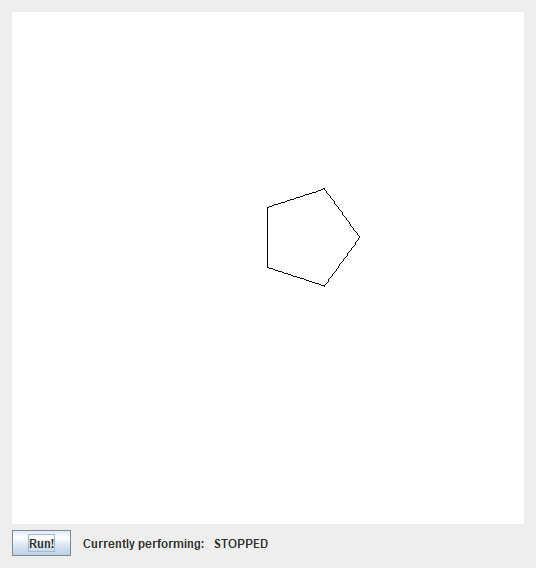
1. 破坏刚才实现的方法使其无法通过测试，认识了解如何根据测试代码查看错误出现的原因。



1. 修回实现的方法
2. 

实现绘制正多边形并在main中调用验证测试。

还是利用forward和turn，但是要注意的是turn需求的是正多边形的外角值，所以在得到内角后要与180度相减。

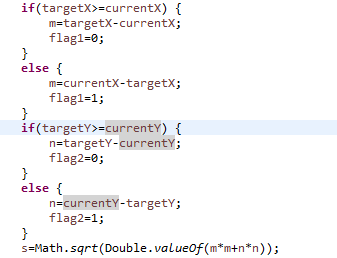


用此方法画出的正五边形。

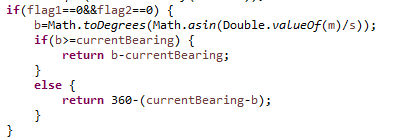
### Problem 6: Calculating Bearings

A.要求实现calculateBearingToPoint，参数是当前turtle面对的角度（以向上为极轴，逆时针为正方向）、turtle现在的坐标和目标坐标，计算出从现在坐标到目标坐标turn所需要转过的角度。

主要利用三角形正弦定理和反三角函数计算出弧度。



m是两点间水平距离，n是两点间垂直距离，s是m、n为直角边的斜边。在计算m和n时利用flag1和flag2来记录两点间位置关系，即以现在点为原点时，目标点所在象限。

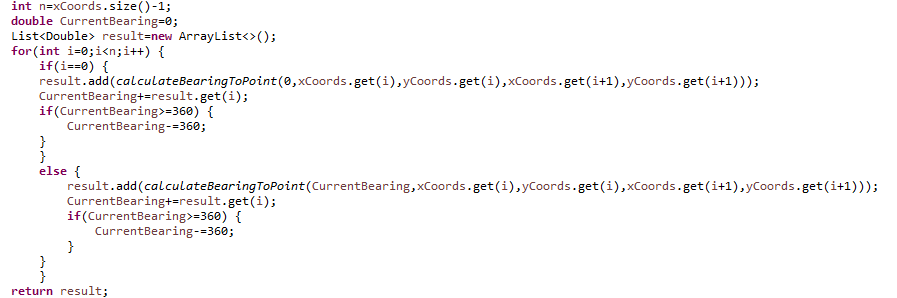


这里举目标点在第一象限时的例子，b是以现在点为原点、极轴向上、逆时针为正方向创建极坐标系时，目标点的极角。下面根据b和现在turtle面向角度的大小关系，返回不同的角度。

其他象限类似的道理。

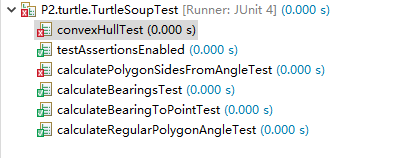
1. 要求实现calculateBearings，利用calculateBearingToPoint，根据输入的两个x、y坐标的列表来返回走过这个路径每步turn需要转过的角度的列表。

注意到返回的列表result长度为输入列表长度-1，并以此设置循环。



除了初始i=0时turtle面向上方即角度为0，后面的turtle的面向角度CurrentBearing为CurrentBearing加上上一次计算出的角度得到的结果。注意加上后超过360要减去。

1. 运行JUnit测试程序，使得前两个方法通过检查。

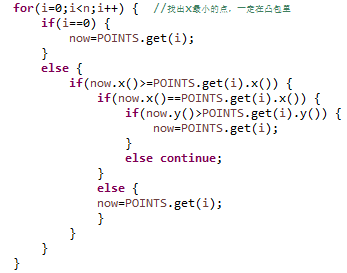


### Problem 7: Convex Hulls



实现计算凸包的算法，这里使用gift-wrapping algorithm算法，因为可以直接调用先前的calculateBearingToPoint。

首先找一个起始点，这里选择x值最小的点。如果同时有多个点x值最小则选其中y最小的点。

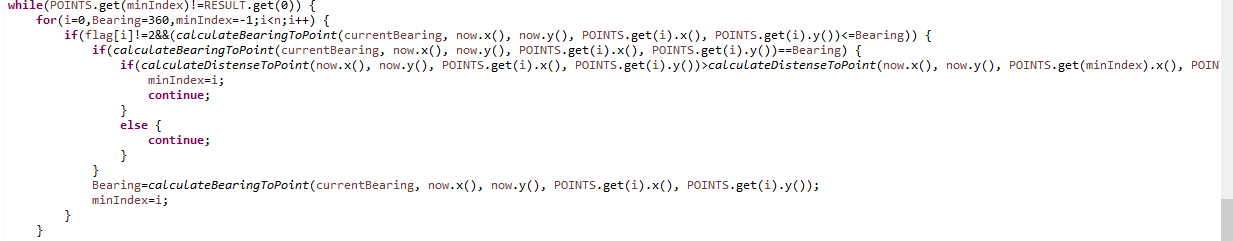


用标记数组flag标记为1，并将该点加入结果set。（flag初始为0）



然后调用calculateBearingToPoint遍历flag为0的点，找到偏转角最小的点。如果有同为偏转角最小的点则选择其中离原点较远的一点作为偏转角最小的点。

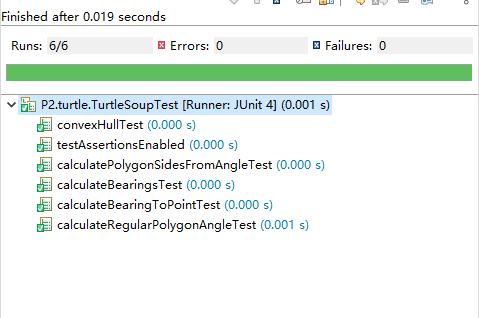
遍历结束后minIndex为偏转角最小的点的下标，令这个点位置的flag=2以和未放入结果set及初始点区别；然后令currentBearing=最小偏转角，因为要用到这个参数以用作以这个偏转角最小的点重新建系并做同样的操作。



就像用绳子把石头包起来一样找到结果set中的点，并以回到初始点作为循环结束的条件。

最后把List转成Set返回集合。

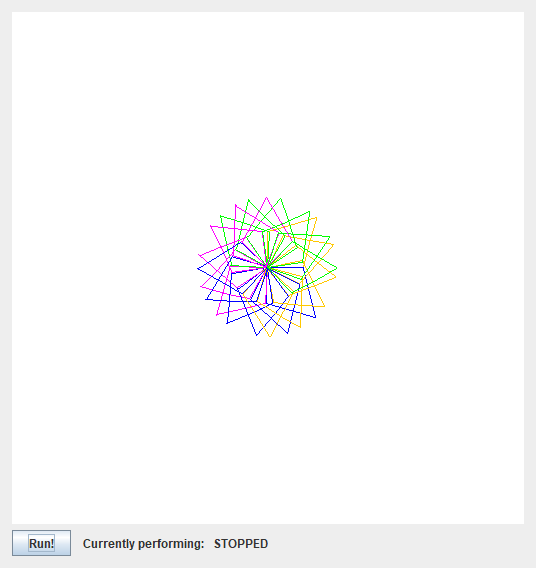




要求通过所有检查并出现大绿条。

### Problem 8: Personal art

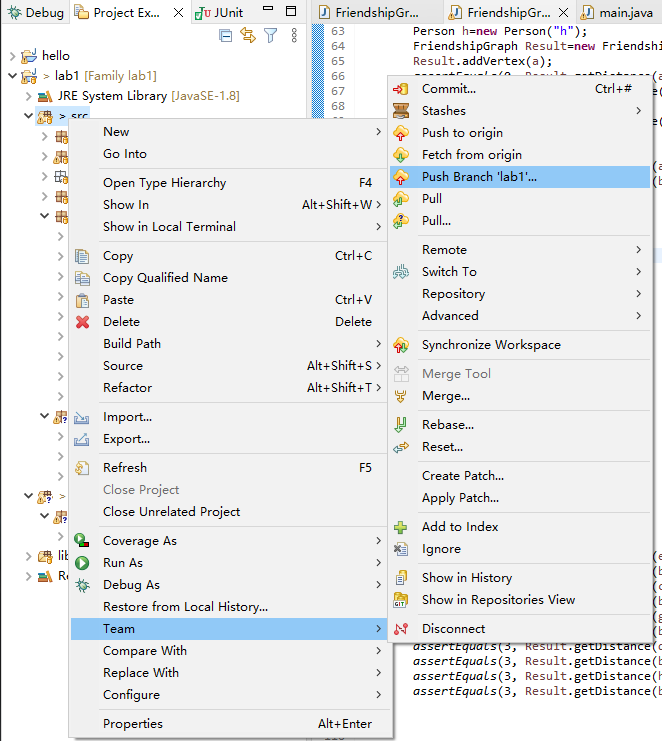
自由发挥画出自己的图案。



这里普通地在画了一个4种颜色的四角星的基础上用旋转画出了个图案。

### Submitting

如何通过Git提交当前版本到GitHub上你的Lab1仓库。



在绑定ssh密钥和配置好本地git后，在commit保存至本地库以后再推到仓库上

## Social Network

在这里简要概述你对该任务的理解。

任务要求实现一个无向图，结点为有名字属性的人，点之间的关联模拟社交网络。

还要实现添加成员，添加关系，计算距离等方法。以及编写测试这些方法的Junit。

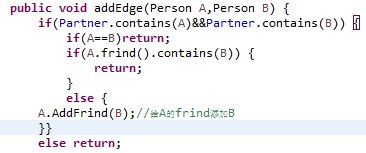
### 设计/实现FriendshipGraph类

给出你的设计和实现思路/过程/结果。

成员为一个Person的List，表示社交网络里面的人。

要实现的方法有添加成员addVertex，约束条件是每个人都是唯一的，所以在添加重名的人时会提示一个错误。

添加边addEdge，为了兼容有向图，所以是单向地添加。



如果A和B是同一个人或者其中有人不在Partner中则不添加边，否则在A的friend中添加B。

计算两个人之间的最短距离getDistance。鉴于是无权的无向图，使用BFS计算最短路径。



### 设计/实现Person类

给出你的设计和实现思路/过程/结果。

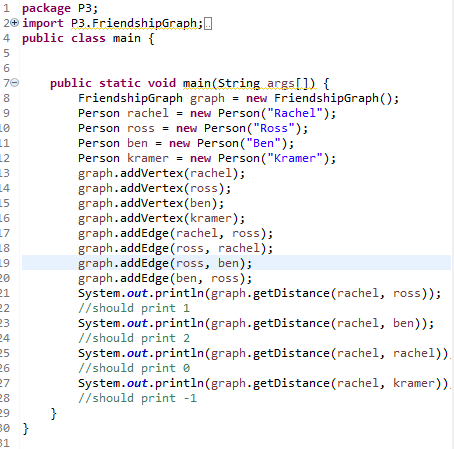
Person类有两个成员，名字Name和朋友Friend，其中Friend是Person的列表List，用来存放这个Person的朋友（指向）。

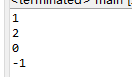
方法AddFriend用于添加朋友，相当于加边。（不重复）

### 设计/实现客户端代码main()

给出你的设计和实现思路/过程/结果。

在实现上述方法后成功运行以下代码：





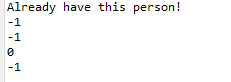
下为上述main的输出。

当注释掉第17行代码，输出变为



因为rachel到ross的联系不存在了，就到不了其他的三个点而返回-1；而到自己的距离默认都是0.

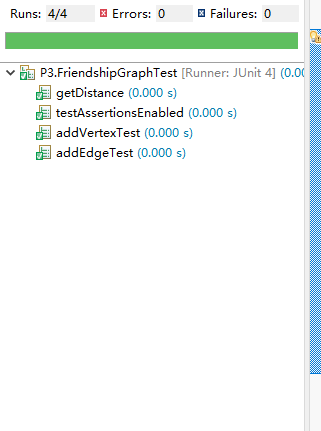
将第10行的Ross替换成Rachel，即出现了重名则输出为



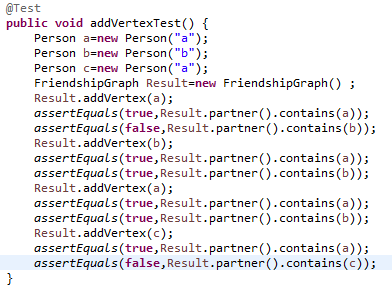
因为重名提醒了一个错误，然后并没有往graph里添加成功，自然边也没有添加成功。之后的输出也符合预期。

### 设计/实现测试用例

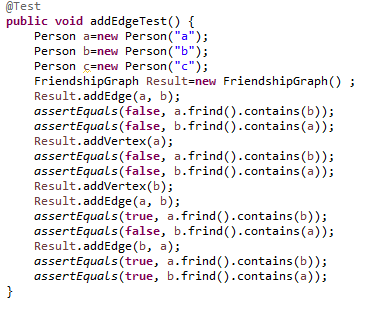
给出你的设计和实现思路/过程/结果。



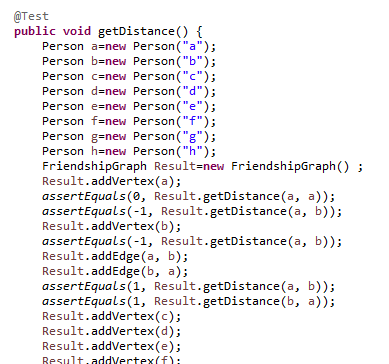
测试addVertex就利用List的contains方法来看是否添加成员成功。其中包括对重名情况的测试。

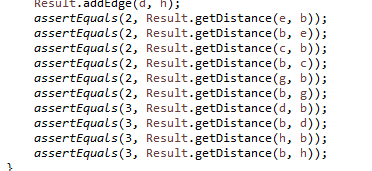


测试addEdgeTest原理同上。还考虑到在添加边之前点是否已被加入图中的情况。



测试getDistance，简单地在一个8顶点的无向图逐个添加顶点并测试getDistance正确性，并改变参数顺序以证明结果等价。





# 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 | 任务 | 实际完成情况 |
| 2022-05-2 | 9:30-15:30 | 编写问题1的所有问题并进行测试； | 按计划完成 |
| 5-3 | 9:30-15:30 | 编写问题2的所有问题并进行测试。 | 按计划完成 |
| 5-4 | 9:30-15:30 | 编写问题3的所有问题并进行测试。 | 按计划完成 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的难点 | 解决途径 |
| 幻方实验的错误报告复杂搞不懂 | 百度查询资料手册以及和同学交流 |
| 凸包问题难以通过测试 | 逐行debug调试找到问题所在 |
| Git配置不完全，推拉进行不顺利 | 百度完整配置流程并慢慢配置尝试 |

# 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

实验过程中体会到过去课程中学习到的编程技巧、数据结构有多么重要。增加了编写完整小程序的经验，并更懂得利用手册、搜索引擎为编程提供帮助。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

还行。还是有很多C++的味道在里面。

1. 关于Eclipse IDE

比较亲密顺手。

1. 关于Git和GitHub

网速经常太慢连不上，其他方面很优秀。

1. 关于CMU和MIT的作业

比较有趣，深入浅出。

1. 关于本实验的工作量、难度、deadline

难度不高但是量可能有些大，deadline有些紧。

1. 关于初接触“软件构造”课程

身体力行亲身体验了软件开发的完整过程，受益匪浅。