

**2022年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 陈本福 |
| 学号 | 1180300526 |
| 班号 | 1903401 |
| 电子邮件 | [marlon\_chan19@outlook.com](https://www.outlook.com/?refd=account.microsoft.com&fref=home.banner.viewinbox) |
| 手机号码 | 13304515838 |

**目录**

[1 实验目标概述 1](#_Toc29325521)

[2 实验环境配置 1](#_Toc29325522)

[3 实验过程 1](#_Toc29325523)

[3.1 Magic Squares 1](#_Toc29325524)

[3.1.1 isLegalMagicSquare() 2](#_Toc29325525)

[3.1.2 generateMagicSquare() 2](#_Toc29325526)

[3.2 Turtle Graphics 5](#_Toc29325527)

[3.2.1 Problem 1: Clone and import 5](#_Toc29325528)

[3.2.2 Problem 3: Turtle graphics and drawSquare 5](#_Toc29325529)

[3.2.3 Problem 5: Drawing polygons 6](#_Toc29325530)

[3.2.4 Problem 6: Calculating Bearings 7](#_Toc29325531)

[3.2.5 Problem 7: Convex Hulls 7](#_Toc29325532)

[3.2.6 Problem 8: Personal art 9](#_Toc29325533)

[3.2.7 Submitting 9](#_Toc29325534)

[3.3 Social Network 10](#_Toc29325535)

[3.3.1 设计/实现FriendshipGraph类 10](#_Toc29325536)

[3.3.2 设计/实现Person类 11](#_Toc29325537)

[3.3.3 设计/实现客户端代码main() 12](#_Toc29325538)

[3.3.4 设计/实现测试用例 13](#_Toc29325539)

[4 实验进度记录 15](#_Toc29325540)

[5 实验过程中遇到的困难与解决途径 16](#_Toc29325541)

[6 实验过程中收获的经验、教训、感想 16](#_Toc29325542)

[6.1 实验过程中收获的经验和教训 16](#_Toc29325543)

[6.2 针对以下方面的感受 16](#_Toc29325544)

# 实验目标概述

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

1. 基本的 Java OO 编程
2. 基于 IDEA IDE 进行 Java 编程
3. 基于 JUnit 的测试
4. 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。

在这里给出你的GitHub Lab1仓库的URL地址。

https://github.com/ComputerScienceHIT/HIT-Lab1-1180300526.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

任务一：从五个 txt 文件中读取数据，判断是否可以构成magic square，如果是就返回true，否就返回false，

任务二：输入一个数字，然后对其判断是否为MagicSquare。如果是奇数并且非负，则将数字传进generateMagicSquare，形成一个相应的矩阵，并且保存到6.txt 。然后对其判断是否为MagicSquare。

### isLegalMagicSquare()

按步骤给出你的设计和实现思路/过程/结果。

**设计思路**：从左往右，从上往下遍历文件存储的矩阵。另外建立一个2n+2大小的数组，分别存储每一行、每一列、以及对角线上的数据和。每遍历到一个数字，就把数据加载到数组对应的位置，在最后遍历数组存储的每个数字是否都相等，就可以判断是否是magic square。

**设计过程**：

1. 行列数不相等、并非矩阵：先算出行数，然后重新遍历每一行，判断每一行的数字个数是否等于行数，如果不相等则返回false，并输出错误原因，相等返回true。
2. 矩阵中某些数字不是正整数：按‘\t‘分割字符串，并且存入一个字符串数组，判断每个字符串里面是否含有’-‘或者’.‘，就可以判断数字是否是正整数了。
3. 不是以\t作为分隔符：若不是以\t作为分隔符，则会在Integer.valueOfs时抛出异常信息，并返回false。

**结果：**

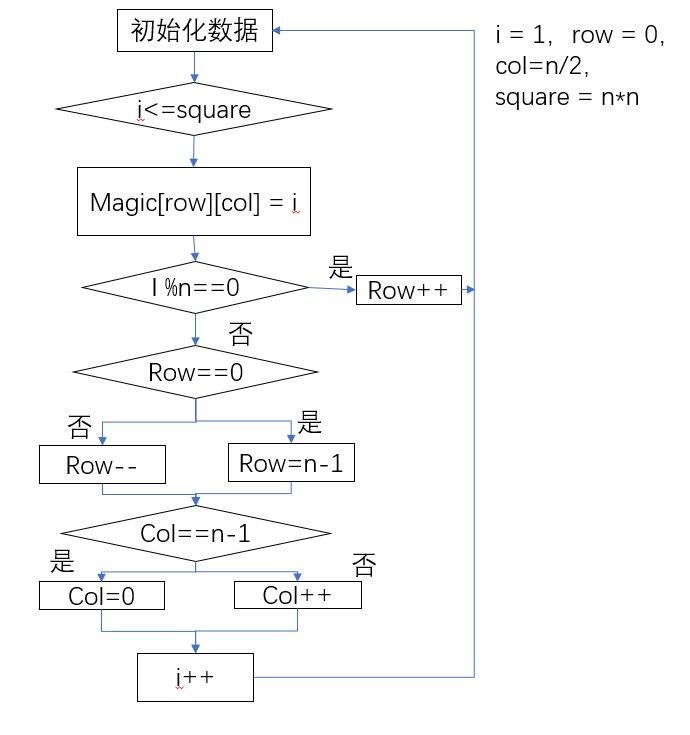


### generateMagicSquare()

按步骤给出你的设计和实现思路/过程/结果。

**实现思路：**

**流程图如下：**

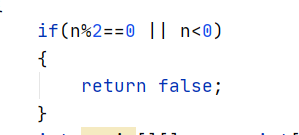


(1) 将产生的magic square写入文件\src\P1\txt\6.txt中；

(2) 当输入的n不合法时（n为偶数、n为负数等），不要该函数抛出异常并非法退出，而是提示错误并“优雅的”退出——函数输出false结束。

**过程：**

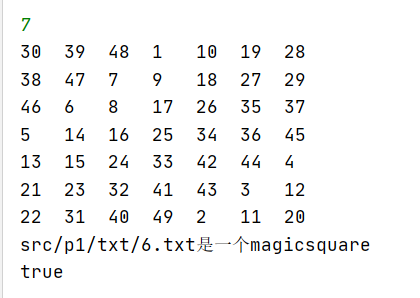
**先判断是否是合法输入：**



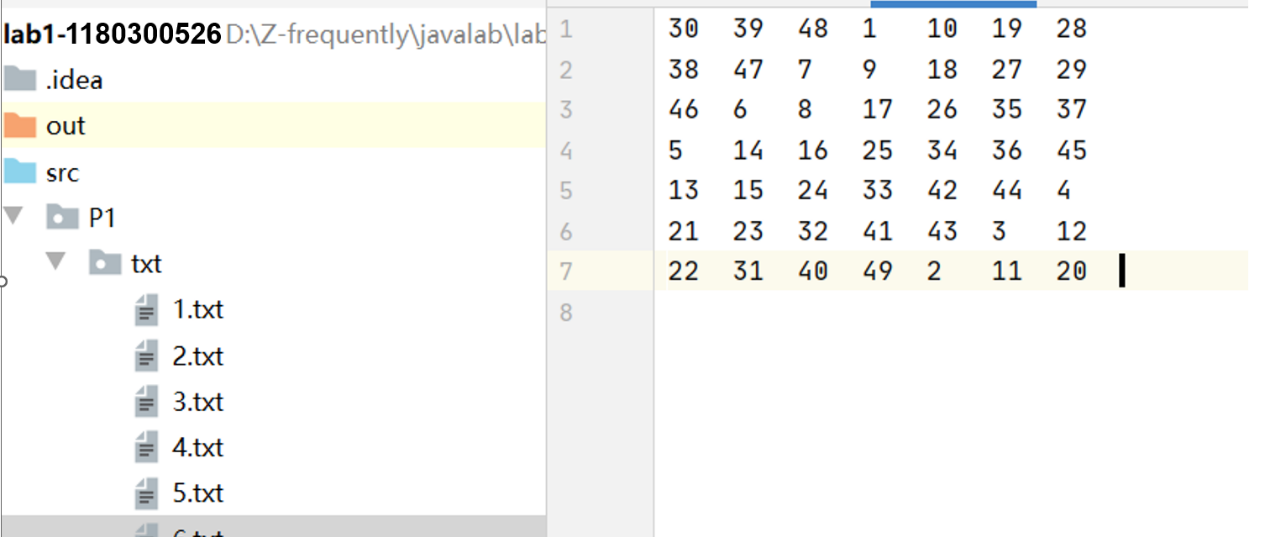
**接下来将得到的矩阵存进6.txt文件**



**结果实现：**





****

## Turtle Graphics

在这里简要概述你对该任务的理解。

这次任务是为了实现一个绘图工具Trurtle Graphics，需要我们完成的任务为：画出一个正方形，通过边计算正多边形内角，由正多边形内角得到边数，计算从一个点到另外一个点所需要转过的角度，给定一系列点计算每次转过的角度并返回在集合中，凸包suanfa1，绘制图形，junit进行单元测试。

### 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。

根据实验手册上所给的地址即可下载本次任务代码，

建立git仓库：git init

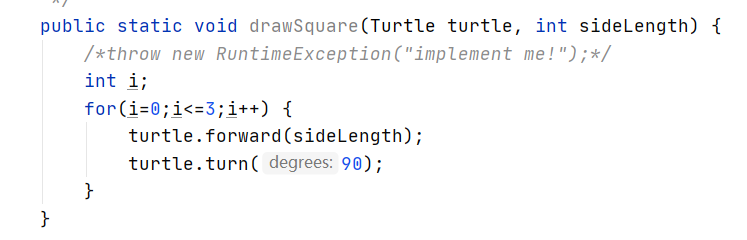
管理本地开发：

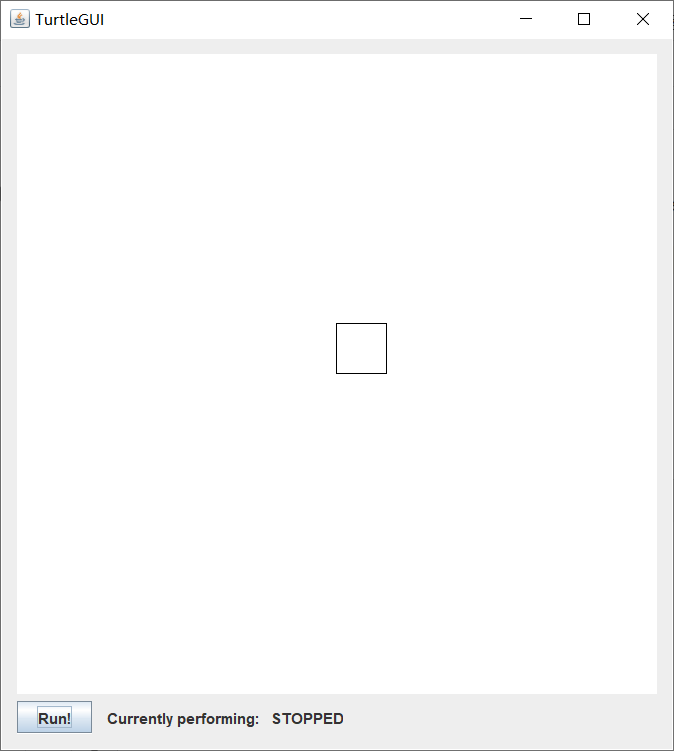
Git add + 文件 将文件加载到工作区

Git commit -m + “文件注释” 上传到仓库

### Problem 3: Turtle graphics and drawSquare

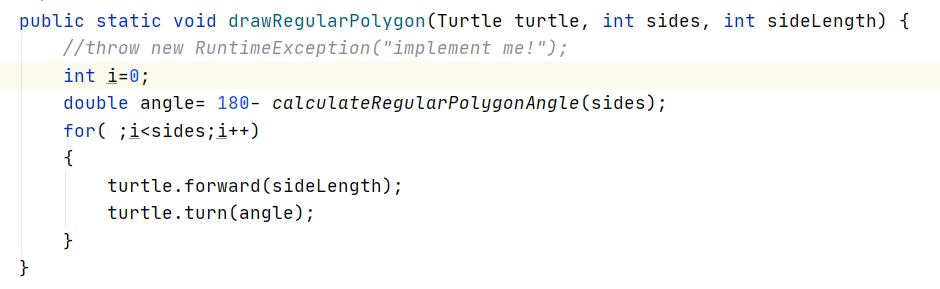
如下图，画一个正方形，循环四次，角度每次改变90，长度固定即可画出一个正方形

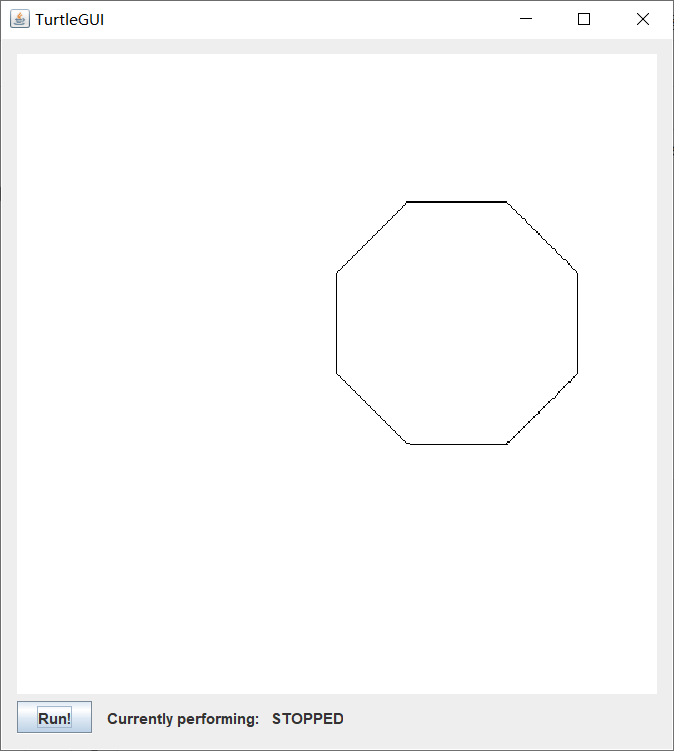




### Problem 5: Drawing polygons

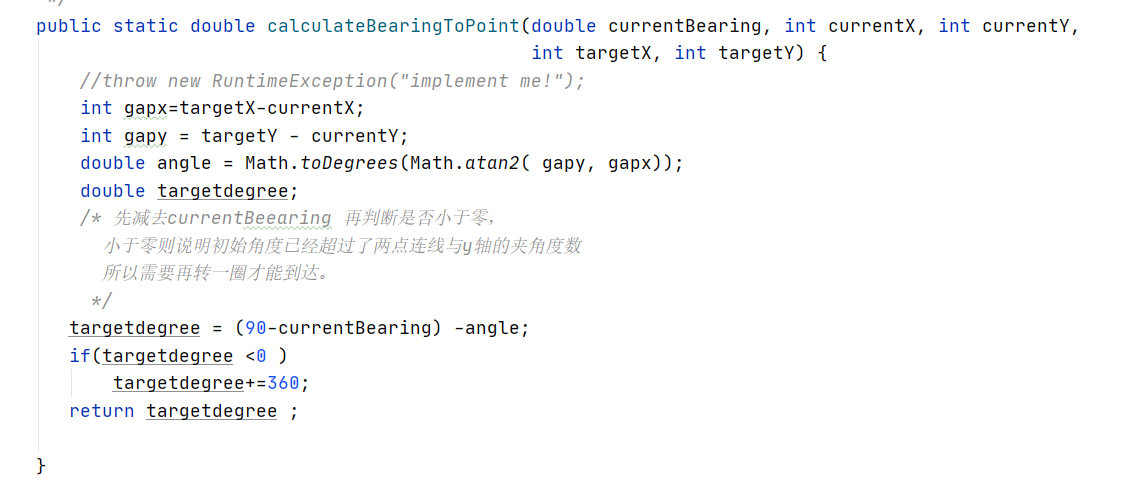
画一个正则多边形，由于我们可以根据之前已经实现了的函数计算每次偏移的角度，就可以知道内角为多少，然后每次画出来的长度固定就可以画出一个正则多边形



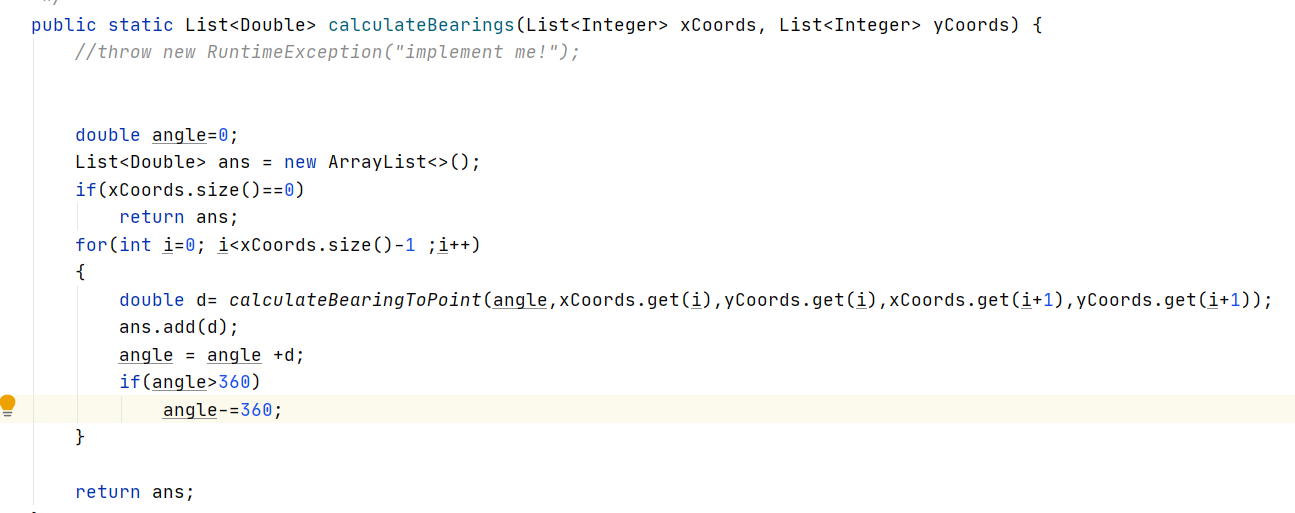


### Problem 6: Calculating Bearings

计算两点之间的所要偏移的角度必须注意不能大于小于0，小于0就加上360即可，因为刚开始有个偏移角度，所以要在初始位置90度-偏移角度。根据公式推导得出代码：

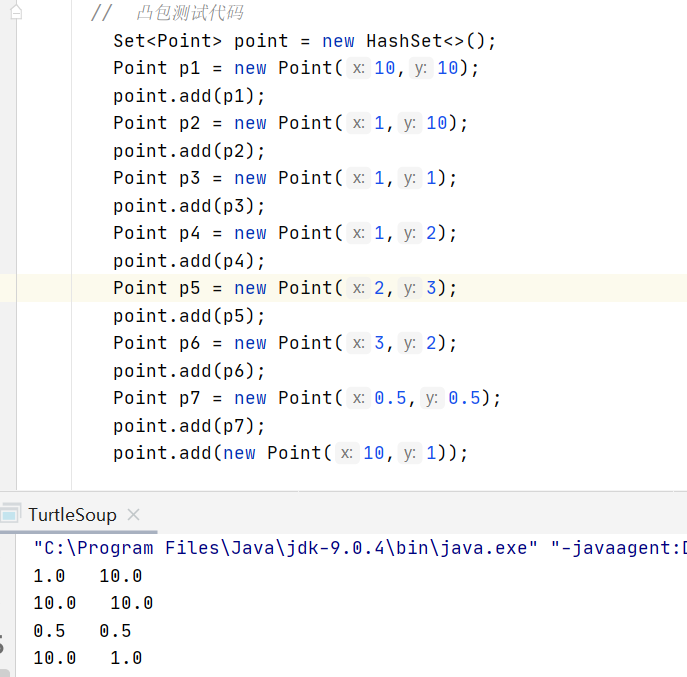


任意列表中的两点之间的度数如下：

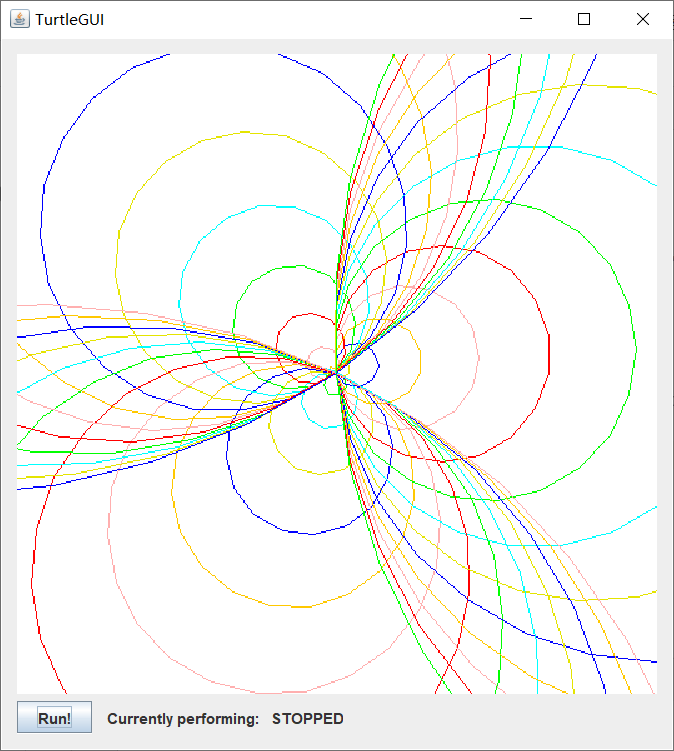


### Problem 7: Convex Hulls

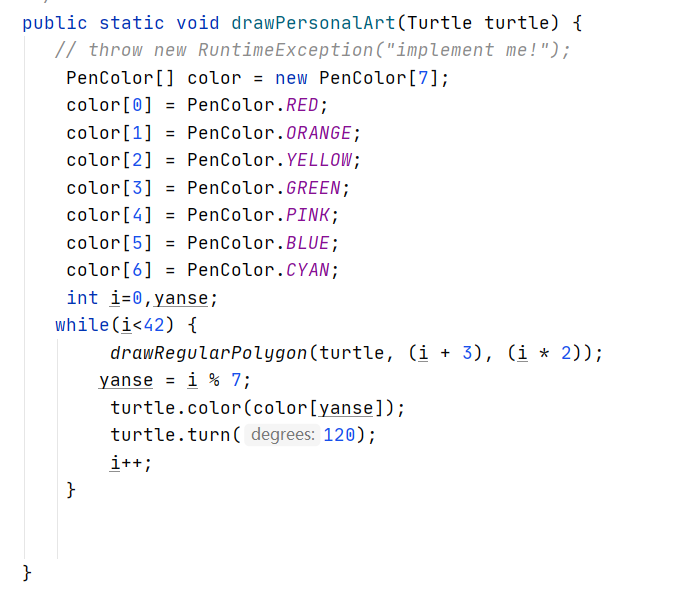
凸包算法，先找到最左下角点，然后根据之前实现了的函数calculating bearings计算每次最小偏移角对于的另外一个点，然后以该点为起点，重复操作，就可以得到一个点集，该点集就是最终结果，运行如下：



### Problem 8: Personal art



实现比较简单：每次枚举，改变颜色以及正则多边形的大小。



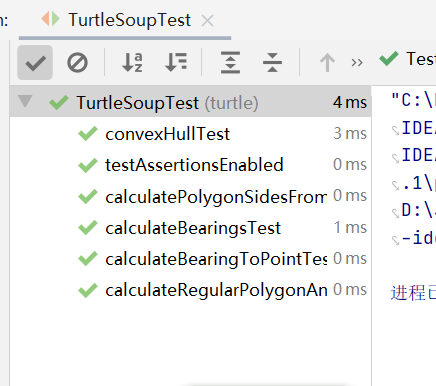
### Submitting

如何通过Git提交当前版本到GitHub上你的Lab1仓库。

先存储到本地仓库，然后使用命令行

Git push 可以传送到Github仓库。

**测试结果：**



测试通过

## Social Network

该部分就是写一遍搜索算法，难度并不大，只是还需要多实现一个Person类。

### 设计/实现FriendshipGraph类

给出你的设计和实现思路/过程/结果。

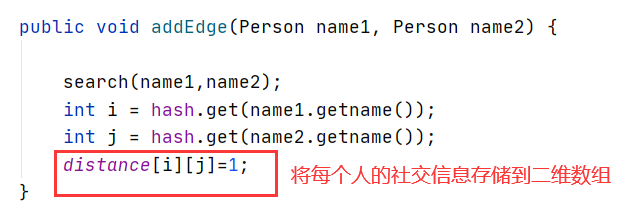
思路：先设计一个Person类，用来表示每个人的名字，然后建立一个hash表用来存储每个被放入Graph的人的信息，并且hash表也使得每个人都是独一无二的。最后当加入每个人与他人的边时候，再建立一个二维数组，用hash表中每个人的编号对应二维数组中的位置，这样我们就可以轻易的构造处出每个人在Graph中的信息。

过程：

第一步，将每个Person放入hash表，并且建立编号

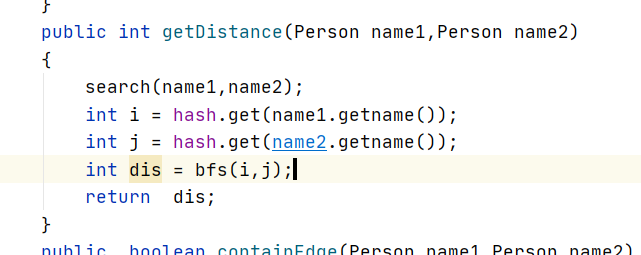


第二步：



这样我们就构造好了社交图。我们可以轻易地知道每个人与他人的关系。

第三部：计算两个人之间的距离，就是一个搜索算法

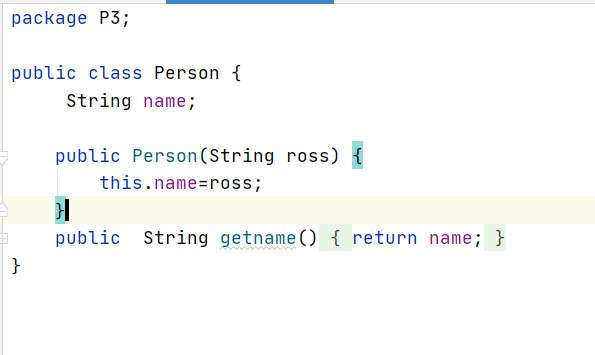


搜索算法为BFS，未截图。

### 设计/实现Person类

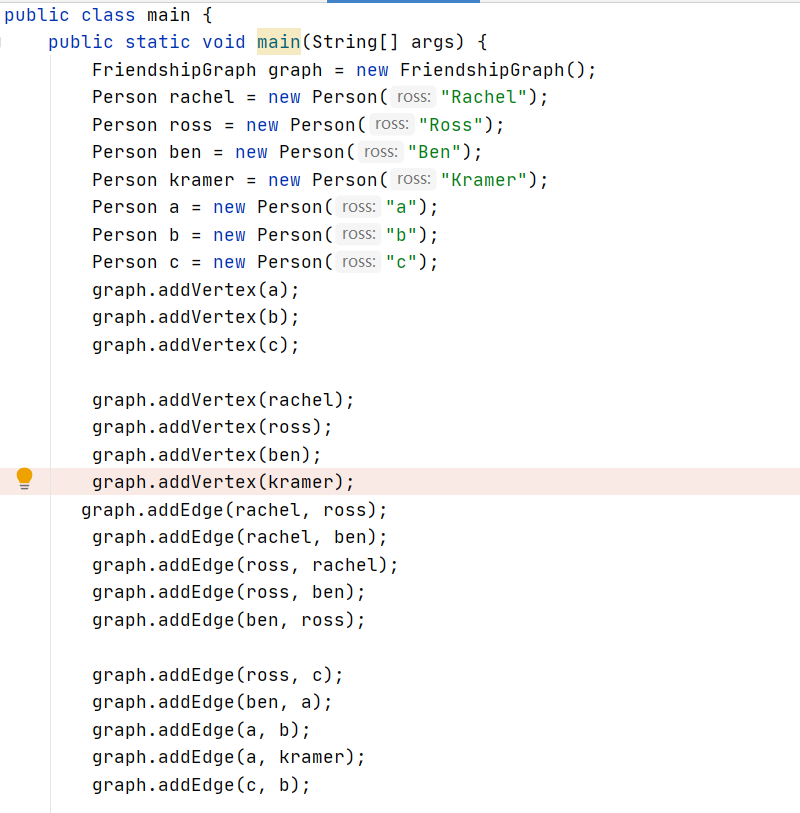
给出你的设计和实现思路/过程/结果。

Person类设计较为简单，并没有用来存储太多东西，仅仅用来存储姓名。



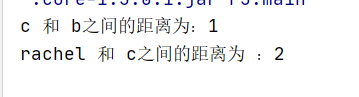
### 设计/实现客户端代码main()

给出你的设计和实现思路/过程/结果。



Main函数主要是用来测试，通过画图，来确认算法的正确性，并没有太多的设计流程。

查看结果：



### 设计/实现测试用例

给出你的设计和实现思路/过程/结果。

1. **AddVertextest 测试**



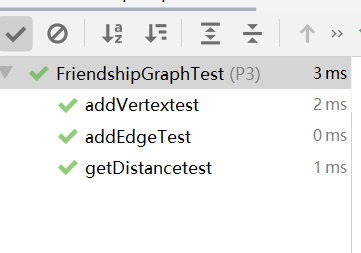
1. **AddEdgeTest**



1. **getDistancetest**



**测试结果：**



# 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 |
| 2022-05-14 | 8.00-21.00 | 编写P1 和P2 | 按计划完成 |
| 2022-05-19 | 14.00-16.00 | 编写P3 | 按计划完成 |
| 2022-5-20 | 21.00-23.00 | 写实验报告 | 按计划完成 |
| 2022-5-23 | 18.00-21.00 | 实验交报告 | 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 |
| java语言不通 | 虽然以前也用过java刷题，但是都是算法题并没有用到多少java的各种类，接口的知识，所以在实验中还得重新去学 |
| Junit 调试出现问题 | 重新下载，配置路径，解决问题 |
|  |  |

# 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

Java语言是面向对象的语言，以前对这句话一直都是靠记忆去记忆，而不是通过理解性的记忆，通过此次实验才真正明白了这句话的意思，对java语言使用的更加熟练。并且学会了实验Github去管理自己的代码。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

是，java比C语言好用多了，有很多已经实现了功能，不需要自己再去写一遍，降低了工作量，比如hashset 。

1. 关于Eclipse IDE；

并未使用Eclipse ，使用的是IDEA，很好用，下次还会再来

1. 关于Git和GitHub；

Git 一直下载不来仓库里面的文件，即使使用了VPN也是一直超时，所以就自己手动添加到仓库去。

Github 看起来还不错，使用也很方便

1. 关于CMU和MIT的作业；

第一题难度不大，仅仅只是对文件操作知识，很简单就可以完成。

第二题分的步骤比较多，考的知识点方面也很多，接口，实现类，实现类对象，枚举，画图，算正则多边形的角度，以及边数，最后是凸包算法。

第三题感觉是最简单的一题，也就是一个bfs算法，没啥难度。

1. 关于本实验的工作量、难度、deadline；

工作量一般，就是第二题工作量较大，文件较多，刚开始有点迷糊。

难度也并不大，时间很充足。

1. 关于初接触“软件构造”课程；

感觉软件构造实验比较有趣，通过自己设计代码去实现一些功能，比如画图，算magicsquare 都很有趣。虽然刚开始java语言不太通，但是也都是一些基本的用法，只要去学一下就可以解决问题。