

**2020年春季学期  
计算机学院《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 冯开来 |
| 学号 | 1190201215 |
| 班号 | 1903007 |
| 电子邮件 | [3223200086@qq.com](mailto:3223200086@qq.com) |
| 手机号码 | 13771513877 |

**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() 1](#_Toc29325525)

[3.1.2 generateMagicSquare() 1](#_Toc29325526)

[3.2 Turtle Graphics 1](#_Toc29325527)

[3.2.1 Problem 1: Clone and import 2](#_Toc29325528)

[3.2.2 Problem 3: Turtle graphics and drawSquare 2](#_Toc29325529)

[3.2.3 Problem 5: Drawing polygons 2](#_Toc29325530)

[3.2.4 Problem 6: Calculating Bearings 2](#_Toc29325531)

[3.2.5 Problem 7: Convex Hulls 2](#_Toc29325532)

[3.2.6 Problem 8: Personal art 2](#_Toc29325533)

[3.2.7 Submitting 2](#_Toc29325534)

[3.3 Social Network 2](#_Toc29325535)

[3.3.1 设计/实现FriendshipGraph类 2](#_Toc29325536)

[3.3.2 设计/实现Person类 2](#_Toc29325537)

[3.3.3 设计/实现客户端代码main() 2](#_Toc29325538)

[3.3.4 设计/实现测试用例 3](#_Toc29325539)

[4 实验进度记录 3](#_Toc29325540)

[5 实验过程中遇到的困难与解决途径 3](#_Toc29325541)

[6 实验过程中收获的经验、教训、感想 3](#_Toc29325542)

[6.1 实验过程中收获的经验和教训 3](#_Toc29325543)

[6.2 针对以下方面的感受 3](#_Toc29325544)

# 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 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

首先在lab0的实验报告中根据给出的网址下载安装了ecilpse，在git官网下载安装了git-bash，然后下载了jdk13。下载了junit4.13和hamcrest-core-1.3的jar包，由于git的操作并不熟悉，所以在csdn上查找了关于用git的操作与github进行联系的一些介绍，详见参考文章：https://blog.csdn.net/qq\_37512323/article/details/80693445

安装eclipse的过程见参考文章：https://blog.csdn.net/woniuxyy/article/details/80283635。后来在配置环境变量时不知道怎么做，教程好像少了一点讲解，然后又去搜索了关于如何配置环境变量的方法，详见参考文章：https://jingyan.baidu.com/article/ed15cb1bacb6a81be2698110.html

在这里给出你的GitHub Lab1仓库的URL地址（Lab1-学号）。

https://github.com/ComputerScienceHIT/HIT-Lab1-1190201215

# 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！）。

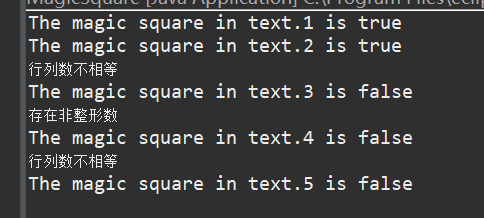
为了条理清晰，可根据需要在各节增加三级标题。

## Magic Squares

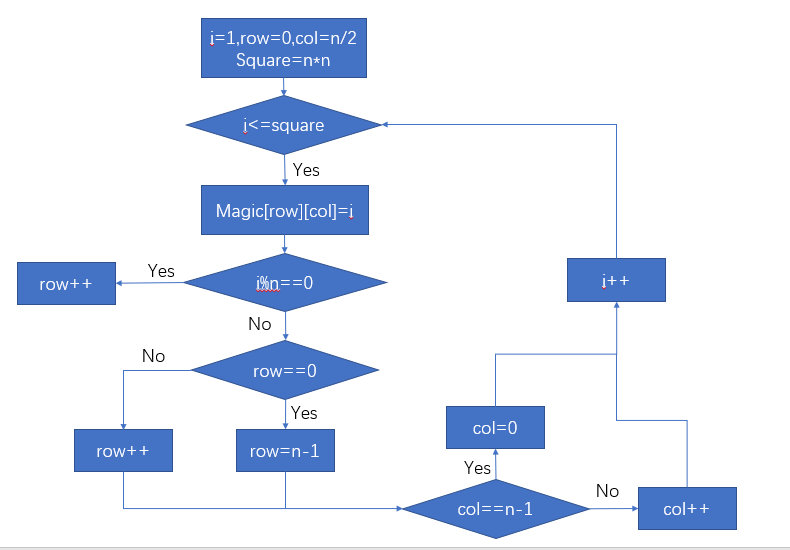
1. 能够读取五个.txt文件。
2. 判断这些文件是否构成MagicSquare，若是返回true，若不是返回false并且说明原因。MagicSquare是一个行列数相等，以\t间隔字符的矩阵，它的每一行、每一列、两个对角线之和是相等的。
3. 构造generateMagicSquare函数，使之产生一个MagicSquare并写入文件6.txt。
4. 然后判断6.txt是否是MagicSquare，如果输入的是偶数或者负数会返回false。

### isLegalMagicSquare()

1. 函数读入文件的数据存储在二维数组中，首先判断行列数是否相等；其次判断数据中是否有非法数字，比如浮点数；最后判断不是以\t作为分隔符的。对三种错误分别进行判断并输出返回false。如果没有这些错误时，分别计算矩阵每一行、每一列、两条对角线的和是否相等来判断能否构成MagicSquare，并返回true或者false。
2. 行列数不相等：我首先定义一个变量在读入每一行时进行加一的操作计算总行数，然后对每一行按照\t进行分割然后计数得到列数。如果列数不等于行数，则返回false。
3. 元素不是整型数：首先按照\t将分割后的字符串装入一个二维数组，判断这些字符串是否包含“.”、“-”，若包含则返回false。
4. 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。并返回false。
5. 判断是否每一行、每一列、对角线之和相等：首先利用第二种错误判断过程中得到的二维数组，然后计算每行每列对角线之和是否相等。若都相等，则返回true，否则返回false。但是很神奇的是，如果不是以\t为分隔符的，实质上会导致计算的行列数不相等，所以它会提前返回行列数不相等。
6. 得到结果：

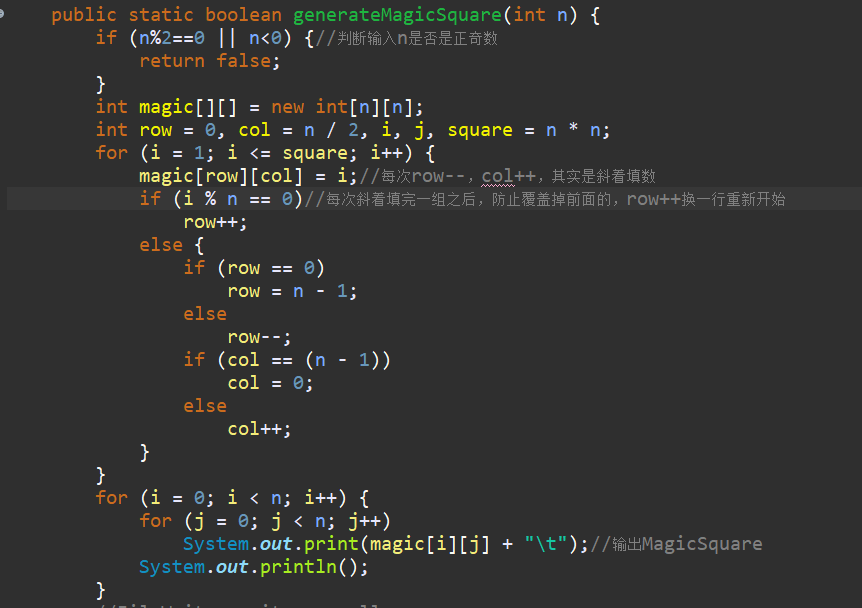


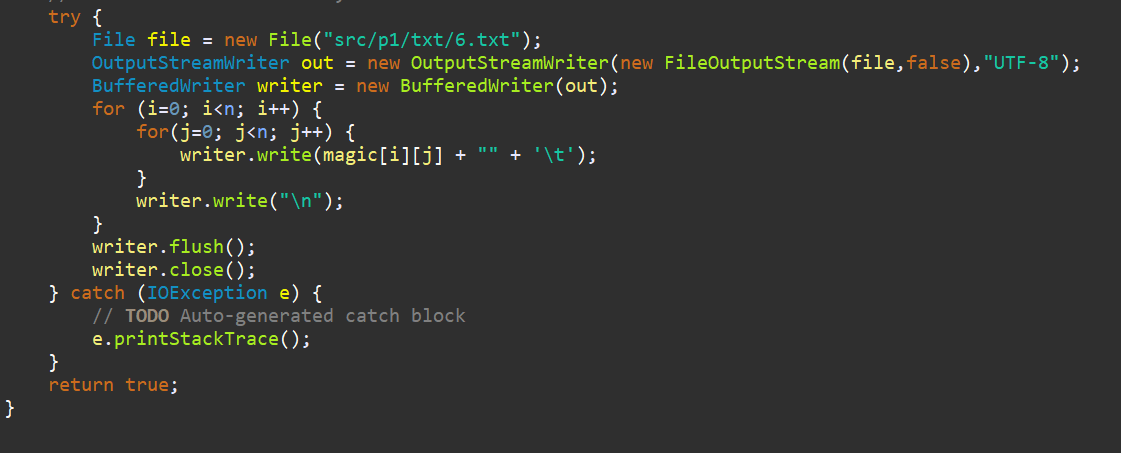
### generateMagicSquare()

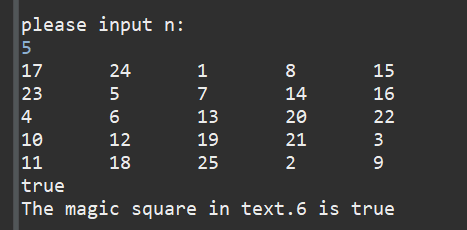
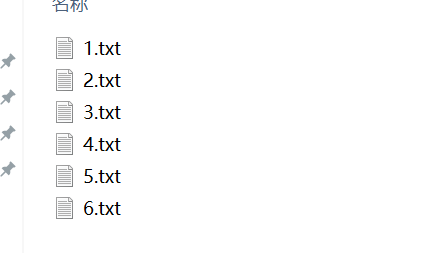
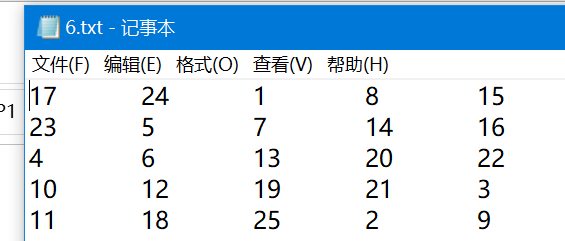


指导书已经给出了函数，我们只要增加它的功能：

1. 判断输入的n是否是正奇数，若不是则返回false而不是抛出异常。
2. 将产生的MagicSquare写入6.txt中，并判断是不是MagicSquare







## Turtle Graphics

制作一个绘图工具：

1. 画出一个正方形。
2. 由边数计算正多边形内角；由内角计算正多边形边数。
3. 计算一个点到另一个点需要转过的角度。
4. 给定一系列点，计算每次转过角度。
5. 定义色彩。
6. 自己绘画图形。
7. 实现使用Junity进行测试。

### Problem 1: Clone and import

创建git仓库的具体步骤当时并没有截图，大概步骤为：

1.获取秘钥： ssh -keygen -t rsa -C "your\_email@youremail.com"；

2.找到秘钥并复制添加到github上；

3.ssh –T git@github.com;

4.git config –global user.name ”Carlo”;

5.git config –global user.emal ”3223200086@qq.com”;

管理本地：

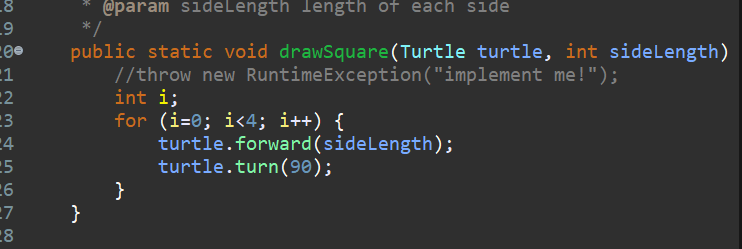
1. git add . Lab1-1190201215

2. git commit -m " srctest" ;

3. git push

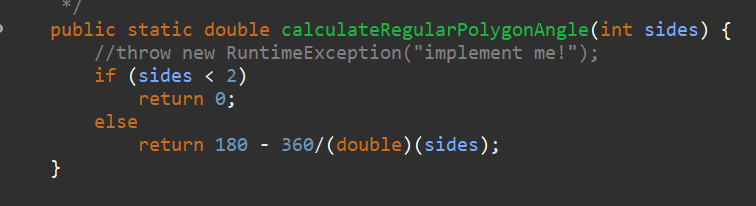
### Problem 3: Turtle graphics and drawSquare

每次转弯90°，共转4次。

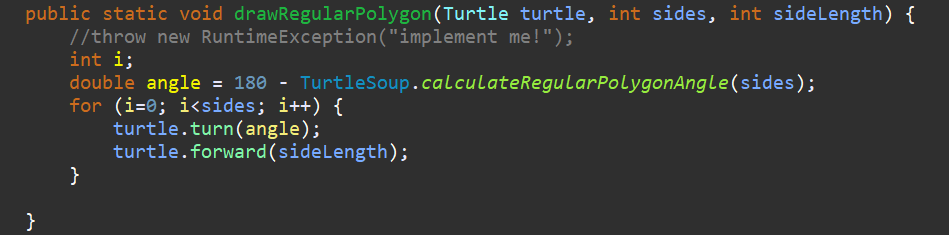


### Problem 5: Drawing polygons

由边数计算内角 180 – 360/边数

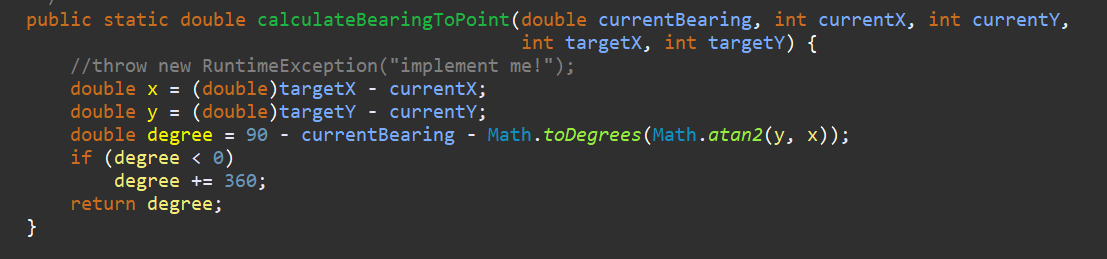


然后调用计算内角的函数可以绘制正多边形

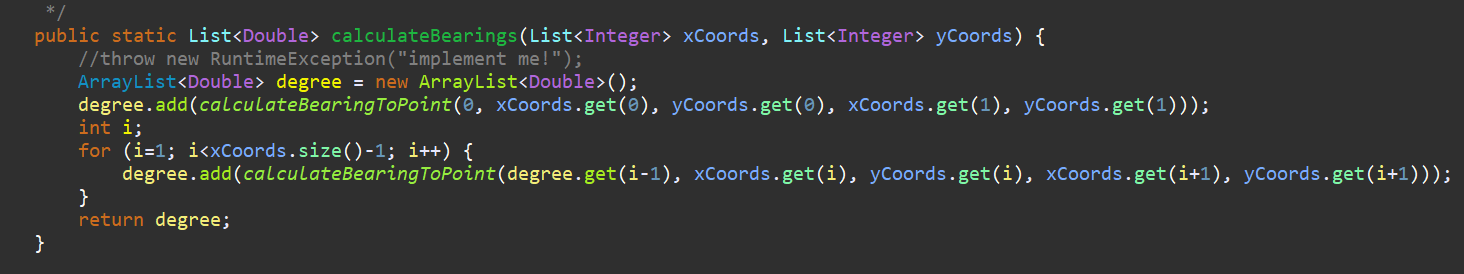


### Problem 6: Calculating Bearings

计算一个点到另一个点偏移多少度。调用Math.atan()得到需要旋转的弧度，然后调用toDegree转换成角度，如果度数为负，则加上360°。



计算一系列点的旋转偏移量。计算出(0,0)到第一个点的偏移角度，然后不断调用上述函数直到结束。



### Problem 7: Convex Hulls

Gift wrapping Algorithm

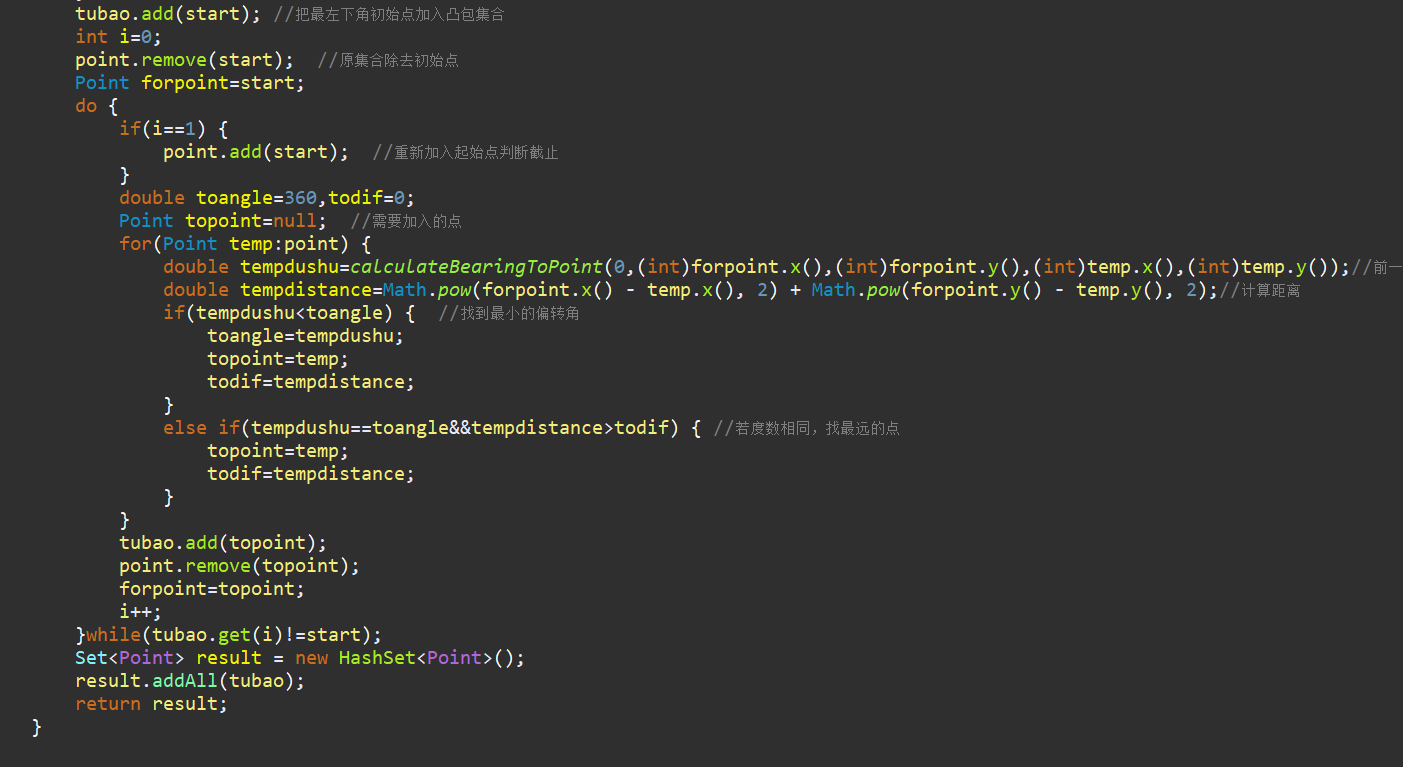
找到最左下角的点加入集合

比较剩余点到此点的偏转角

找到最小的偏转角加入集合，当偏转角相同的时候找最长的一条边的点加入集合

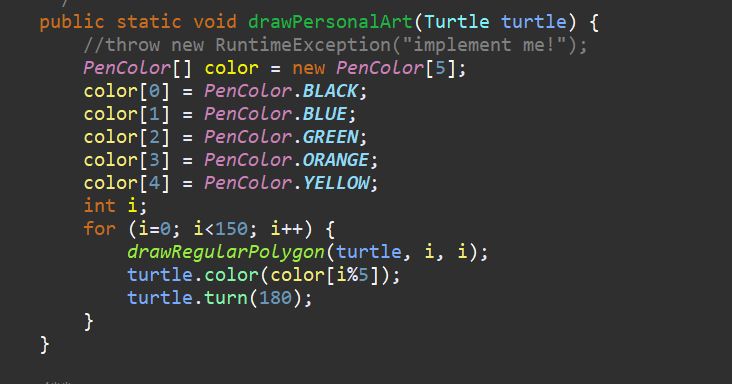
最后得到的集合为凸包。

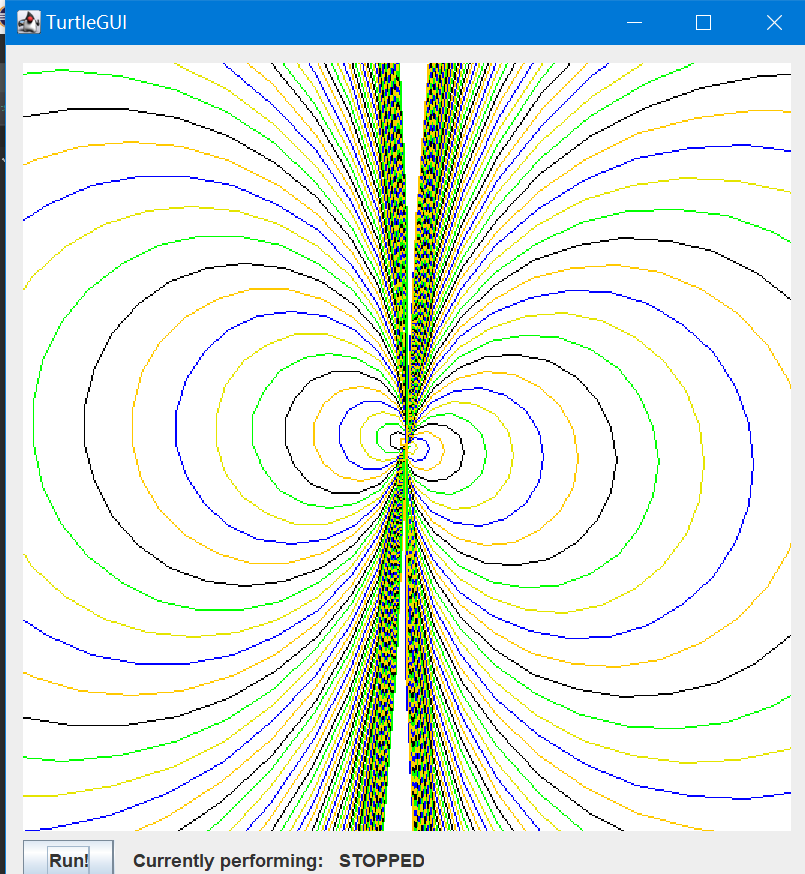




### Problem 8: Personal art

随便画的，没什么艺术可言





### Submitting

在项目下的文件夹中打开git-bash

git add . Lab1-1190201215

git commit -m “src”

git push

## Social Network

本次任务要求实现Person和FriendshipGraph两个类。

用FriendshipGraph来构建Person之间的关系来模拟社交网络，能够计算出每两个Person之间的最短路径。

### 设计/实现FriendshipGraph类

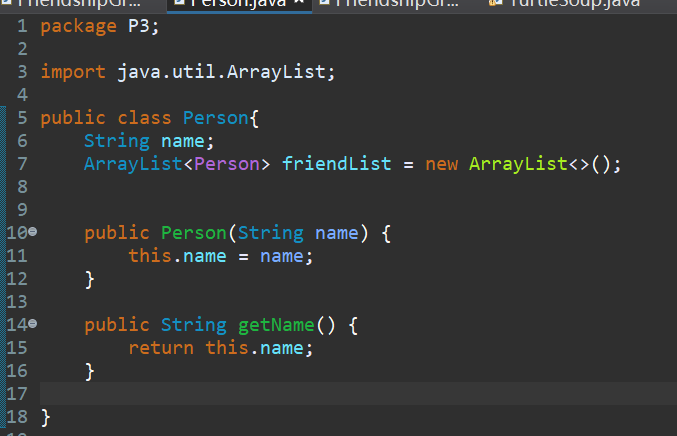
1. ArrayList<Person> people 人群的一个动态数组
2. Set<String> nameSet 名字表
3. 函数addVertex(Person a) people中添加一个Person，名字表中添加一个名字
4. 函数addEdge(Person m, Person n)，在m的好友列表中添加n
5. 函数getDistance(Person m, Person n)，计算m和n的距离（使用BFS）



### 设计/实现Person类

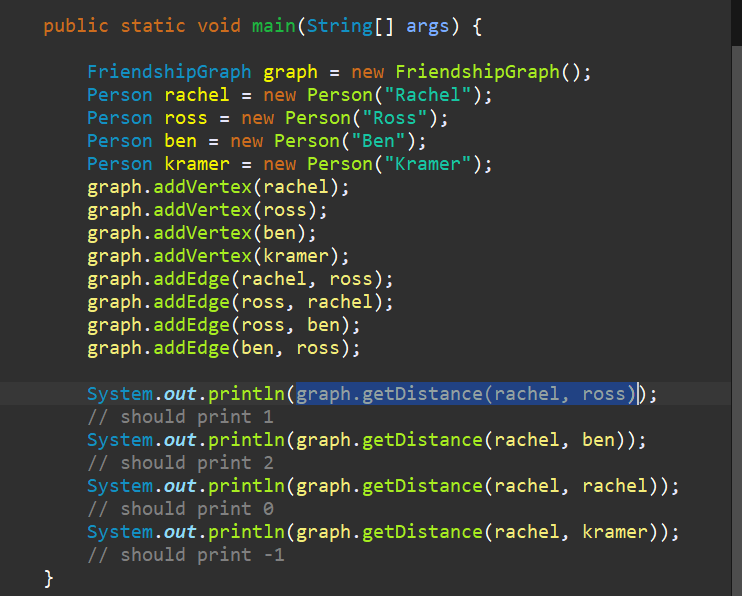
ArrayList<Person> friendList，一个朋友列表

函数getName()，返回这个Person的名字

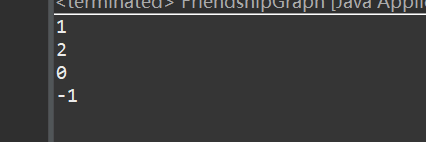


### 设计/实现客户端代码main()

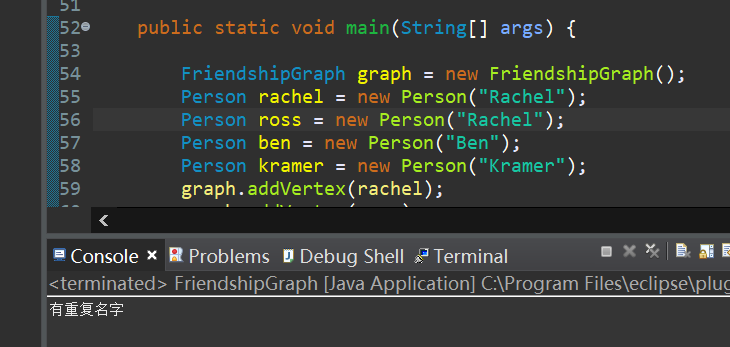
实验指导书已经给出。



结果：



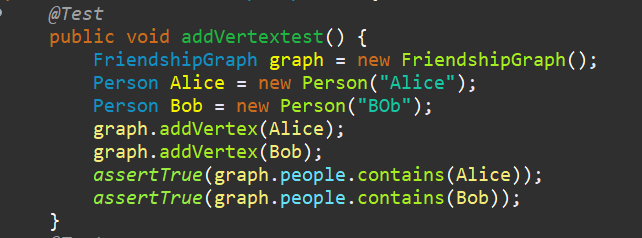
将第三行Ross换为Rachel后输出结果：有重复名字



### 设计/实现测试用例

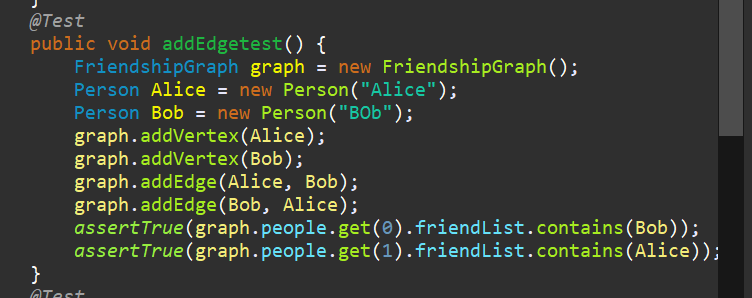
addVertextest()

检验人群中是否由Alice和Bob



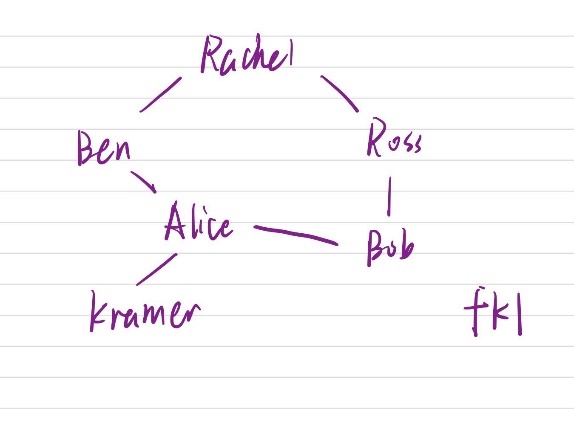
addEdgetest()

检验Alice朋友列表里是否有Bob，Bob列表里是否有Alice



getDistancetest()

构造如下关系网：



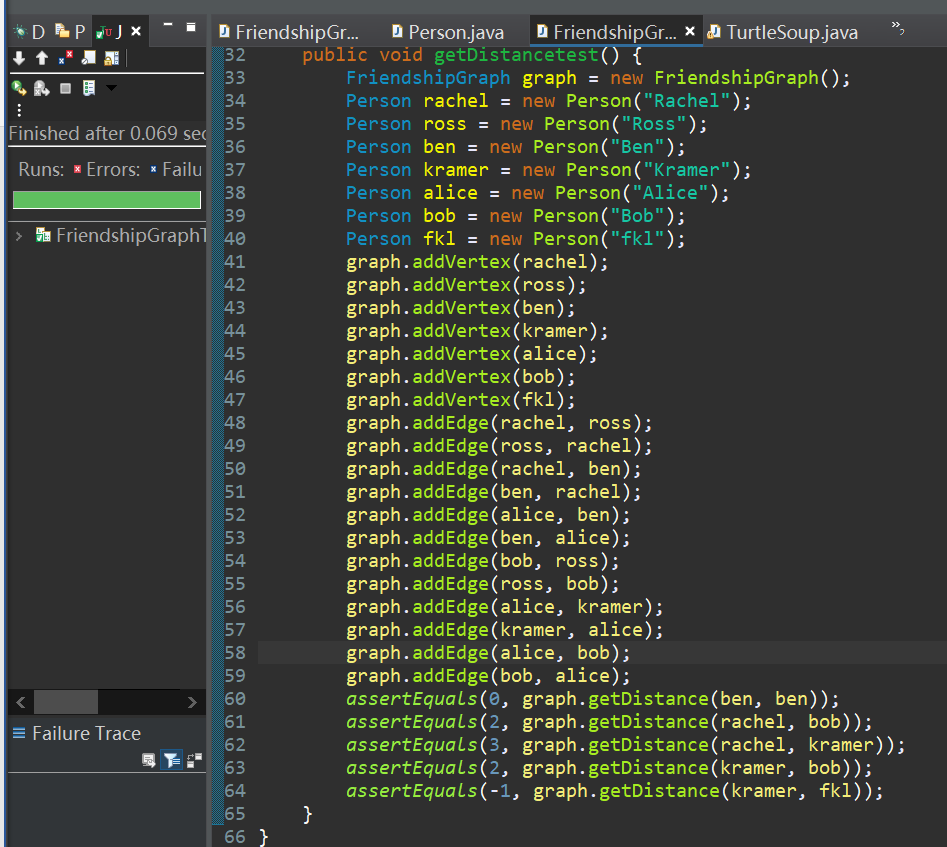
检查ben与ben是否为0

检查rachel，bob是否为2

检查rachel，kramer是否为3

检查kramer，bob是否为2

检查kramer，fkl是否为0



# 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 |
| 2020-05-14 | 14:00-15:30 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按计划完成 |
| 2020-05-15 | 12:00-14:00 | P1 generateMagicSquare | 按计划完成 |
| 2020-05-16 | 12:00-14:00 | P2 除了凸包所有的函数 | 按计划完成 |
| 2020-05-20 | 14:00-16:00 | P2 凸包 | 遇到困难，未完成 |
| 2020-05-21 | 10:00-02:30 | P3 FriendshipGraph、Person、test | 遇到困难，严重延迟 |
| 2020-05-22 | 11:00-12:00 | 完成报告P1 | 按计划完成 |
| 2020-05-23 | 14:00-16:00 | 完成报告P2、P3、总结部分 | 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 |
| Git不会使用 | 询问同学，查看csdn |
| 凸包不会编 | 询问同学，查看csdn |
| Person类的构造 | 询问同学，查看csdn |

# 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. Java编程语言是否对你的口味？

还行吧。一般般，没什么感觉

1. 关于Eclipse IDE

太棒了，遇到有bug的地方还有自动帮助修改

1. 关于Git和GitHub

这个东西看起来挺智能，但我不是很熟悉

1. 关于CMU和MIT的作业

还行吧，有点难度

1. 关于本实验的工作量、难度、deadline

工作量太大了！！

难度太高了！！

Deadline太紧了！！

1. 关于初接触“软件构造”课程

有点意思，是一个很有学问的学科。