

**2021年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 王钲皓 |
| 学号 | 1190300107 |
| 班号 | 1903006 |
| 电子邮件 | [289410265@qq.com](mailto:289410265@qq.com) |
| 手机号码 | 13037561267 |

**目录**

[1 实验目标概述 1](#_Toc72671464)

[2 实验环境配置 1](#_Toc72671465)

[3 实验过程 1](#_Toc72671466)

[3.1 Magic Squares 2](#_Toc72671467)

[3.1.1 isLegalMagicSquare() 2](#_Toc72671468)

[3.1.2 generateMagicSquare() 3](#_Toc72671469)

[3.2 Turtle Graphics 4](#_Toc72671470)

[3.2.1 Problem 1: Clone and import 4](#_Toc72671471)

[3.2.2 Problem 3: Turtle graphics and drawSquare 4](#_Toc72671472)

[3.2.3 Problem 5: Drawing polygons 5](#_Toc72671473)

[3.2.4 Problem 6: Calculating Bearings 5](#_Toc72671474)

[3.2.5 Problem 7: Convex Hulls 6](#_Toc72671475)

[3.2.6 Problem 8: Personal art 8](#_Toc72671476)

[3.2.7 Submitting 9](#_Toc72671477)

[3.3 Social Network 9](#_Toc72671478)

[3.3.1 设计/实现FriendshipGraph类 9](#_Toc72671479)

[3.3.2 设计/实现Person类 11](#_Toc72671480)

[3.3.3 设计/实现客户端代码main() 12](#_Toc72671481)

[3.3.4 设计/实现测试用例 12](#_Toc72671482)

[4 实验进度记录 13](#_Toc72671483)

[5 实验过程中遇到的困难与解决途径 14](#_Toc72671484)

[6 实验过程中收获的经验、教训、感想 14](#_Toc72671485)

[6.1 实验过程中收获的经验和教训 14](#_Toc72671486)

[6.2 针对以下方面的感受 14](#_Toc72671487)

# 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/IDEA 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

在官网上下载Eclipse、JDK和Git并安装。在为JDK配置环境变量的时候，在系统环境变量中新建一个叫“JAVA\_HOME”的环境变量，变量值为JDK的安装路径。然后在Path的环境变量中添加一个“%JAVA\_HOME%bin”的变量值。省去很多其它的步骤。

https://github.com/ComputerScienceHIT/HIT-Lab1-1190300107.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。

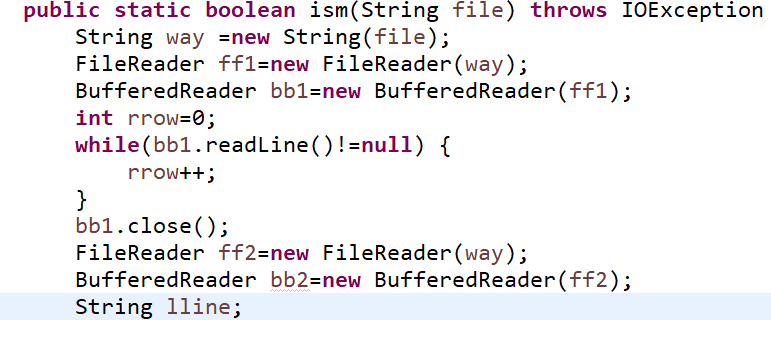
## Magic Squares

Magic Squares即幻方，n阶幻方是一个正方形n×n个数（通常是不同的整数）的排列，使得所有行、所有列和两个对角线中的n个数总和为同一常数。

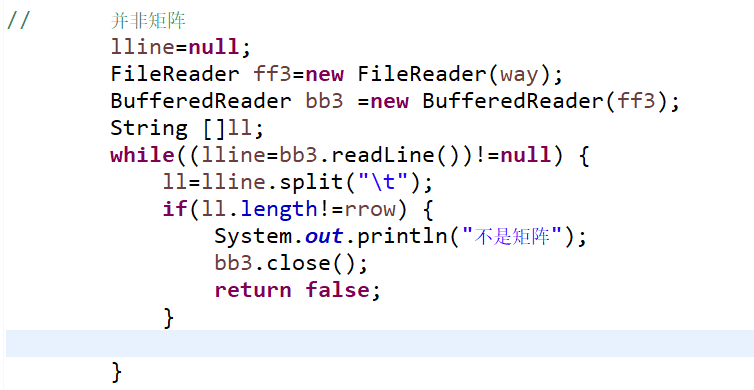
### isLegalMagicSquare()

本题要求在main()函数中调用五次isLegalMagicSquare（）函数，将5个文本文件名分别作为参数输入进去，看其是否得到正确的输出（true，false）。

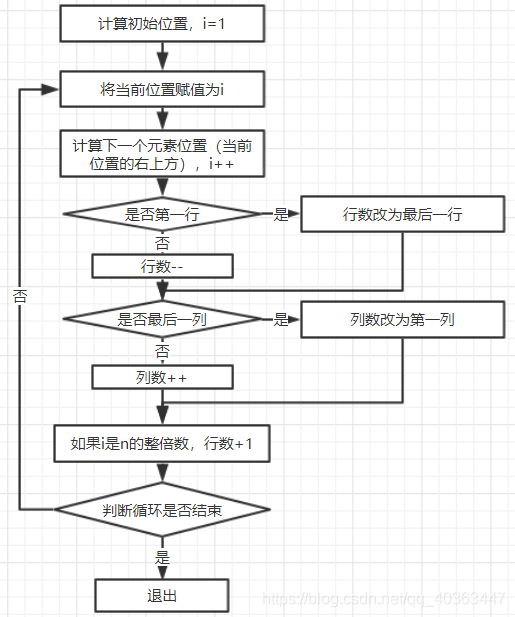
首先，逐行阅读文件，得到“矩阵”的行数。



重新读取文件，使用myLine.split("\t")方法划分得到一个字符串数组。

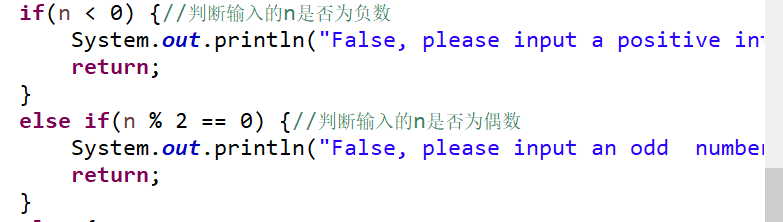
数组的长度即为矩阵的阶，将它和行数比较。使用String.contains判断格式是否正确。使用Integer.valueof方法，判断格式是否正确，同时判断是否为正整数，转换字符串为整数得到二维int型数组。计算二维数组行，列，对角线之和。

### generateMagicSquare()



首先计算初始位置（0，n/2），赋值为1，之后每次取当前位置的右上角的位置，设置的值每次加1，如果当前行是第一行，则下一行为最后一行，如果当前列是最右边的列，则下一次取左边第一列，如此重复平方次，就对整个矩阵赋完值了，且满足幻方定义。

在函数开头判断输入是否为偶数，负数，输出相应的提示信息。



## Turtle Graphics

根据代码注释提示补全代码从而实现一个绘制工具Turtle Graphics。

1、从Github上获取任务的源代码，在本地创建git仓库，使用git管理本地开发

2、获取代码并导入eclipse

3、忽略

4、画出正四边形。

5、Commit and push

6、画出正多边形

7、计算偏转角

8、计算凸包

9、绘制自己的图形

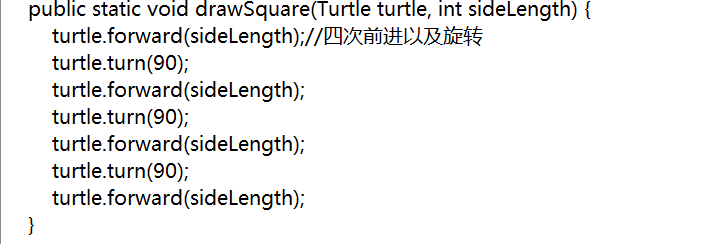
### Problem 1: Clone and import

从github上获取代码：网页download。从本地创建git仓库：1、git init初始化本地仓库2、git add remote添加远程库源3、在远程仓库创建master分支4、git pull origin master将远程仓库同步到本地5、git add \* ->git commit -m “”->git push将本地文件加入到本地仓库，将本地仓库同步到远程仓库。

### Problem 3: Turtle graphics and drawSquare

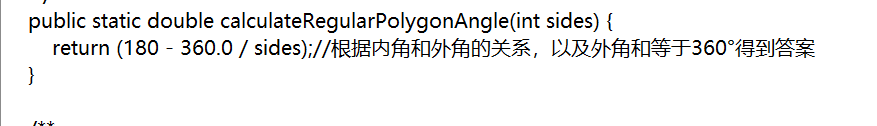
1、首先发现forward、turn函数的功能是实现给定参数的前进和转向

2、我们通过四次循环，每次前进+转90°就可以得到我们想要的正方形

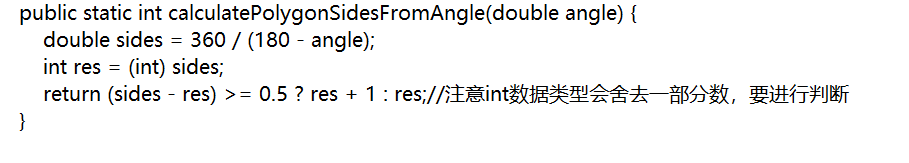


### Problem 5: Drawing polygons

1. 补全calculateRegularPolygonAngle。正多边形的每个内角等于180.0 – 360.0 / 边数，

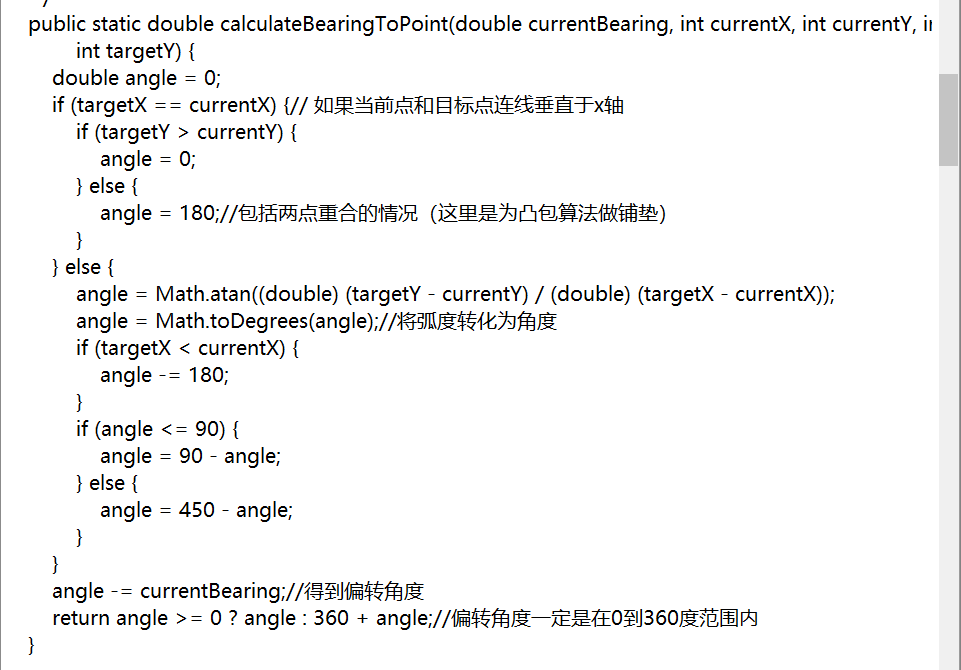


1. 补全calculatePolygonSidesFromAngle。知道正多边形的内角，求边数，根据数学公式可以直接得出：

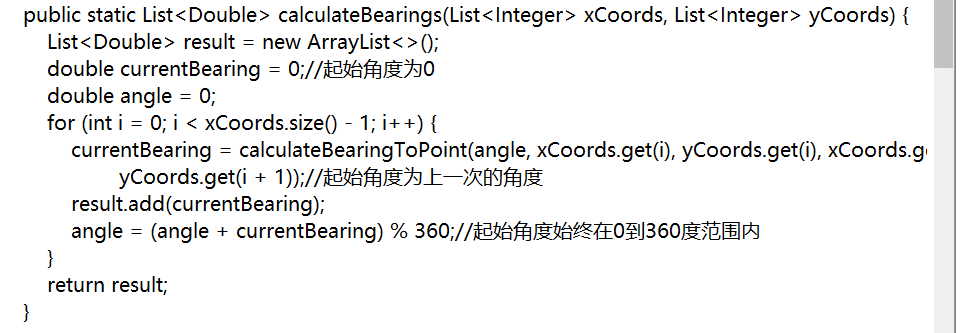


### Problem 6: Calculating Bearings

1. 补全calculateBearingToPoint。该函数的功能是当前点朝向的向量与当前点朝向目标点构成的向量之间的夹角。



1. 补全calculateBearings。对列表中的每两个相邻点调用calculateBearingToPoint计算夹角，列表保存并返回。



### Problem 7: Convex Hulls

1、补全Convex Hulls。该函数的功能是求出包围所有点的凸包。

2、首先选取最左下角的点作为起始点，然后每次选择转向角最小的点（同时距离最长）加入集合

public static Set<Point> convexHull(Set<Point> points) throws InterruptedException {

Set<Point> convexHull = new HashSet<Point>();

List<Point> pointsSet = new ArrayList<>();

convexHull.clear();

if (points.size() == 0) {

return convexHull;//得到集合中的元素

}

int count = 0;

for (Point point : points) {

pointsSet.add(count, point);

count++;

} // 寻找最左下方的点

double xleft = pointsSet.get(0).x();// 用于记录最小的x

int index = 0;// 用于记录最左下角的点

for (int i = 0; i < count; i++) {

if ((int)pointsSet.get(i).x() < xleft) {// 判断x值的大小，找到较小的x及其所在的位置

xleft = pointsSet.get(i).x();

index = i;

continue;

}

if ((int)pointsSet.get(i).x() == xleft) {// 若该点与我们选的x值相等，则判断y值大小

if ((int)pointsSet.get(i).y() < pointsSet.get(i).y()) {

index = i;

}

}

}

convexHull.add(pointsSet.get(index));// 遍历寻找turn的角度最小的点集

int indexfinal = index;// 用于记录选择的下一个点，同时用于while循环的比较出口

double angle1 = 0, angle2 = 0;

int k = 0;// 用于更新待选取的点

Boolean[] flags = new Boolean[count];

for (int i = 0; i < count; i++) {

flags[i] = false;

}

flags[index] = true;

do {//为了方便计算，所有的起始角度均设为0度

for (int i = 0; i < count; i++) {// 寻找一个起始的计算点

if (!flags[i] || i == index) {

angle1 = calculateBearingToPoint(0, (int) pointsSet.get(indexfinal).x(),

(int) pointsSet.get(indexfinal).y(), (int) pointsSet.get(i).x(), (int) pointsSet.get(i).y());

k = i;

break;

}

}

for (int i = 0; i < count; i++) {

if (!flags[i] || i == index) {//比较得到一个最小的偏转角所在的终点

angle2 = calculateBearingToPoint(0, (int) pointsSet.get(indexfinal).x(),

(int) pointsSet.get(indexfinal).y(), (int) pointsSet.get(i).x(), (int) pointsSet.get(i).y());

if (angle2 < angle1) {

angle1 = angle2;

k = i;

continue;

}

if (angle2 == angle1) {// 若有两个点偏转角度一样，则选择距离计算目前点较远的点

if (Math.abs(pointsSet.get(indexfinal).x() - pointsSet.get(i).x()) > Math

.abs(pointsSet.get(k).x() - pointsSet.get(indexfinal).x())) {

k = i;

}

}

}

}

indexfinal = k;

if (indexfinal != index) {//如果终点就是起点，则无需再次加入该点

convexHull.add(pointsSet.get(indexfinal));

System.out.println("加入点"+"("+pointsSet.get(indexfinal).x()+","+(pointsSet.get(indexfinal).y())+")");

}

flags[indexfinal] = true;

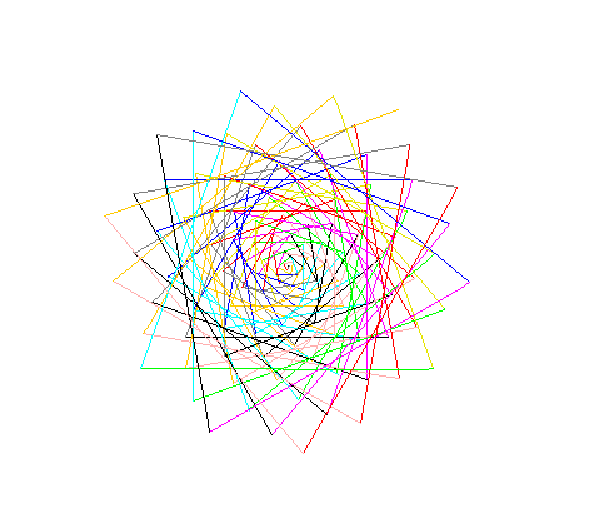
} while (index != indexfinal);

return convexHull;

}

### Problem 8: Personal art

简单的for循环，前进量是变量就可以画出好看的图形，偏转角要合适，可以通过switch语句改变图形颜色



### Submitting

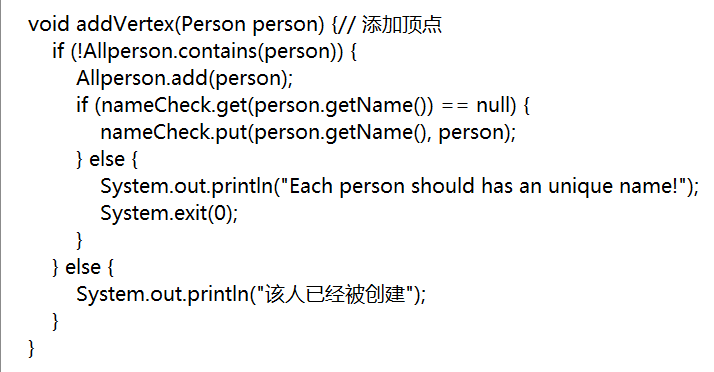
git add \*->git commit -m “”->git push提交到Lab1仓库

## Social Network

利用图结构实现一张人际关系网

### 设计/实现FriendshipGraph类

1设计void addVertex(Person person)函数,实现在网中添加顶点



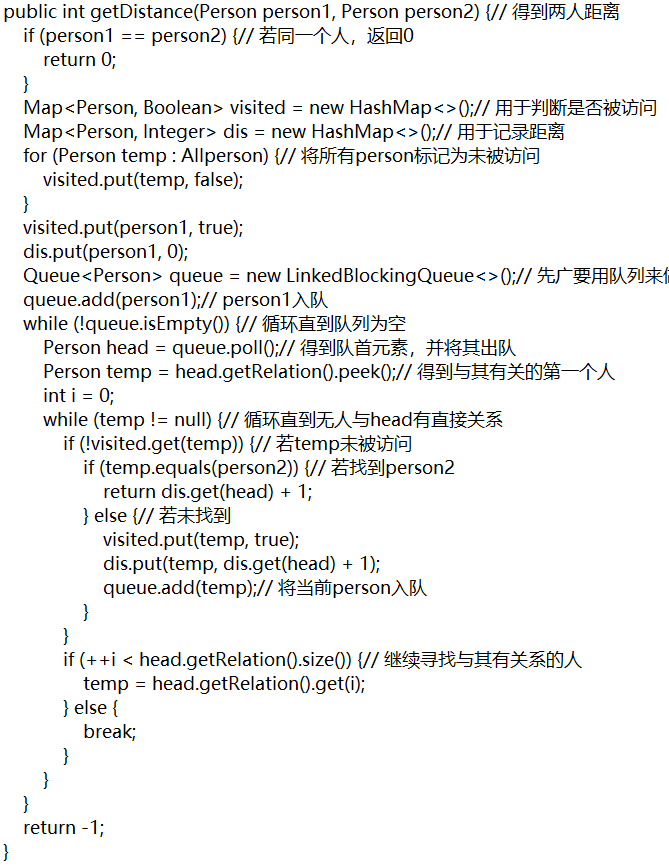
2.设计void addEdge(Person person1, Person person2)添加边

void addEdge(Person person1, Person person2) {// 添加关系边

person1.addRelation(person2);

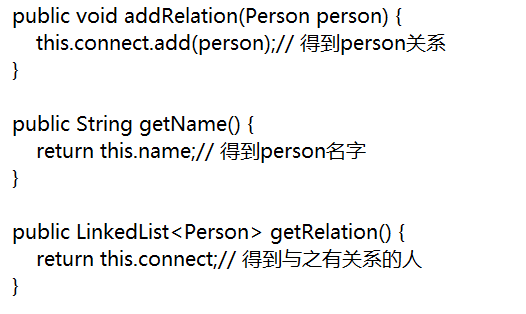
}

3.设计public int getDistance(Person person1, Person person2)函数得到最近距离，采用BFS得到结果



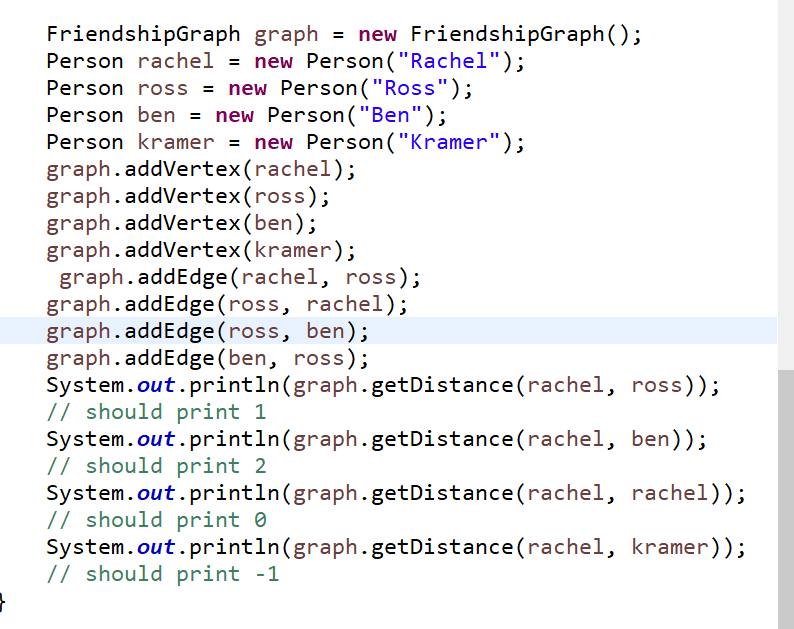
### 设计/实现Person类

Person类是对人的刻画，应该含有名字（私有），朋友的集合（私有），构造方法，结识朋友，得到名字，得到朋友网这些方法。



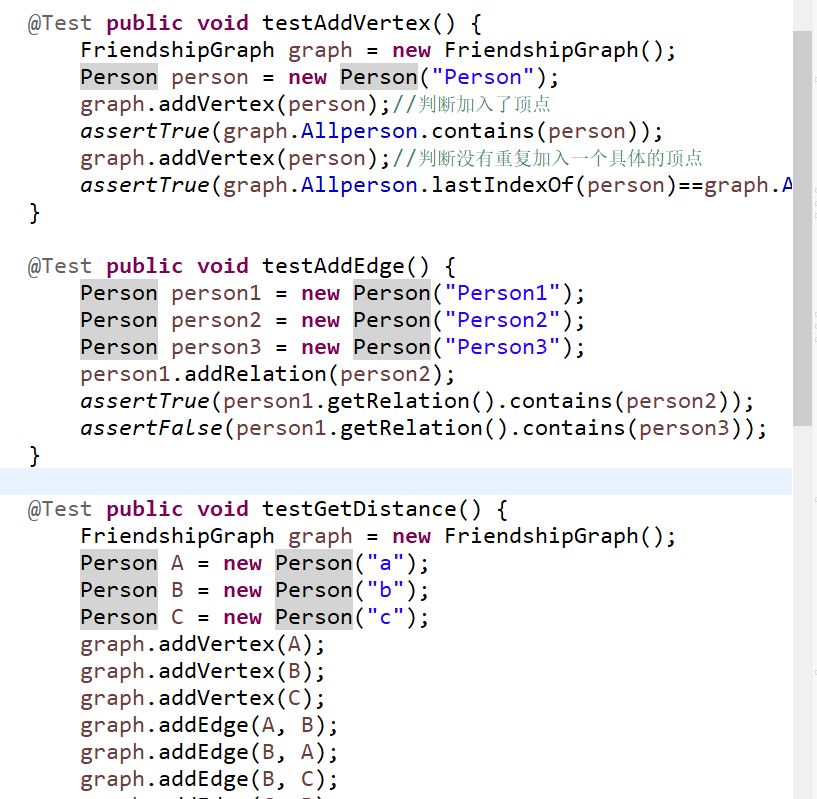
### 设计/实现客户端代码main()

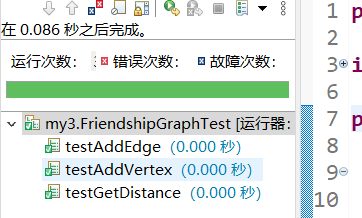
根据实验指导，先创建关系网graph,创建一些Person，然后把它们添加到图中，接着添加边。



### 设计/实现测试用例

对每个方法进行依此测试，使用assertTrue,assertFalse,assertEquals方法测试。





# 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 |
| 2021-05-13 | 18:30-19:30 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按计划完成 |
| 2021-05-17 | 8：00-17：00 | 编写问题2的Turtlesoup | 完成 |
| 2021-05-18 | 8：00-17：00 | 编写问题3的Feriendship和Person | 完成 |
| 2021-05-19 | 8：00-17：00 | 编写问题3的测试用例 | 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 |
| 实验英语看不懂 | 百度翻译 |
| 实验一文件打开不会 | 上CSDN查，翻书学，看网课 |
| 实验三测试用例不会写 | 和同学交流 |

# 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

英语很重要，我得好好学英语。

Java比较生疏，还需要多加练习。

基本的数据结构有点忘了，应该及时复习。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？还行
2. 关于Eclipse IDE；还可以
3. 关于Git和GitHub；太难了，都是英语
4. 关于CMU和MIT的作业；很好，但是看不完全明白
5. 关于本实验的工作量、难度、deadline；比较难
6. 关于初接触“软件构造”课程；感觉上课讲的和实验课联系不太大啊