

**2022年春季学期**

**计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 高岩 |
| 学号 | 120L021526 |
| 班号 | 2003002 |
| 电子邮件 | 2494005425@qq.com |
| 手机号码 | 18079298013 |

**目录**

[1 实验目标概述 1](#_Toc2092)

[2 实验环境配置 1](#_Toc21866)

[3 实验过程 1](#_Toc10604)

[3.1 Magic Squares 1](#_Toc21155)

[3.1.1 isLegalMagicSquare() 1](#_Toc7670)

[3.1.2 generateMagicSquare() 2](#_Toc778)

[3.2 Turtle Graphics 3](#_Toc27349)

[3.2.1 Problem 1: Clone and import 3](#_Toc27085)

[3.2.2 Problem 3: Turtle graphics and drawSquare 3](#_Toc6029)

[3.2.3 Problem 5: Drawing polygons 3](#_Toc25922)

[3.2.4 Problem 6: Calculating Bearings 4](#_Toc12918)

[3.2.5 Problem 7: Convex Hulls 4](#_Toc13911)

[3.2.6 Problem 8: Personal art 4](#_Toc31979)

[3.2.7 Submitting 6](#_Toc13432)

[3.3 Social Network 6](#_Toc9765)

[3.3.1 设计/实现FriendshipGraph类 6](#_Toc6541)

[3.3.2 设计/实现Person类 6](#_Toc29112)

[3.3.3 设计/实现客户端代码main() 6](#_Toc28416)

[3.3.4 设计/实现测试用例 6](#_Toc20223)

[4 实验进度记录 6](#_Toc11248)

[5 实验过程中遇到的困难与解决途径 7](#_Toc20742)

[6 实验过程中收获的经验、教训、感想 7](#_Toc22008)

[6.1 实验过程中收获的经验和教训（必答） 7](#_Toc12980)

[6.2 针对以下方面的感受（必答） 7](#_Toc16694)

# 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。

l 基本的 Java OO 编程

l 基于 Eclipse IDE 进行 Java 编程

l 基于 JUnit 的测试

l 基于 Git 的代码配置管理

# 实验环境配置

1.下载安装eclipse和jdk

2.下载安装git

3.配置JUnit

Lab1仓库URL：<https://github.com/ComputerScienceHIT/HIT-Lab1-120L021526>

# 实验过程

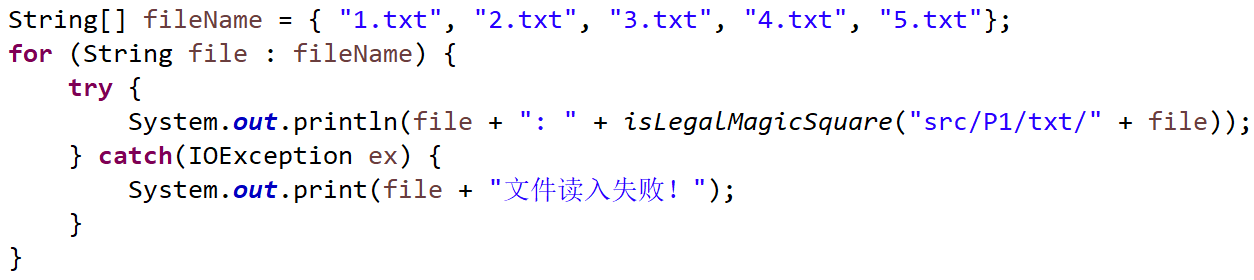
## Magic Squares

Magic Squares是一个n阶的魔方，由n×n个数字（通常是不同的整数）排列而成的正方形。在该正方形中，所有行、所有列和两条对角线上的n个数字的总和为同一个常数。

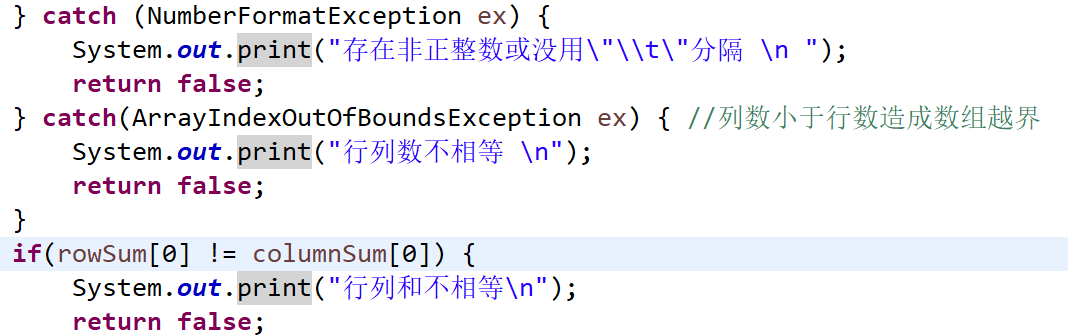
### isLegalMagicSquare()

该函数实现从文件读入一个矩阵并判断其是否幻方的功能。

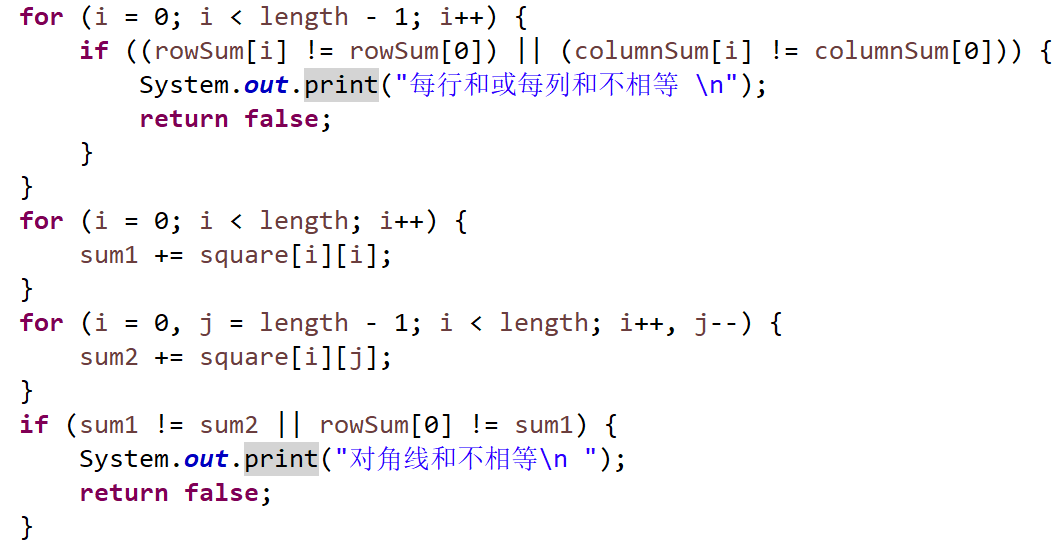
其先读入所需的五个txt文件，然后通过读入一行数据并依据数据总数创建方阵，



再经过for循环读入整个矩阵，如果读入的数含有负数、小数等数据，或者数据未按要求经\t分隔，则会出现NumberFormatException异常，如果由矩阵行列数不等而造成在读入矩阵数据时访问数组越界，则会出现ArrayIndexOutOfBoundsException异常，运行后则会输出提示信息并返回false。



当输入数据没有任何问题时，我们就只需要判断矩阵各行、各列和两个对角线上数据的和是否都相等，如果相等则读入的为Magic Squares，返回true，不然则不是幻方矩阵返回false。



### generateMagicSquare()

这里生成幻方矩阵的思路是根据传入的n值，确定一个大小为n\*n的方阵，然后依据下面流程图所显示的规则将数填入方阵。



## Turtle Graphics

由MIT的实验要求，完成TurtleSoup.java中未完成的部分，从而实现Turtle绘图工具并创作图案。

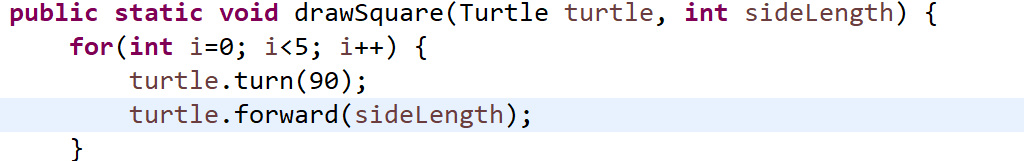
### Problem 1: Clone and import

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。

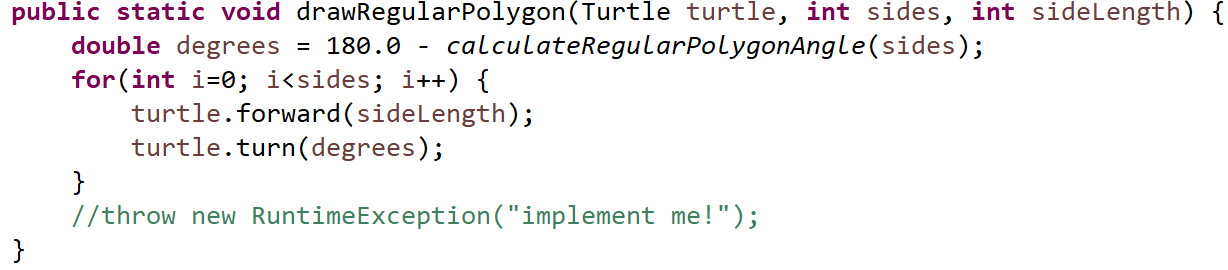
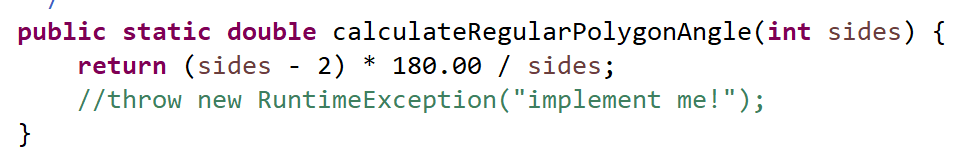
访问实验手册给出的URL地址，然后打开目标文件夹，右键点击git bash，输入git clone https://github.com/rainywang/Spring2022\_HITCS\_SC\_Lab1.git即可。

### Problem 3: Turtle graphics and drawSquare

我们使用drawSquare方法，然后给定turtle的forward和turn方法，实现前进和旋转，然后对该过程重复多次即可。



### Problem 5: Drawing polygons

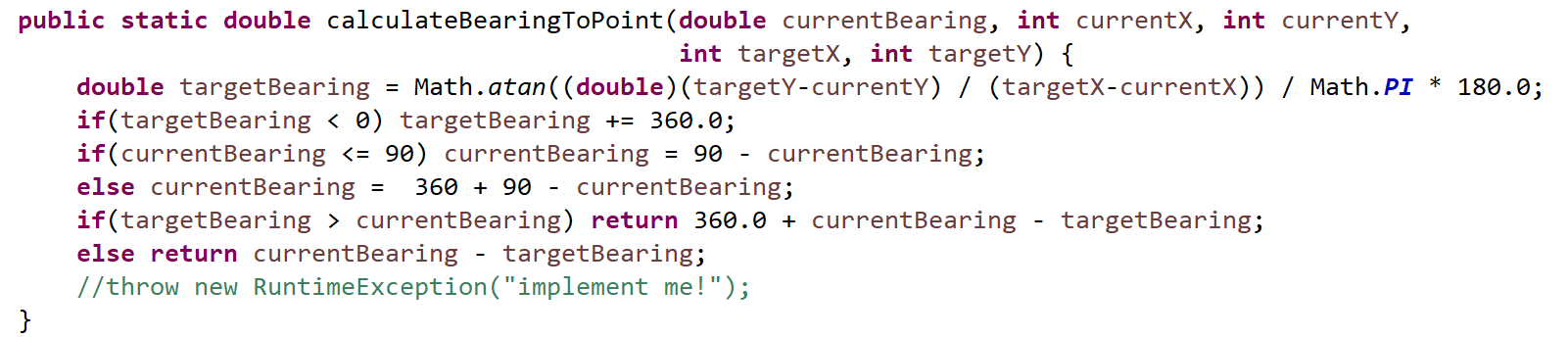
先给出calculateRegularPolygonAngle方法，然后再定义drawRegularPolygon方法，利用查阅到的正多边形边数和内角计算公式（sides-2)\*180.0/sides，调用calculateRegularPolygonAngle方法计算正多边形内角，再用180减去内角大小即得到转过的角度。

### Problem 6: Calculating Bearings

我们使用calculateBearingToPoint方法，计算以现在的方向从当前位置到目标位置需要转过的角度，即当前点到目标点的向量和当前方向向量之间的夹角。

由两个点的坐标计算夹角的正切值 (targetY-currentY) / (targetX-currentX)，然后调用Math类的静态反三角函数方法Math.atan得到弧度值，然后除以Math.PI乘以180得到角度大小，如果得到的角度是负值，需要将其加360°转换为正值。然后判断当前的角度是否小于90°，小于就可以直接转换为极坐标下的极角，如果大于，就需要在计算后的值上加360°转换为正值。通过以上的转换，我们成功的把角度转换到了极坐标下，然后就可以直接相减得到需要转过的角度，而当结果是负值时同样需要加360°转换为正值。

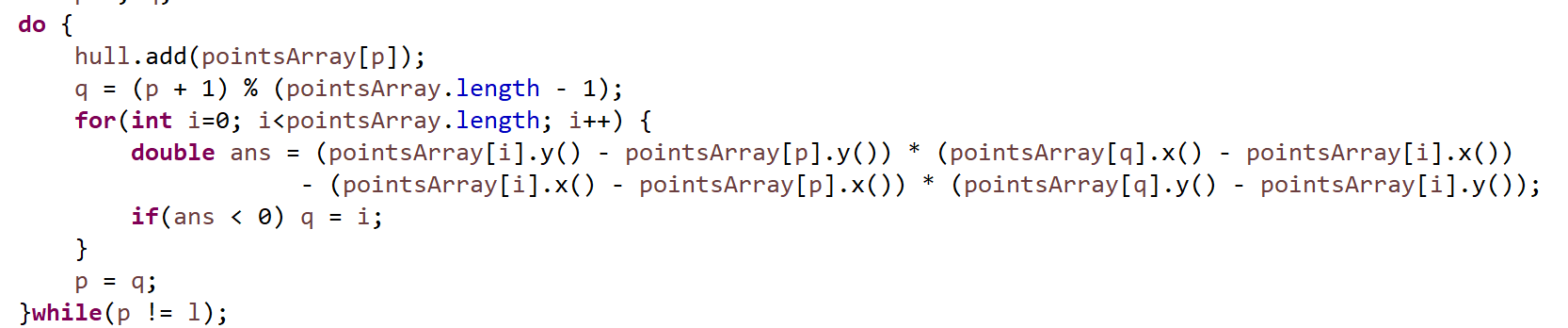
实现calculateBearings，目的是计算从第一个点开始，每次到下一个点需要转过的角度。只需要调用calculateBearingToPoint方法计算即可。



### Problem 7: Convex Hulls

实现convexHull，如果点集中的点数小于3时，直接返回原点集即可。否则，先找出这个点集中最左的点，即横坐标最小，然后使用向量的外积寻找以这个点到下一个逆时针转过角度最大的点，然后以下一个点为起点继续寻找，直到找到的点是最开始的起点，返回所经过的点集即可。

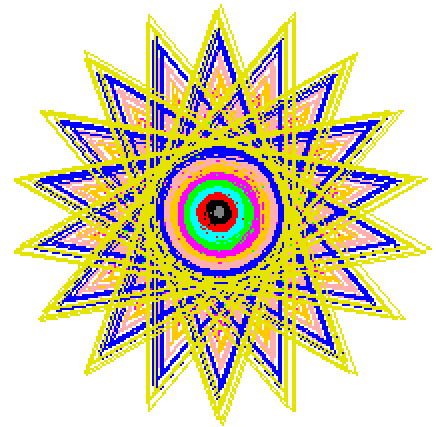
下图所示部分代码为寻找某个点为起点，判断其他点，如果ans小于零，说明到i这个点比到q这个点逆时针转过的角度更大，所以更新q=i。

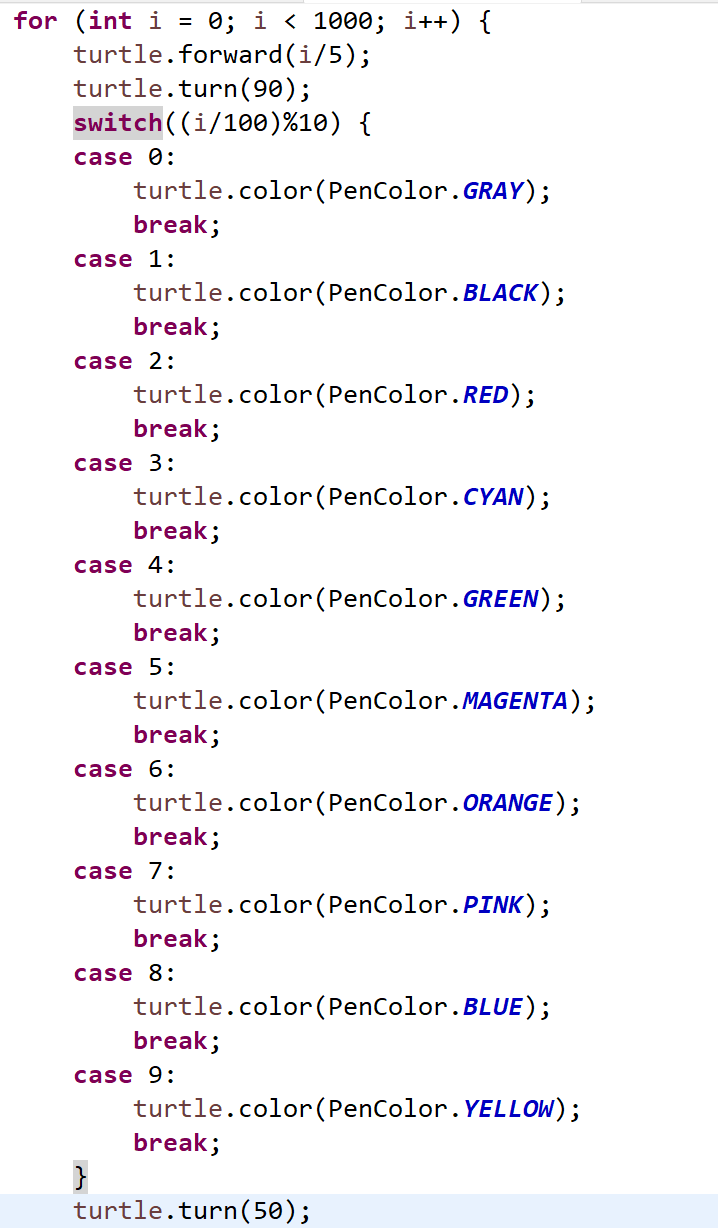


### Problem 8: Personal art

最终通过自己实现的功能绘出如下图形。

方法：通过for循环，由于以循环变量为步长的效果过大，线条过于疏散，因此把每次前进的步长缩减为原来的i/5，并转过90°，每次以i的百位为标准，然后着色，再旋转50°进入下一次循环。





### Submitting

Git add

Git commit -m “complete P2”

Git push

## Social Network

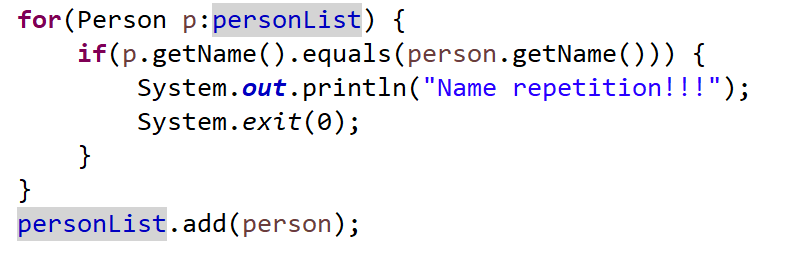
填充完成Person和FriendshipGraph两个类，模拟社交网络，能够实现添加节点、节点之间添加边的功能，并且可以计算两节点之间最短路。设计/实现FriendshipGrap。

### 设计/实现FriendshipGraph类

一共要求实现三个方法：addVertex向社交网络中添加新的节点(人)，addEdge向社交网络中添加新的边(社交关系)，getDistance用来求两个点之间的最短距离。

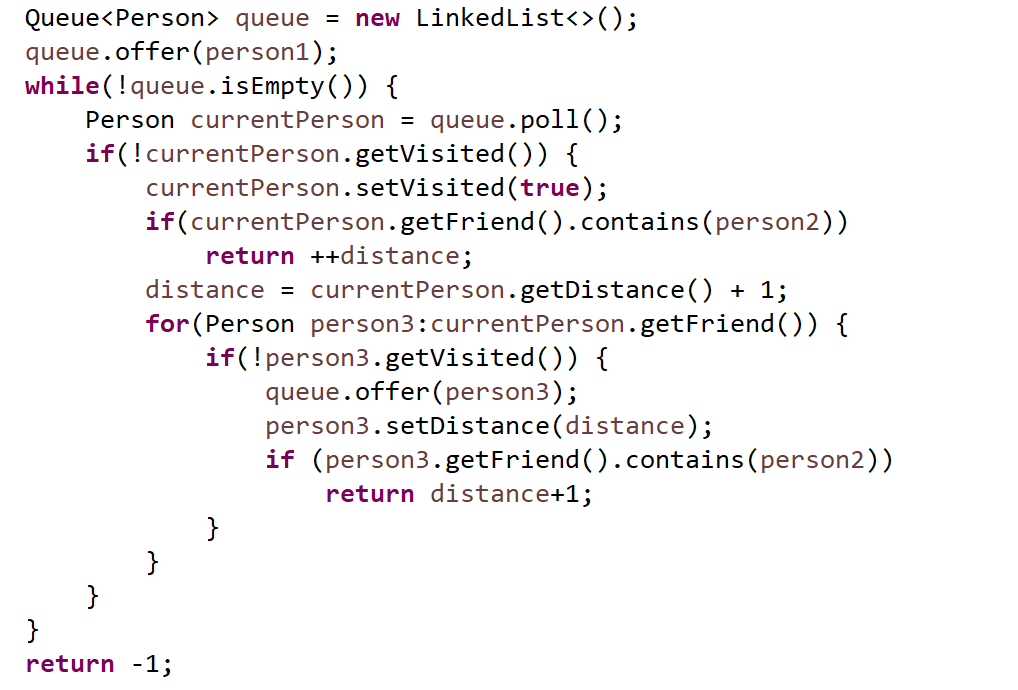
由于要保存所有的点，所以定义了一个List<Person> personList来保存所有的人。

addVertex：在社交网络中，根据要求，每个人由唯一的名字，所以在将新的点加入personList之前要先检查这个人的名字是否在之前已经出现过，如果出现过，打印提示信息并结束程序的运行，如果没有，将这个人加入personList中。



addEdge：调用第一个人的addFriend方法将第二个人加入到第一个人的朋友的集合中。

getDistance：先判断person1和person2是不是同一个人，如果是，直接返回距离为0；如果不是再继续进行下去，然后就是一个BFS算法，根据需要增加了计算距离的功能。（1）位置是person2是currentPerson的朋友，只要把当前distance加一返回就可以了。（2）位置是重新设置distance为currentPerson的距离加一，代表currentPerson的朋友到person1的距离，然后在（3）处设置currentPerson的每一个朋友的distance。（4）处判断的是person2是不是currentPerson的朋友的朋友，如果是返回currentPerson的朋友的distance+1即可。



### 设计/实现Person类

private String name：保存Person的名字

private boolean visited：标识这个Person在BFS中是否已经被访问过

private Set<Person> friend：这个Person所有的朋友

private int distance：保存在getDistance中所到达的距离

GetName：获得这个Person的名字；

getVisited：获得访问状态；

setVisited：设置访问状态；

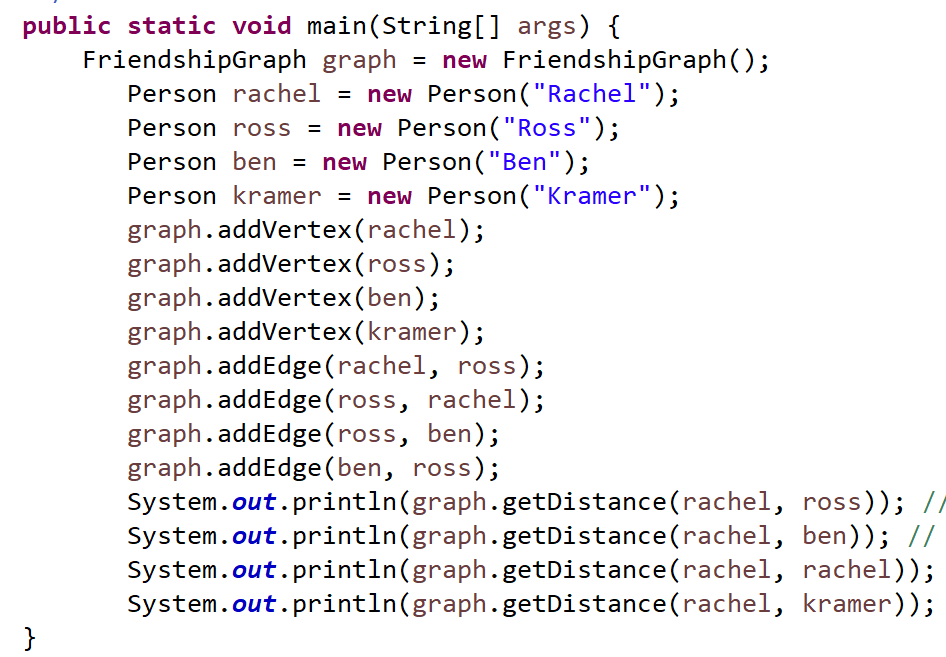
getFriend：获得这个Person所有的朋友；

addFriend：为这个Person添加新的朋友，用于FriendshipGraph类中添加边；

getDistance：获得这个Person的距离；

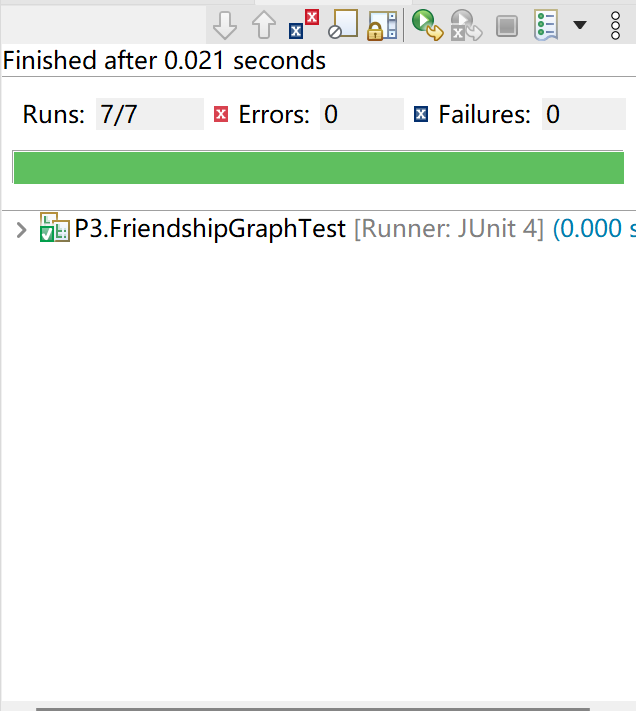
setDistance：设置这个Person的距离。

### 设计/实现客户端代码main()



### 设计/实现测试用例

运行测试代码得：



# 实验进度记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 2022-05-01 | 18:30-20:30 | 编写问题1 | 按计划完成 |
| 2022-05-02 | 18:30-22:30 | 编写问题2 | 延期1小时完成 |
| 2022-05-03 | 18:30-22:30 | 编写问题3和剩余部分 | 延期1小时完成 |

# 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语言使用不熟练 | 在进行实验得同时并继续学习java |
| git不太会用 | 查阅大量资料进行学习 |
| 凸包算法 | 在网上搜索相关的算法，最终得到初步理解 |

# 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）

编写代码只是实现的一部分，更重要的是对已实现部分的测试；

对掌握不熟练的java和git一定得学习透彻；

写测试代码要保证测试用例足够，要注意边界的情况，尤其要测试在复杂情况下的程序的表现。

## 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？

是。Java作为一种面向对象的编程语言，提供了比面向过程的C语言更加丰富的数据结构，但是感觉难一些。

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

仅使用过Eclipse，觉得挺好，各方面都适合用户。IDEA不了解。

1. 关于Git和GitHub，是否感受到了它在版本控制方面的价值；

是的。Git的入门上手较为简单，尤其是版本控制功能尤为好用。Github可以用来与他人共享自己的成果。

1. 关于CMU和MIT的作业，你有何感受；

太难了。

1. 关于本实验的工作量、难度、deadline；

工作量挺大，难度也不小，deadline感觉可以适当再长一些。

1. 关于初接触“软件构造”课程；

有点陌生，开始两周所学的课程基本上都属于概念层面，感觉学习比较吃力。