

**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 | 梁浩 |
| 学号 | 1190200717 |
| 班号 | 1903008 |
| 电子邮件 | [3235962608@qq.com](mailto:3235962608@qq.com) |
| 手机号码 | 15961817952 |

**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() 3](#_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 6](#_Toc29325531)

[3.2.5 Problem 7: Convex Hulls 6](#_Toc29325532)

[3.2.6 Problem 8: Personal art 7](#_Toc29325533)

[3.2.7 Submitting 7](#_Toc29325534)

[3.3 Social Network 7](#_Toc29325535)

[3.3.1 设计/实现FriendshipGraph类 8](#_Toc29325536)

[3.3.2 设计/实现Person类 9](#_Toc29325537)

[3.3.3 设计/实现客户端代码main() 10](#_Toc29325538)

[3.3.4 设计/实现测试用例 12](#_Toc29325539)

[4 实验进度记录 12](#_Toc29325540)

[5 实验过程中遇到的困难与解决途径 13](#_Toc29325541)

[6 实验过程中收获的经验、教训、感想 13](#_Toc29325542)

[6.1 实验过程中收获的经验和教训 13](#_Toc29325543)

[6.2 针对以下方面的感受 13](#_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

通过学生认证下载了IDEA ultimate版本，由于之前大一写过Java web，所以JDK、Jre、环境变量等等都已经配置好了。

Github Lab1仓库的URL地址为：

<https://github.com/ComputerScienceHIT/HIT-Lab1-1190200717/tree/master>

# 实验过程

## Magic Squares

Magic Squares是一种特殊的矩阵，它满足每行、每列和对角线上的数字和都相等。我们需要做的就是从已有的txt文件中读取矩阵并判断是否满足上述性质（同时代码还要能判断txt中的数据是否符合要求）。此外，我们可以利用generateMagicSquare(int n)函数来生成n\*n大小的幻方。

### isLegalMagicSquare()

题目要求写一个函数来判断已有txt文件中存储的数据是否是符合幻方性质的矩阵，函数返回类型为boolean。

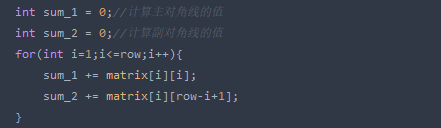
首先创建字符输入流buf、matrix数组，buf用来读取txt文件中的数据，matrix数组存储对应的数据。



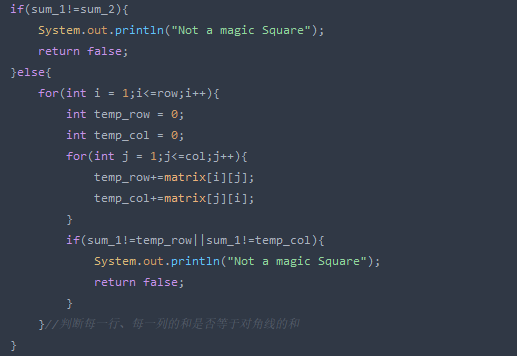
然后按行读入txt文件，如果读到的数据是负数、浮点数或者数据间不是以’\t’分开抛出异常，打印”数据格式有误!”。如果数据无误，统计行数row和列数col，如果不相等，说明不是矩阵，抛出异常，打印” 输入的不是矩阵!”。



如果函数能运行到这里，说明txt中存储的矩阵数据格式没有问题，接下来判断它是否满足幻方的性质。先通过一次循环，获取主对角线元素的和sum\_1和副对角线的和sum\_2。



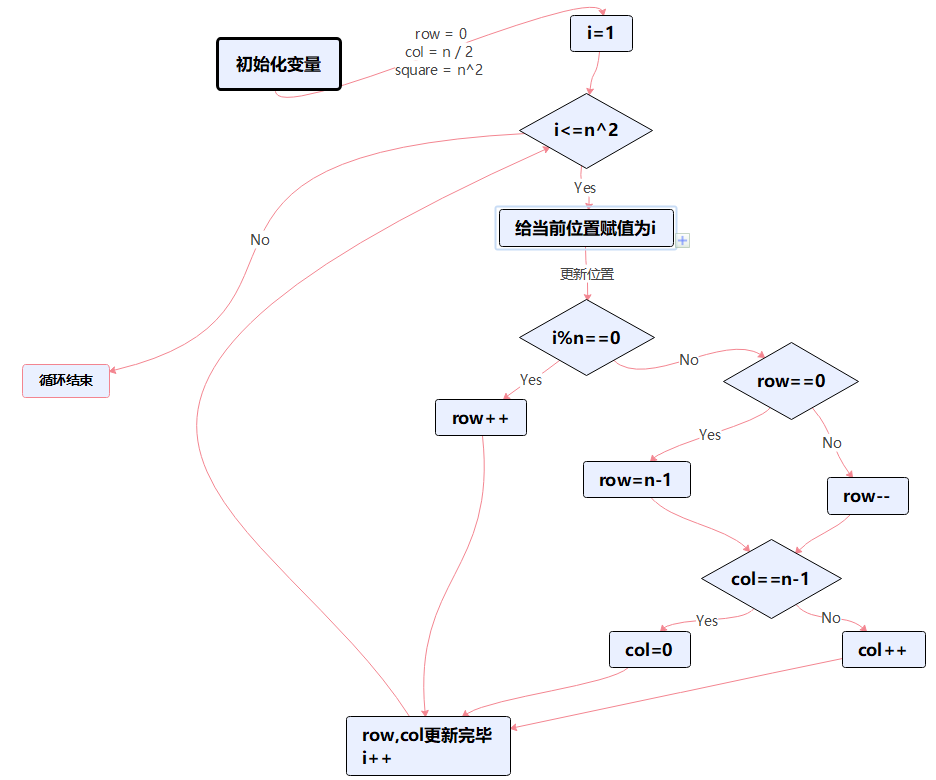
如果不等，直接打印"Not a magic Square"。如果相等，继续比较每一行、每一列的和是否和对角线的和相等。



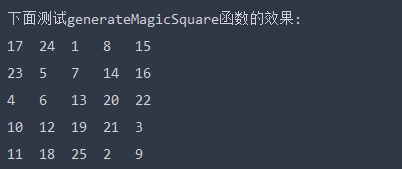
如果程序执行到这里，说明txt中存储的矩阵满足幻方性质，返回true。

### generateMagicSquare()

函数生成幻方的流程如下：



可以用5阶幻方举例：

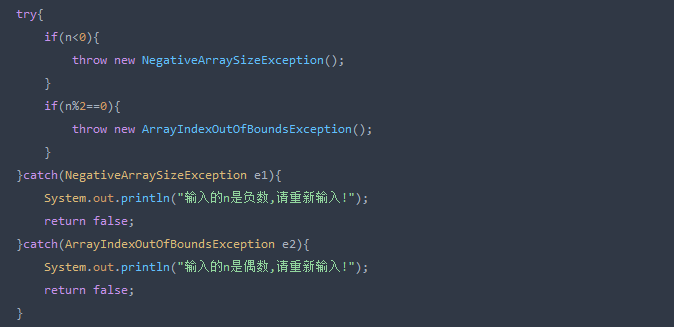


* 把1放在第一行的中间
* 按照顺序将2，3，4，5，......等数放在当前位置的右上方的格子中
* 如果当前位置的右上方格子已经出界，分为三种情况：
  + 如果是行出界，把数填入原本应填入的位置的对应列中的最下方
  + 如果是列出界，把数填入原本应填入的位置的对应行中的最左方
  + 如果行、列同时出界，把数填入当前位置的下方
* 如果当前位置的右上方格子中已经有数，则把数填入当前位置正下方的格子中
* 按照以上的步骤直至填完n^2个格子

对原有的函数进行扩展，

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为负数时），打印错误，函数返回false



## Turtle Graphics

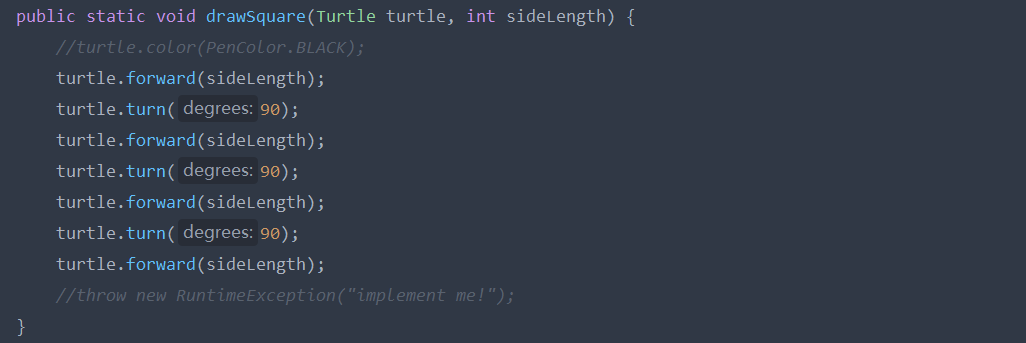
先通过git将已经写好的代码clone下来，通过实现drawSquare函数、calculateRegularPolygonAngle函数、drawRegularPolygon函数、calculateBearingToPoint函数、calculateBearingsh函数、convexHull函数、drawPersonalArt函数来完成所有的问题，同时还要通过TurtleSoupTest.java中的测试用例。

### Problem 1: Clone and import

在B站学习了Git的基本操作，这里选择用命令行直接操作（IDEA本身也支持Git的相关操作）

### Problem 3: Turtle graphics and drawSquare

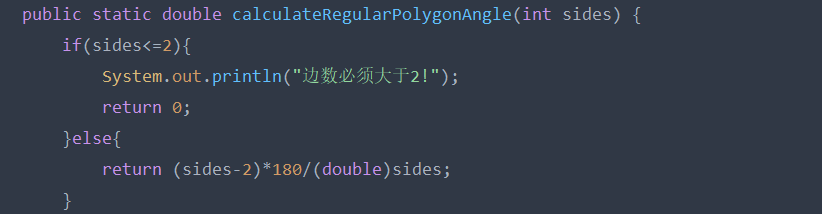
forward是前进，turn是顺时针旋转，要想画出正方形，只需要前进四次，每次转90°即可。



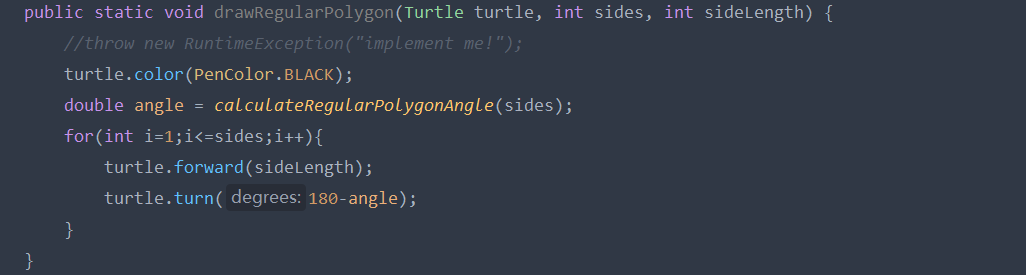
### Problem 5: Drawing polygons

先推出正n边形每个内角的度数：，实现calculateRegularPolygonAngle函数

如果，直接打印错误并退出。



然后实现drawRegularPolygon函数，只需要画n次，每次画完旋转即可。



### Problem 6: Calculating Bearings

先实现calculateBearingToPoint函数：

这里可以利用atan2(double y, double x)函数来计算(currentX , currentY)到(targetX , targetY)的夹角，注意函数返回的是弧度制，要转化为角度制。



当算出来的角度小于0时，加上360之后再返回。



再实现calculateBearings函数，

只要用循环多跑几次calculateBearingToPoint函数即可，将每次算出的结果存在List中。

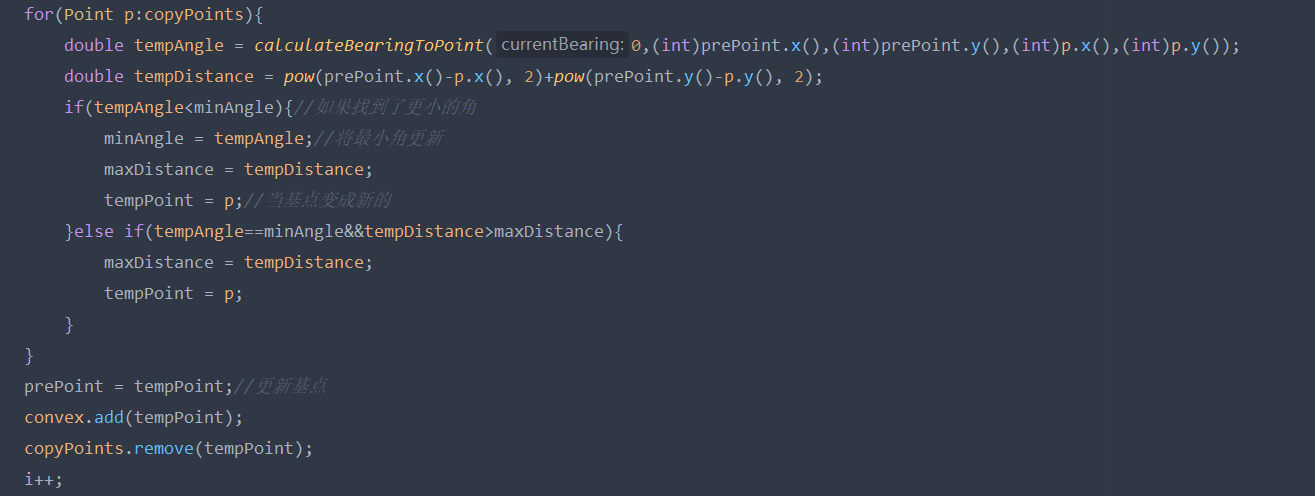
### Problem 7: Convex Hulls

根据提示，这里求凸包时利用The gift-wrapping algorithm，具体步骤如下：

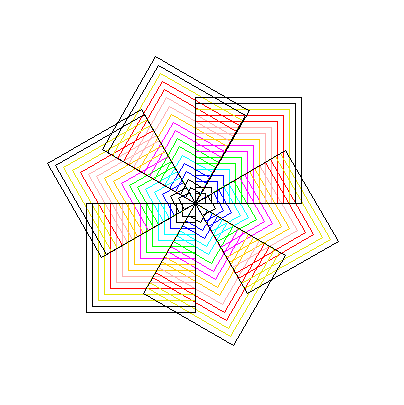
1）先找到所有点中最左下角的点，即x最小，y最小

2）将1）中的点作为初始点，从初始点开始寻找和它夹角最小的点，如果出现夹角相等的情况取距离远的点

3）将2）中取到的点作为新的点，重复操作，直到取到的点是最开始设置的初始点



### Problem 8: Personal art



主要是利用drawSquare函数绘画而成。

### Submitting



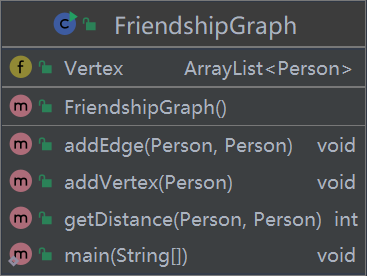
在IDEA关联好自己的远程仓库后可以直接在IDE里选择推送

## Social Network

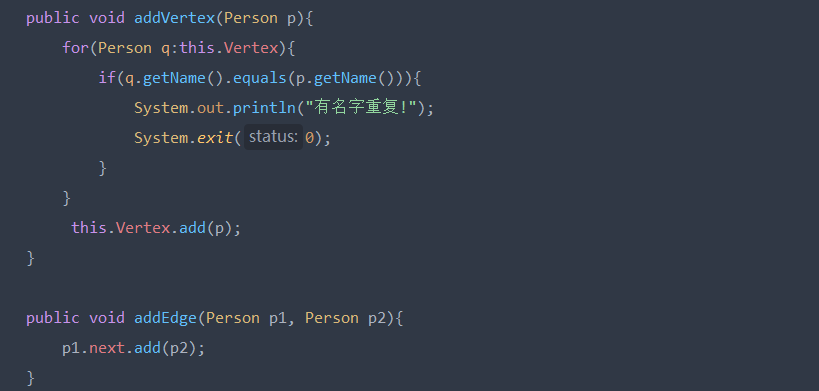
我们需要根据问题设计出两个类Person类和FriendshipGraph类，根据要实现的功能来设计类中的函数。同时要注意些特殊情况，比如两个人的名字相同需要报错，设计的类要支持扩展到有向图。对于getDistance(Person p1, Person p2)函数可以使用BFS算法，同时要注意p1=p2等特殊情况。

### 设计/实现FriendshipGraph类

FriendshipGraph类对应的UML图如下：



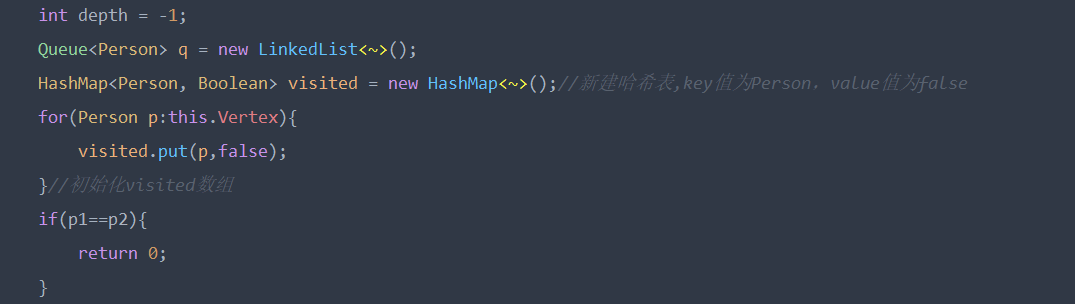
FriendshipGraph类主要的代码如下：



对于addVertex(Person p)函数，要先将现有人的名字和新准备加入的名字进行比较，如果相等，说明有名字重复，程序结束，如果不相等再加入。

对于addEdge(Person p1, Person p2)函数，要注意参数的传递顺序，是将p2存到p1的next列表中，这样可以方便扩展到有向图。

对于getDistance(Person p1, Person p2)函数，借助队列利用BFS算法求解从p1到p2的最短路径，当p1和p2相等时，直接返回0。先创建哈希表visited，其中key为每个人，value为boolean类型，如果该人已经被访问过了，value = true。

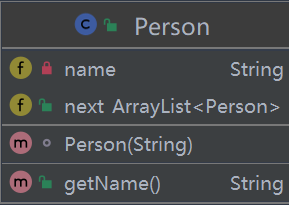


然后将p1入队，p1对应的value值变为true，然后开始BFS算法寻找最短路径：

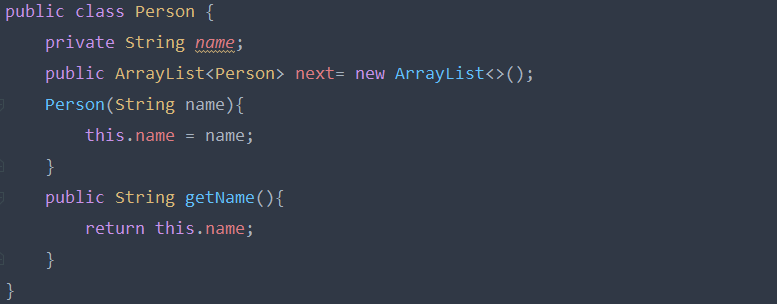


### 设计/实现Person类

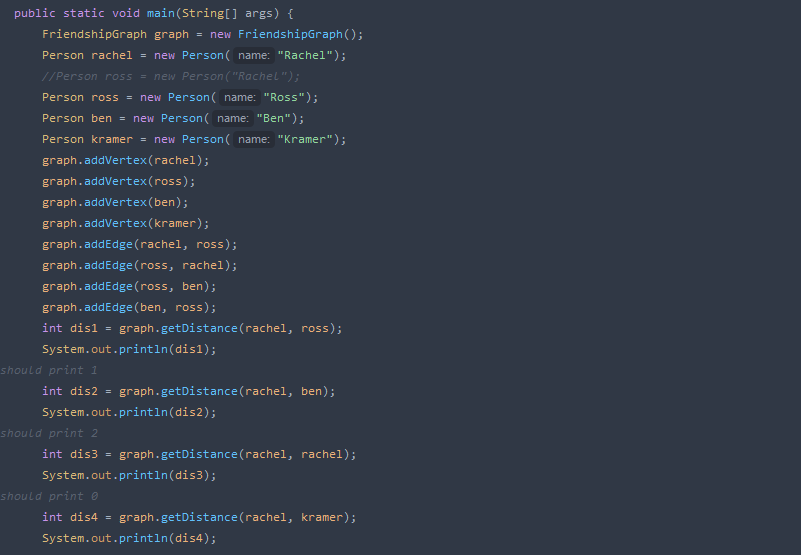
Person类对应的UML图如下：



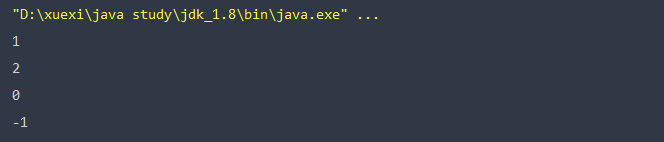
Person类的具体实现代码：



### 设计/实现客户端代码main()

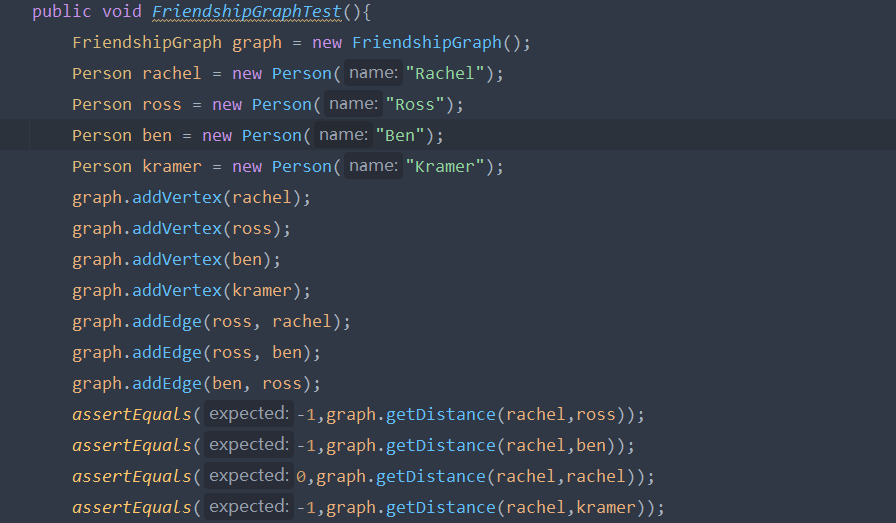


采用原本的样例，输入的结果为：

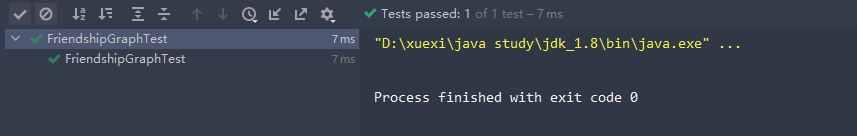


答案与预期相同。

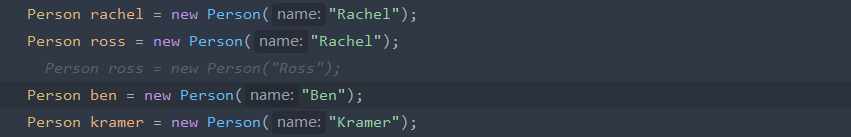
如果将原代码中的第10行注释掉（意即rachel和ross之间只存在单向的社交关系 ross->rachel），

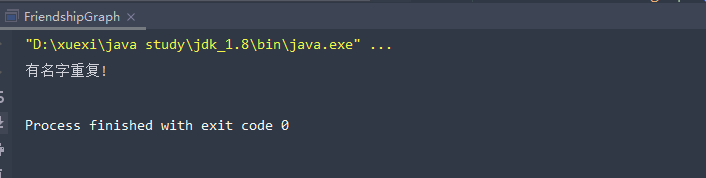


测试结果通过，与预期相同。



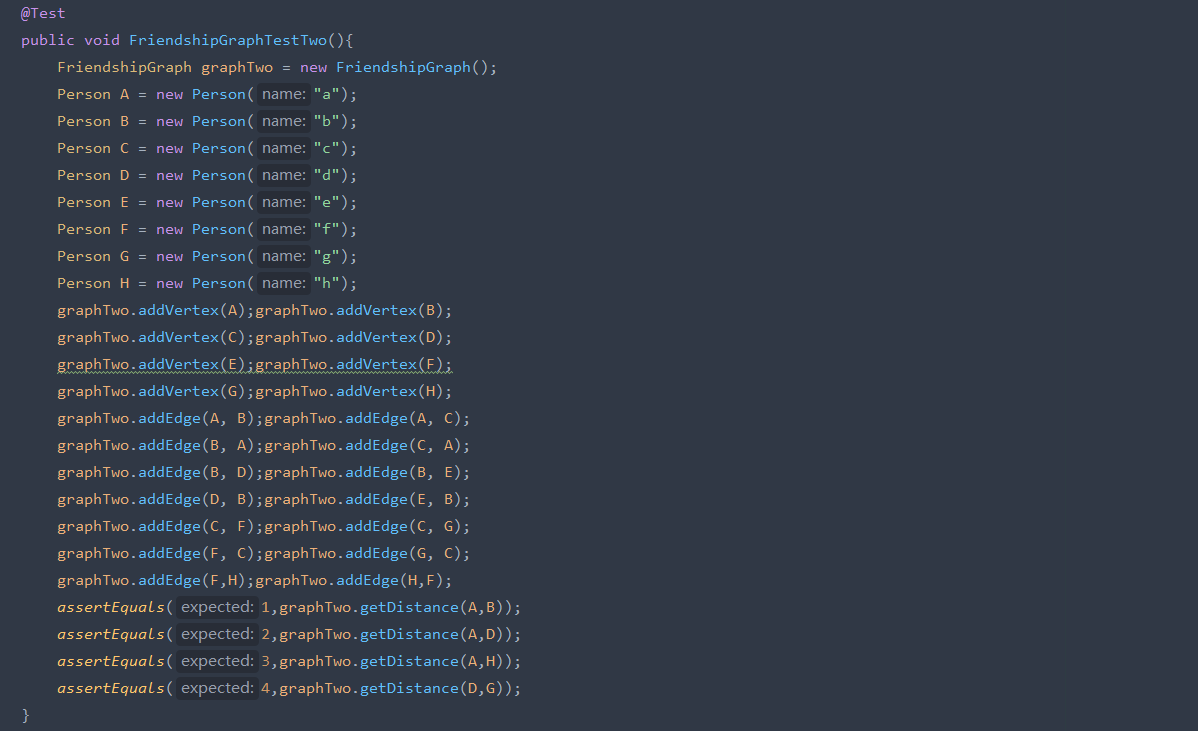
尝试着输入两个相同名字的人，



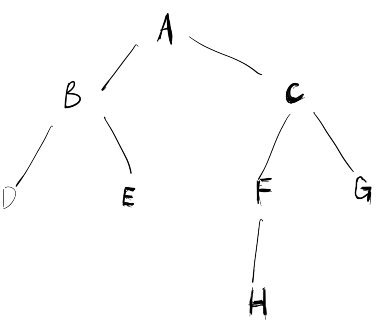


打印错误，程序直接结束。

### 设计/实现测试用例



关系图如下所示：



测试通过。

# 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。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 2021-05-20 | 15:45-17:45 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按计划完成 |
| 2021-05-20 | 19：00-21：00 | 分析了generateMagicSquare函数，研究了代码的内在逻辑 | 按计划完成 |
| 2021-05-20 | 21：00-22：00 | 对P1进行查漏补缺，增加了些注释 | 按计划完成 |
| 2021-05-21 | 15；45-17：45 | 看B站学习git基本操作 | 按计划完成 |
| 2021-05-21 | 19：00-20：00 | 完成P2的前几个问题 | 按计划完成 |
| 2021-05-21 | 20：00-21：00 | 研究calculateBearingToPoint函数 | 按计划完成 |
| 2021-05-21 | 21：00-23：00 | 完成P2的凸包和个人艺术 | 按计划完成 |
| 2021-05-22 | 10：00-11：00 | 分析P3中的类，设计UML图 | 按计划完成 |
| 2021-05-22 | 15；45-17：45 | 完成P3 | 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中的基本操作不会 | 查看B站教程并做好笔记 |
| BFS算法有点遗忘 | 回看以前的数据结构课件 |
| 凸包问题有点陌生 | 上网了解凸包问题 |

# 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

通过这次实验我熟悉了Java程序的基本编写方式，同时还学会了Git的基本操作，学会了如何在本地创建仓库，以及如何推送到远程的Github上。在完成实验的过程中也深刻感受到了自己平时敲代码的时间还不够，以后会多注重实践。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

很对胃口，写起来确实比C++容易上手。

1. 关于Eclipse IDE  
   之前感觉Eclipse使用起来不是很灵敏，所以这次使用了IDEA
2. 关于Git和GitHub

学会了基本的操作，感觉以后写代码更加方便

1. 关于CMU和MIT的作业

　　　向世界一流大学的计算机专业学习

1. 关于本实验的工作量、难度、deadline

工作量还行，难度适中，deadline如果和别的课冲突也是挺累的

1. 关于初接触“软件构造”课程

有了更多写代码的机会