

**2022年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 李世轩 |
| 学号 | 120L022109 |
| 班号 | 2003007 |
| 电子邮件 | [1146887979@qq.com](mailto:1146887979@qq.com) |
| 手机号码 | 15234117960 |

**目录**

[1 实验目标概述 1](#_Toc102161939)

[2 实验环境配置 1](#_Toc102161940)

[2.1 IntelliJ IDEA 1](#_Toc102161941)

[2.2 Junit配置过程 1](#_Toc102161942)

[2.3 Git的配置 3](#_Toc102161943)

[3 实验过程 4](#_Toc102161944)

[3.1 Magic Squares 4](#_Toc102161945)

[3.1.1 isLegalMagicSquare() 4](#_Toc102161946)

[3.1.2 generateMagicSquare() 5](#_Toc102161947)

[3.2 Turtle Graphics 6](#_Toc102161948)

[3.2.1 Problem 1: Clone and import 6](#_Toc102161949)

[3.2.2 Problem 3: Turtle graphics and drawSquare 6](#_Toc102161950)

[3.2.3 Problem 5: Drawing polygons 7](#_Toc102161951)

[3.2.4 Problem 6: Calculating Bearings 7](#_Toc102161952)

[3.2.5 Problem 7: Convex Hulls 8](#_Toc102161953)

[3.2.6 Problem 8: Personal art 9](#_Toc102161954)

[3.2.7 Submitting 10](#_Toc102161955)

[3.3 Social Network 10](#_Toc102161956)

[3.3.1 设计/实现FriendshipGraph类 10](#_Toc102161957)

[3.3.2 设计/实现Person类 11](#_Toc102161958)

[3.3.3 设计/实现客户端代码main() 11](#_Toc102161959)

[3.3.4 设计/实现测试用例 11](#_Toc102161960)

[4 实验进度记录 13](#_Toc102161961)

[5 实验过程中遇到的困难与解决途径 13](#_Toc102161962)

[6 实验过程中收获的经验、教训、感想 14](#_Toc102161963)

[6.1 实验过程中收获的经验和教训（必答） 14](#_Toc102161964)

[6.2 针对以下方面的感受（必答） 14](#_Toc102161965)

# 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。

# 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。

开发环境：IntelliJ IDEA

运行环境：IntelliJ IDEA

测试环境：Junit 4.13.1

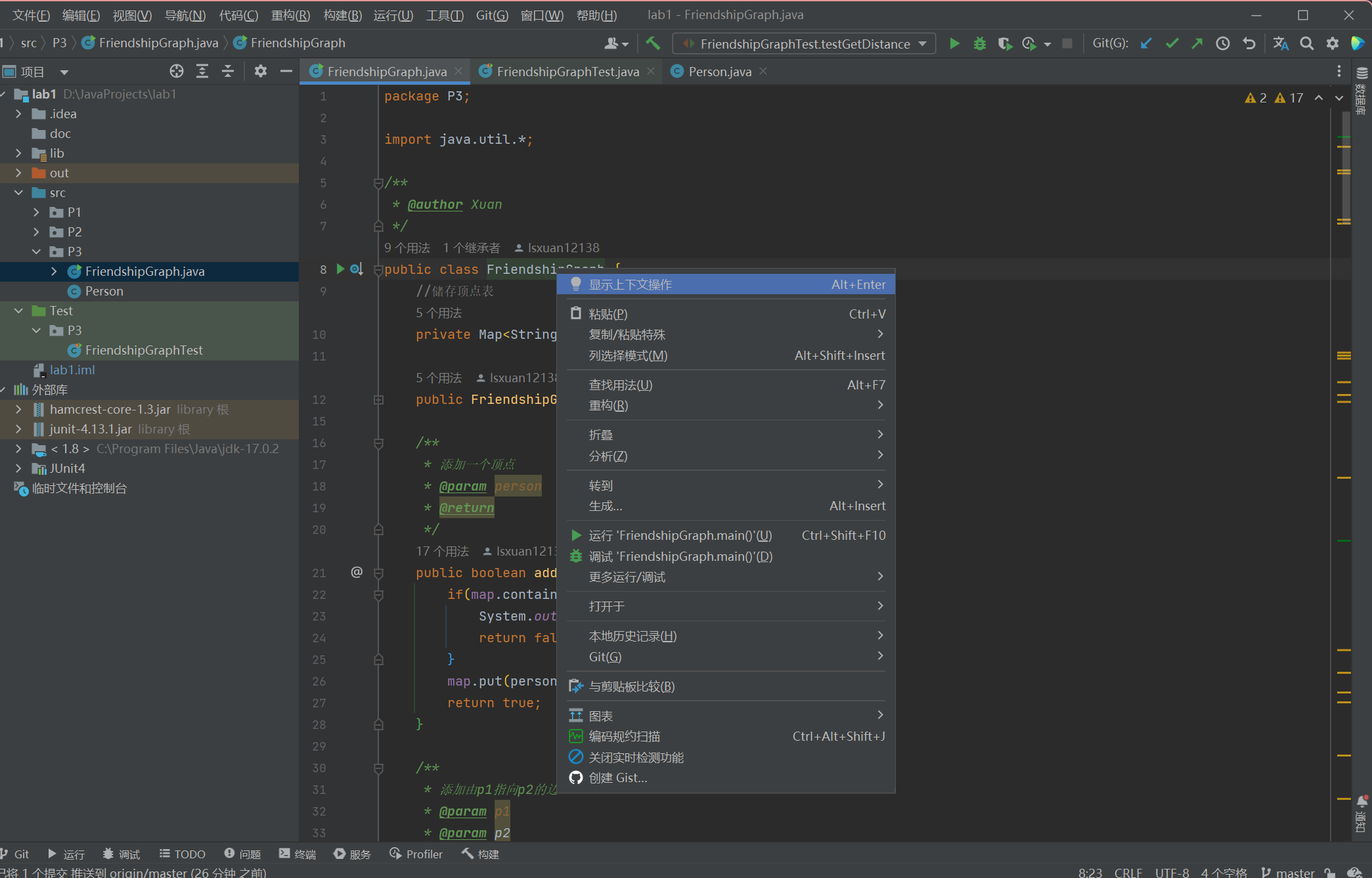
## IntelliJ IDEA

按照提示安装即可

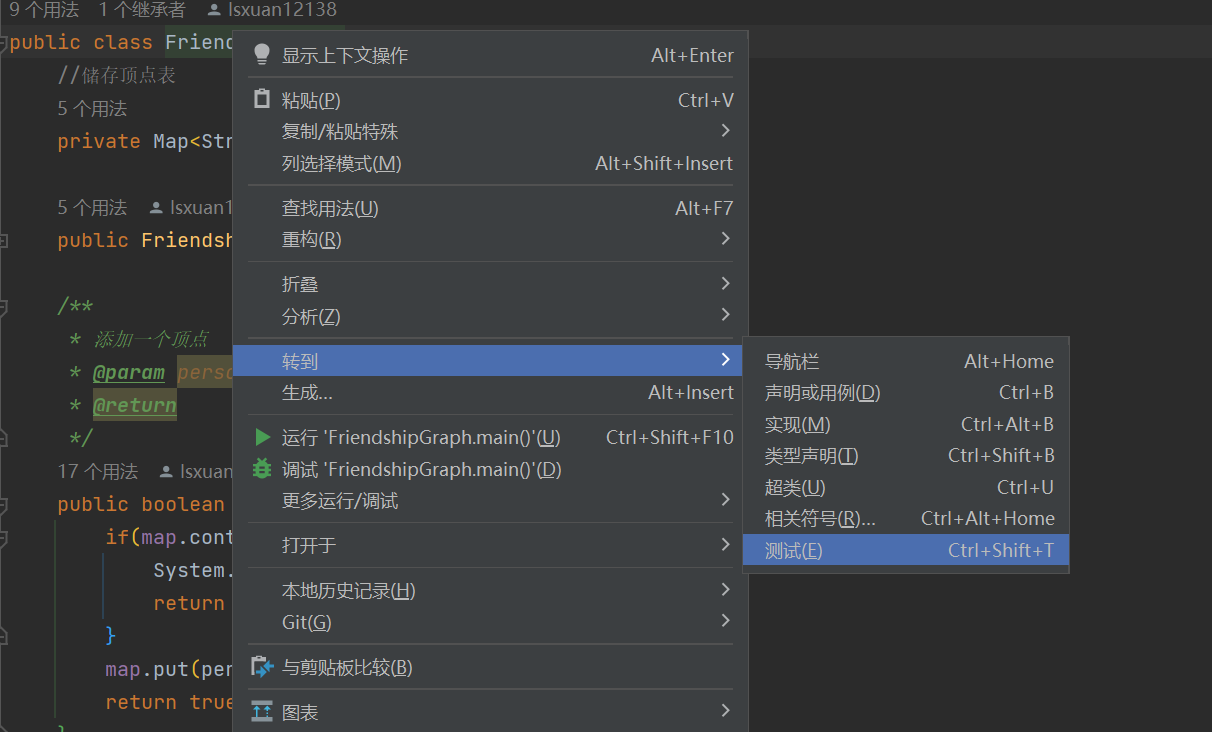
## Junit配置过程

因为IntelliJ IDEA内置了Junit的库，直接使用即可

-先在某一个需要测试的类中，对类名右键

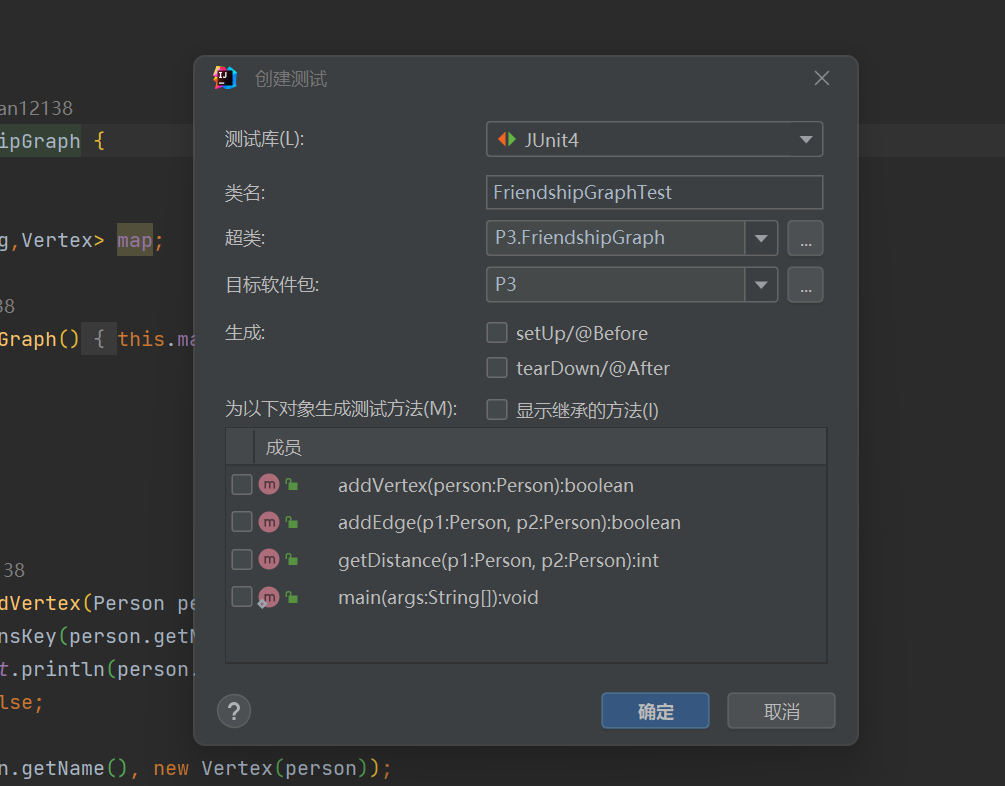


-在右键菜单中选择->转到->测试



-点击创建新测试





-在弹出的窗口中选择要测试的方法，即可生成测试与方法。

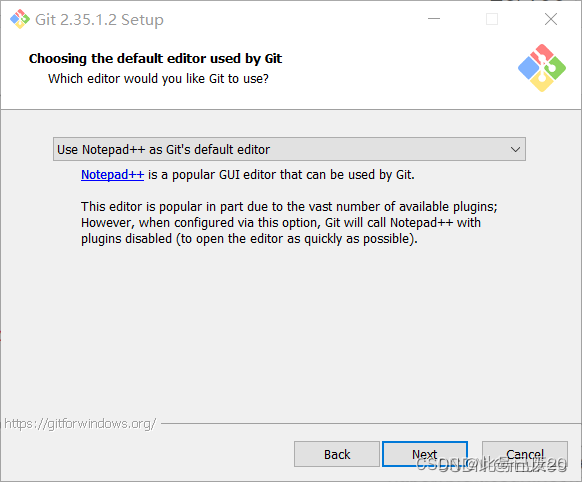
## Git的配置

安装过程中参考了一篇博客

<https://blog.csdn.net/mukes/article/details/115693833>

但是在安装过程中遇到一个问题

在其博客里*2.2.5 选择Git默认编辑器*中，我选择Sublime Textwe默认浏览器后。但是在完成编辑器的安装后，且配置完成环境变量后Git的安装界面中的Next并没有如博主一样亮起，无法点击。



解决办法是，若next未亮起，再多按几次Back，Next，返回前一界面再到这个界面，这个Next按钮就会亮起，然后就可以继续正常安装。

在这里给出你的GitHub Lab1仓库的URL地址。

https://github.com/ComputerScienceHIT/HIT-Lab1-120L022109

# 实验过程

## Magic Squares

幻方为一个每个横行，每个纵行，每个对角线上元素的和都相同的二位矩阵

该任务主要涉及两个问题，幻方的判断和生成。

### isLegalMagicSquare()

这个方法分为两个阶段

一个是从文件中读入幻方，一个是对读入的幻方进行判断。

首先对幻方进行读入，在这个过程中涉及对各种非法情况的处理

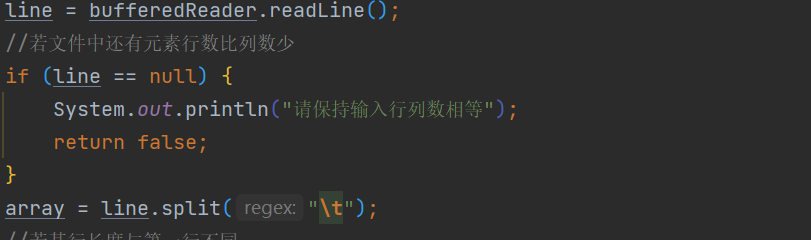
大致思路如下

先读入第一行，若其中不含有”\t”，说明它不符合对幻方的要求则直接返回，且输出报错信息。

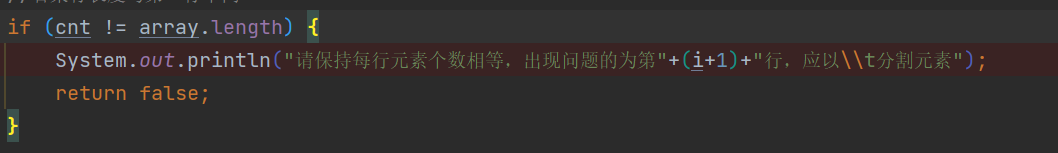
将读入的字符串split("\t")在第一行中我们可以得到该幻方的列数，以及正确的列数信息。

接下来按照行列数进行循环，每次读入一行，并将其处理放入准备的整数矩阵中，其中对一些非法信息进行处理

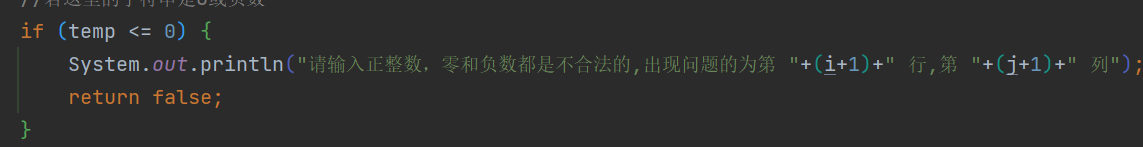
若在循环中读到line为null而循环还没有结束，说明文件中的行数比列数少



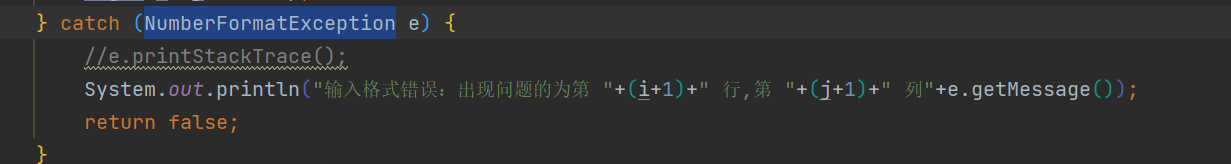
若分割后的数组长度与第一行元素个数不同，结束



若处理后得到非正数，则结束



若parseInt抛出NumberFormatException异常，说明输入的元素格式不符合整数



接下来对已经读入的幻方进行判断

先求得第一行元素的和

分别求取每行每列及对角线元素的和，若其与之前的值不同，返回false

### generateMagicSquare()

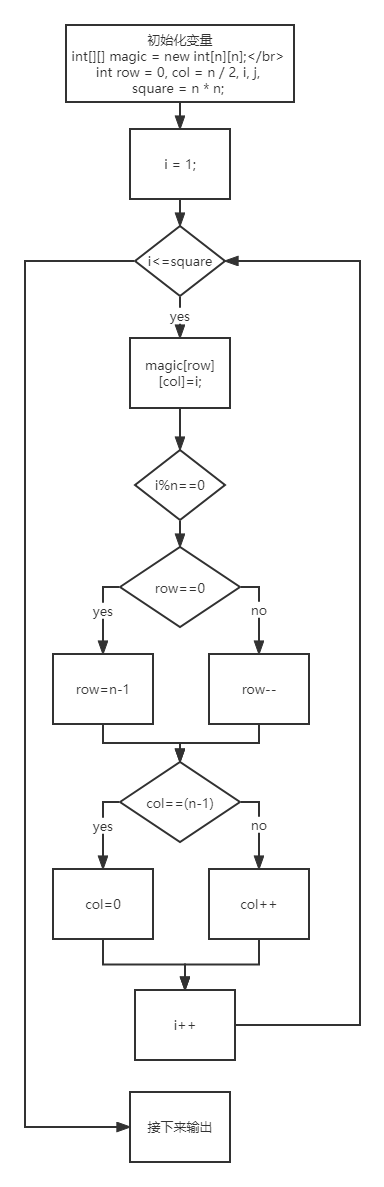
方法思路：

先初始化变量row = 0, col = n / 2, i, j, square = n \* n;

/共向矩阵中填写n\*n个数，从1到n\*n

让row从n-1到0从大到小不断循环，且初值为0，让col从0到n-1从小到大不断循环，且初值为n/2；

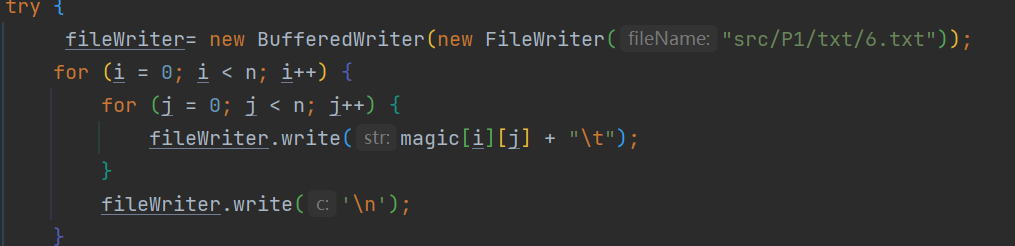
每放置n个数，row+1，防止覆盖前面已放置的数



对该函数做扩展：

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

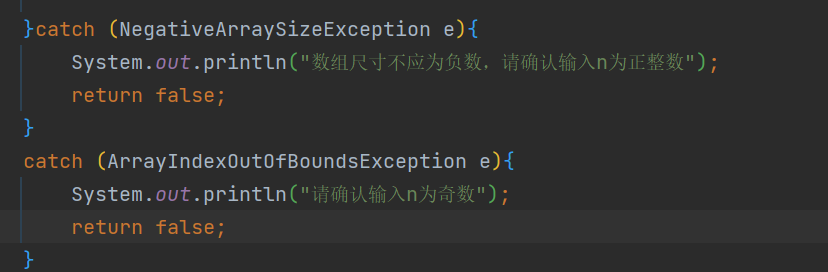
使用文件输出流，将产生的幻方写入文件



1. 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

将产生幻方的部分用try{}catch(){}包裹

Catch对应的异常



## Turtle Graphics

使用Java简单实现Logo语言中的Turtle 图形。

### Problem 1: Clone and import

如何从GitHub获取该任务的代码

在Git Bash中输入

git clone "仓库地址"

即可从GitHub

在本地创建git仓库

进入某一目录后，使用

git init

即可在本地创建git仓库

使用git管理本地开发。

添加到缓存区

git add <filename>

git add \*

提交

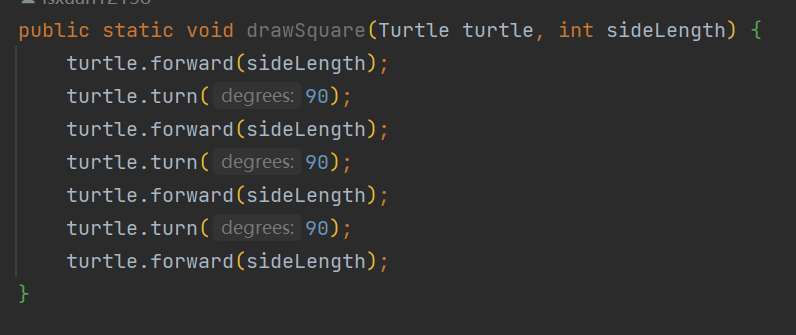
git commit -m “代码提交信息”

### Problem 3: Turtle graphics and drawSquare

该题为画一个正方形

向前sideLength后应该转90度

循环4次



### Problem 5: Drawing polygons

该题意为画一个正多边形，边数为sides，每个边的长度为sideLength

首先计算内角，使用方法

angle = calculateRegularPolygonAngle(sides)

其实现思路为

我们知道在转过sides个弯后，应该转过360度

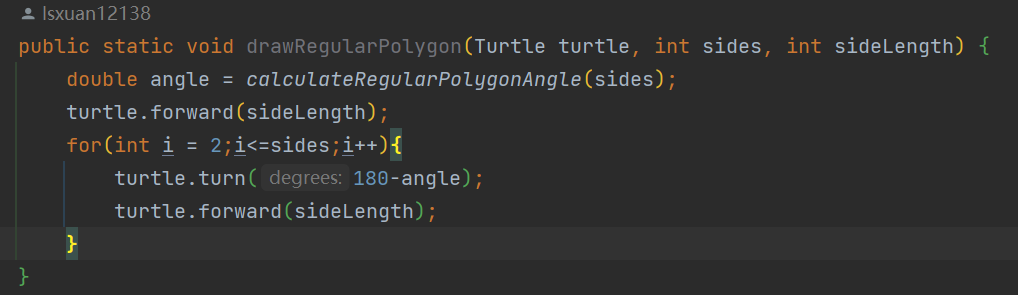
所以内角的补角应该为360/sides

所以内角大小为180-360/sides

在得到内角角度后

开始画正多边形

循环sides次，每次转弯angle度，向前走sideLength



### Problem 6: Calculating Bearings

public static List<Double> calculateBearings(List<Integer> xCoords, List<Integer> yCoords)

该题意为在若干个点之间按顺序移动，求每两个点之间应该转动的角度为多少

其中调用的子方法为double calculateBearingToPoint(double currentBearing, int currentX, int currentY, int targetX, int targetY)

子方法实现的思路为

先计算 current到target直接向量的方向角，相对于正北，以顺时针为正

然后减去当前角度angle即可

定义angle为当前角度，初始值为0，delta为变化角度

先获取点的个数，确定循环次数为个数-1，接下来开始循环，从第一个点开始计算

计算当前点到下一个点的角度，将其装入结果

angle+=deltaDegree; angle%=360;获取转过角度后的当前角度

循环完成后返回结果



### Problem 7: Convex Hulls

该题为计算若干点的闭包

使用Graham(格拉翰)扫描法

思路为

1把所有点放在二维坐标系中，则纵坐标最小的点一定是凸包上的点，记为P0。

2把所有点的坐标平移一下，使P0作为原点。

3计算各个点相对于P0的幅角 α ，按从小到大的顺序对各个点排序。当 α 相同时，距离P0比较近的排在前面。假设得到的结果为P1，P2，P3，P4，P5，P6，P7，P8。我们由几何知识可以知道，结果中第一个点P1和最后一个点P8一定是凸包上的点。

4以上，我们已经知道了凸包上的第一个点P0和第二个点P1，我们把它们放在栈里面。现在从步骤3求得的那个结果里，把P1后面的那个点拿出来做当前点，即P2。接下来开始找第三个点。

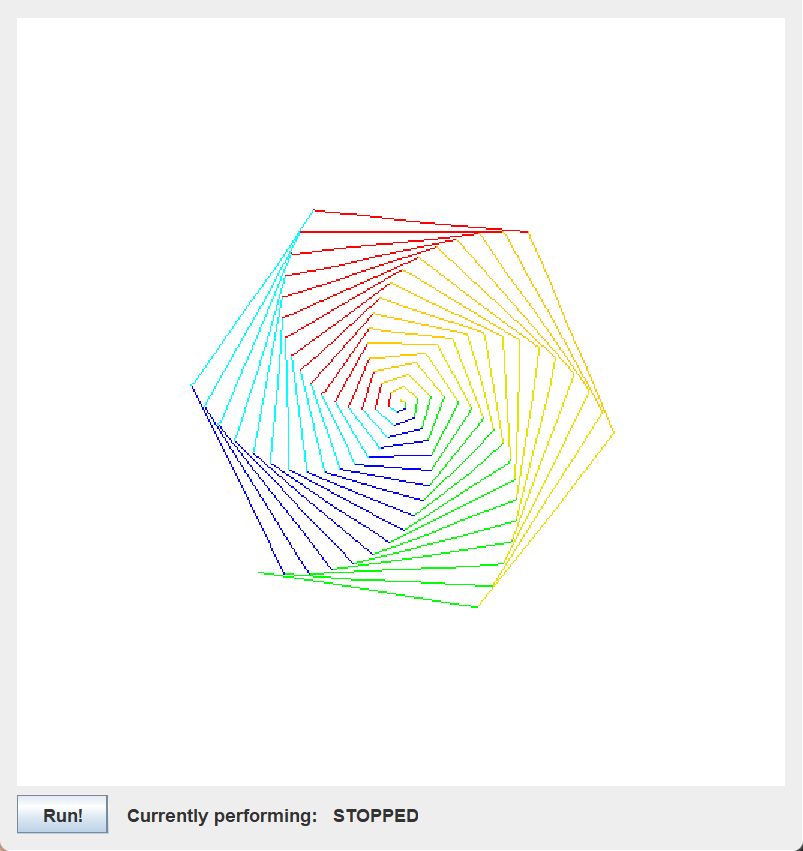
5连接栈最上面的两个元素，得到直线L。看当前点是在直线L的右边还是左边。如果在直线的右边就执行步骤6；如果在直线上，或者在直线的左边就执行步骤7。

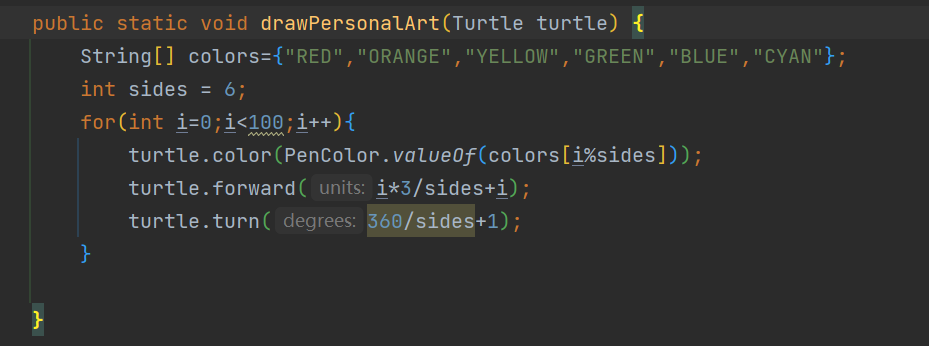
6如果在右边，则栈顶的那个元素不是凸包上的点，把栈顶元素出栈。执行步骤5。

7当前点是凸包上的点，把它压入栈，执行步骤8。

8检查当前的点是不是步骤3那个结果的最后一个元素。是最后一个元素的话就结束。如果不是的话就把当前点后面那个点做当前点，返回步骤5。

### Problem 8: Personal art





### Submitting

使用指令

git push origin master

其中的master可以换为任何其他分支

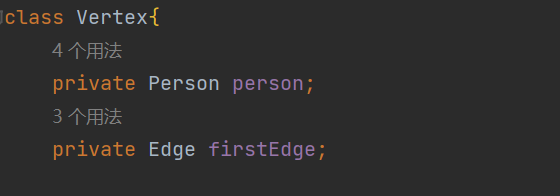
## Social Network

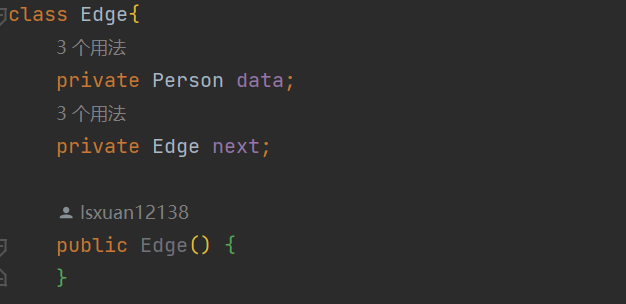
使用有向图模拟社交网络，其中结点为人。

实现对有向图添加节点，添加边，获取两个节点间最短路径长度的方法

### 设计/实现FriendshipGraph类

构建两个子类





使用HashMap<String,Vertex>来存储图。

#### addVertex(Person person)方法实现

向map中添加键值对< person.getName(), new Vertex(person)>

#### addEdge(Person p1,Person p2)方法实现

先检查顶点的firstEdge是否为null，若是，直接设置vertex.setFirstEdge(new Edge(p2));

若不是，沿着链表向后找到末尾后edge.setNext(new Edge(p2));

#### getDistance(Person p1,Person p2)方法实现

以图的广度优先遍历为基础

以cnt记录访问的深度，即最短路径长度

在遍历过程加入空节点作为标志，表示访问完一层的节点，使cnt加一

若遍历过程未返回，说明p2与p1未连接，返回-1

### 设计/实现Person类

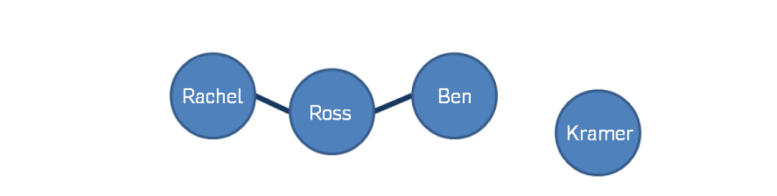
因为当前需要的唯一变量为名字

定义成员变量private String name;

实现相应构造方法，Setter and Getter

### 设计/实现客户端代码main()

使用使用指导书中的代码。构造这样的图



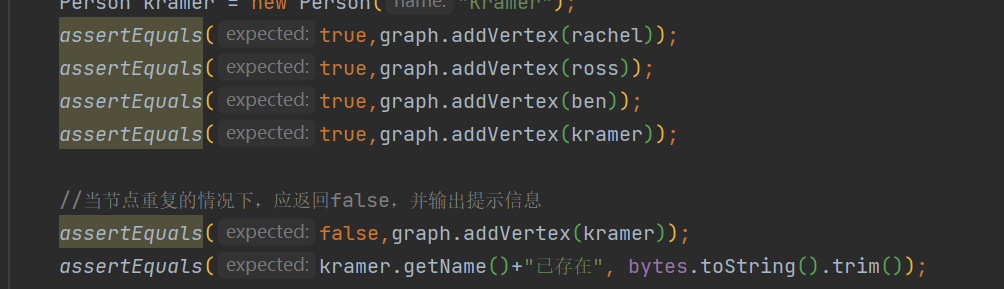
并根据其中代码进行测试

### 设计/实现测试用例

#### 测试testAddVertex

测试正常添加节点的返回值是否正确

测试当节点重复添加时的反应

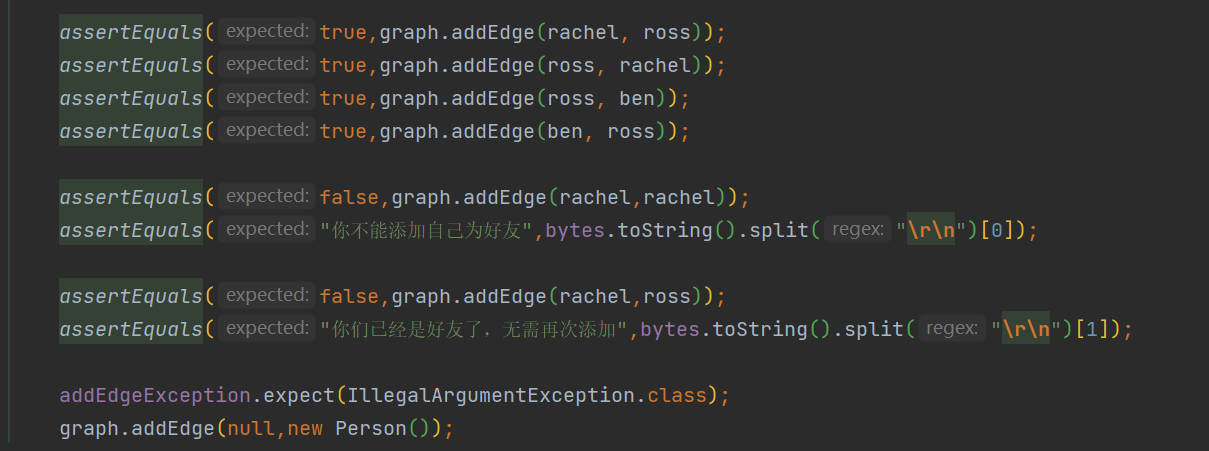


#### 测试testAddEdge

测试正常添加边的返回值是否正确

测试当p1，p2相同时，程序的反应

测试重复添加边时的反应



#### 测试testGetDistance

测试当两个点间只有一条有向边的反应

测试两个点间有正常的两条边的结果是否正确



# 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 |
| 2022-04-27 | 18:30-19:30 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按计划完成 |
| 2022-04-27 | 17:30-20:00 | 完成问题2 | 按计划完成 |
| 2022-04-27 | 20:30-21:00 | 完成问题2中1，3，5 | 完成 |
| 2022-04-27 | 21:30-22:30 | 完成问题2中6 | 完成 |
| 2022-04-28 | 14:30-15:30 | 完成问题2中7 | 未完成，延长 |
|  | 15:30-17:30 | 完成问题2中7 | 未完成，延长 |
|  | 18:00-19:00: | 完成问题2中7 | 完成 |
| 2022-4-28 | 20:00-20:30 | 完成问题3中两个类基本构建 | 完成 |
| 2022-4-28 | 20:30-21:00 | 完成问题3中addVertex | 完成 |
| 2022-4-28 | 21:00-21:30 | 完成问题3中addEdge | 完成 |
| 2022-4-28 | 21:30-22:30 | 完成问题3中getDistance | 完成 |
| 2022-4-29 | 14:00-15:30 | 完成问题3中Test部分，并修改bug | 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 |
| 求凸包的算法  gift-wrapping algorithm没看懂 | 改用Graham(格拉翰)扫描法 |
| Graham(格拉翰)扫描法实现过程中涉及到求点的极坐标并排序 | 先用Map存储点Point对应的极角  在根据冒泡排序去排序并放到一个List中 |
| 修改你的FriendshipGraph类和Person 类，使该约束能够始终被满足（意即：一旦该条件被违反，提示出错并结束程序运行）。 | 在Person中添加一个静态成员变量public static Set<String> namePool来记录所有出现过的人名，当再次new Person对象时，先进行坚持，若已经出现过，直接结束程序运行 |

# 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）

对一个方法的测试一定要做的全面，否则在之后其他方法中调用它出问题时，debug会非常困难

## 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？

因为我在高考结束后自学java，所以其实Java是我接触的第一个编程语言，所以很对我口味

与其他编程语言相比，Java运行在JVM上，所以java程序的可移植性非常强，可以说是即插即用。Java本身提供一个非常大的类库，我们常用的方法不需要再去找第三方库。Java完全规避了指针，虽然想要深入理解还是得明白指针的用法，但是程序员不再需要担心内存泄漏等问题

Java的缺点就是，相对其他语言，因为它在虚拟机上运行，所以处理速度相等慢很多

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

IntelliJ IDEA是一个非常强大的IDE，他除了基本的代码补全等功能，还可以开发maven项目等等大型项目。其调试功能也很强大。还可以在其中配置Git等等方便开发的功能

不足：正是由于它的强大，如果过分依赖它，非常不利于程序员的进步

1. 关于Git和GitHub，是否感受到了它在版本控制方面的价值；

Git和Github可以让我们很方便的看到各个版本的代码变化，以及开发进度

1. 关于CMU和MIT的作业，你有何感受；

国外顶尖大学的作业不仅难度较大，而且非常贴近实际，能很好的考察和锻炼程序员的能力

1. 关于本实验的工作量、难度、deadline；

工作量比较大，难度也比较大，有些需要查阅资料，并且细致的思考

Deadline比较合理

1. 关于初接触“软件构造”课程；

软件构造课程让我对软件开发有了一个跟全面的认识，从过去完全在做数学题，或是模仿数据结构到了更高的领域