

**2019年春季学期  
计算机学院《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 任永鹏 |
| 学号 | 1173710107 |
| 班号 | 1737101 |
| 电子邮件 | [805402160@qq.com](mailto:805402160@qq.com) |
| 手机号码 | 18800420869 |

**目录**

[1 实验目标概述 1](#_Toc3043453)

[2 实验环境配置 1](#_Toc3043454)

[3 实验过程 1](#_Toc3043455)

[3.1 Magic Squares 1](#_Toc3043456)

[3.1.1 isLegalMagicSquare() 1](#_Toc3043457)

[3.1.2 generateMagicSquare() 2](#_Toc3043458)

[3.2 Turtle Graphics 4](#_Toc3043459)

[3.2.1 Problem 1: Clone and import 5](#_Toc3043460)

[3.2.2 Problem 3: Turtle graphics and drawSquare 5](#_Toc3043461)

[3.2.3 Problem 5: Drawing polygons 6](#_Toc3043462)

[3.2.4 Problem 6: Calculating Bearings 7](#_Toc3043463)

[3.2.5 Problem 7: Convex Hulls 8](#_Toc3043464)

[3.2.6 Problem 8: Personal art 8](#_Toc3043465)

[3.2.7 Submitting 9](#_Toc3043466)

[3.3 Social Network 9](#_Toc3043467)

[3.3.1 设计/实现FriendshipGraph类 9](#_Toc3043468)

[3.3.2 设计/实现Person类 10](#_Toc3043469)

[3.3.3 设计/实现客户端代码main() 10](#_Toc3043470)

[3.3.4 设计/实现测试用例 10](#_Toc3043471)

[3.4 Tweet Tweet 10](#_Toc3043472)

[3.4.1 Problem 1: Extracting data from tweets 10](#_Toc3043473)

[3.4.2 Problem 2: Filtering lists of tweets 11](#_Toc3043474)

[3.4.3 Problem 3: Inferring a social network 11](#_Toc3043475)

[3.4.4 Problem 4: Get smarter 11](#_Toc3043476)

[4 实验进度记录 12](#_Toc3043477)

[5 实验过程中遇到的困难与解决途径 12](#_Toc3043478)

[6 实验过程中收获的经验、教训、感想 13](#_Toc3043479)

[6.1 实验过程中收获的经验和教训 13](#_Toc3043480)

[6.2 针对以下方面的感受 13](#_Toc3043481)

# 实验目标概述

本次实验通过求解四个问题，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。 λ 基本的 Java OO 编程 λ 基于 Eclipse IDE 进行 Java 编程 λ 基于 JUnit 的测试 λ 基于 Git 的代码配置管理

# 实验环境配置

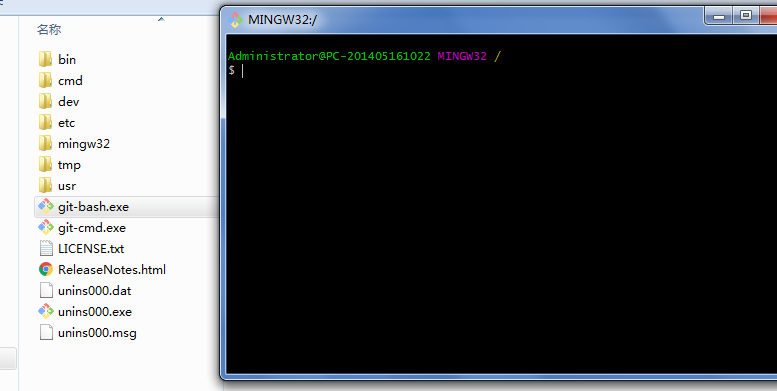
简要陈述你配置本次实验所需开发、测试、运行环境的过程，必要时可以给出屏幕截图。

特别是要记录配置过程中遇到的问题和困难，以及如何解决的。

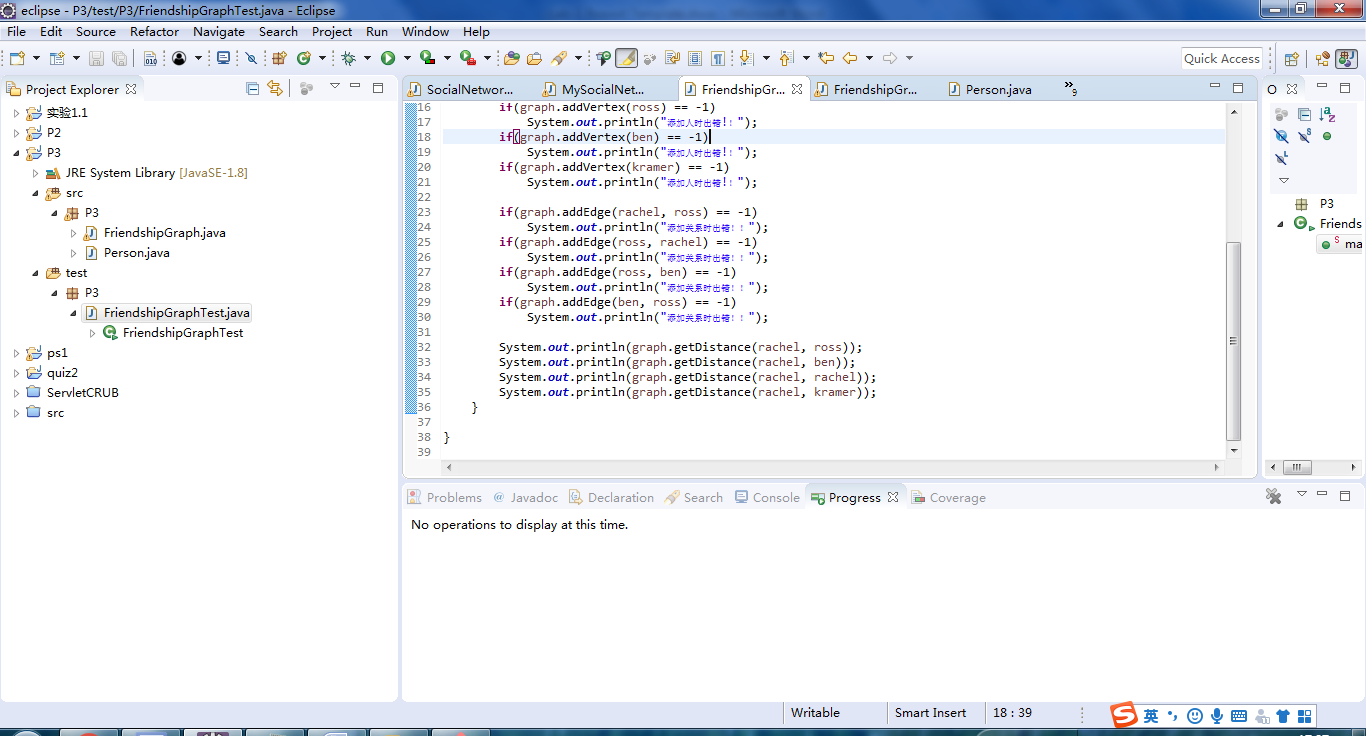
1安装Java

从Java Development Kit 10下载，下载Java SE Development Kit 10.0.2并运行安装程序。此处需要配置环境变量。

2安装Git



3安装eclipse



在这里给出你的GitHub Lab1仓库的URL地址（Lab1-学号）。

https://github.com/ComputerScienceHIT/Lab1-1173710107.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

在这里简要概述你对该任务的理解。

该任务主要是让我们把五个txt文件读入，并检测是否符合方阵的判断条件，进而通过一系列的计算判断该方阵是否是幻方，在控制台输出结果。之后让我们将任务中给的函数添加进去，并仔细分析该函数是如何根据输入的n产生幻方的。由于该函数产生的是奇数阶的幻方，因此当n输入偶数或者负数的情况时，会抛出异常，我们需要分析异常出现的原因，并对于该异常做出合理的程序改进。

### isLegalMagicSquare()

按步骤给出你的设计和实现思路/过程/结果。

先将文件中的信息按行读入到arraylist中，计数器count初始化为0，每读入一行信息，count++，然后将每行的信息以\t为分隔符分开，并记录分隔后当前行的数组的长度，并和第一行的数组长度进行比较。

如果不相同，则证明文件中的当前行或者是元素与之前的行元素不相同，或者是分隔符并非是\t，总之这两种情况不符合实验的要求，控制台输出错误提示，函数返回false。

如果满足上面的条件，则进入下一步，判断当前的字符串是否全部都是由0-9的字符组成的，如果不是，表明当前的数有可能存在负数，或者小数，不符合实验要求，控制台输出错误提示，函数返回false。

读入文件结束之后，判断count与第一行的数组的个数是否相同，不相同的话，控制台输出报错信息，函数返回false。

相同的话，利用I,j的两层循环，对每行和每列进行累加

sum[i] = sum[i] + list.get(count\*i+j);//行累加

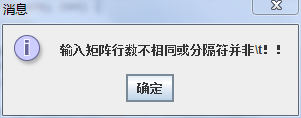
sum[i+count] = sum[i+count] + list.get(count\*j+i);//列累加

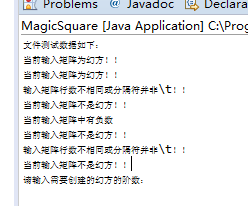
然后单独对左对角线和右对角线进行相加。

先将两个对角线的累加值和第一行的累加值比较，然后以第一行累加值为参照，依次比较之后的累加值，只要有依次不相同，证明不是幻方，函数返回false，当全部相同的时候，证明当前的方阵是幻方，控制台输入正确的信息，函数返回true.。

下面是运行的结果：

当错误时，会有弹窗提示



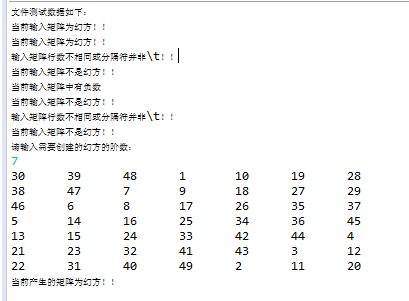


### generateMagicSquare()

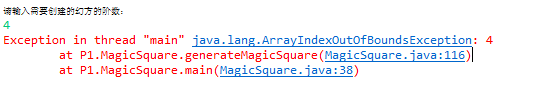
按步骤给出你的设计和实现思路/过程/结果。

将产生的方阵按照上面文件的格式输出到文件中，并调用*isLegalMagicSquare()*来通过返回值进行判断上面产生的方阵是不是幻方，根据返回值在工作台输出不同的提示结果！

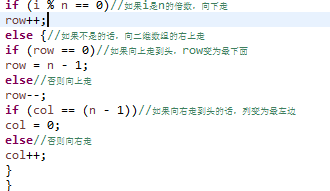
下面是n=7的结果图：



输入错误分析：

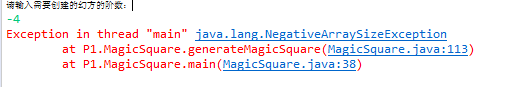
1. 

当输入的n为偶数时，会出现上面的错误提示，经过查询可以知道该提示的意思是数组越界，分析代码，理解幻方的构造原理之后



可以知道，在最外面的if-else语句中，else部分进行了边界的限制，不可能出现数组越界的情况，唯一的解释就是在if的条件下出现了问题，根据奇数阶的幻方构建规则可知，当i是阶数的整数倍，也就是余数等于零时，坐标向下移一行，那么当n为偶数时候，就有可能会出现下面情况：此时row已经是最后一行，又符合if中的条件，那么row就会出现等于n的情况，进一步进行循环时，数组越界的问题就会存在了。下面是改进后的结果：



1. 

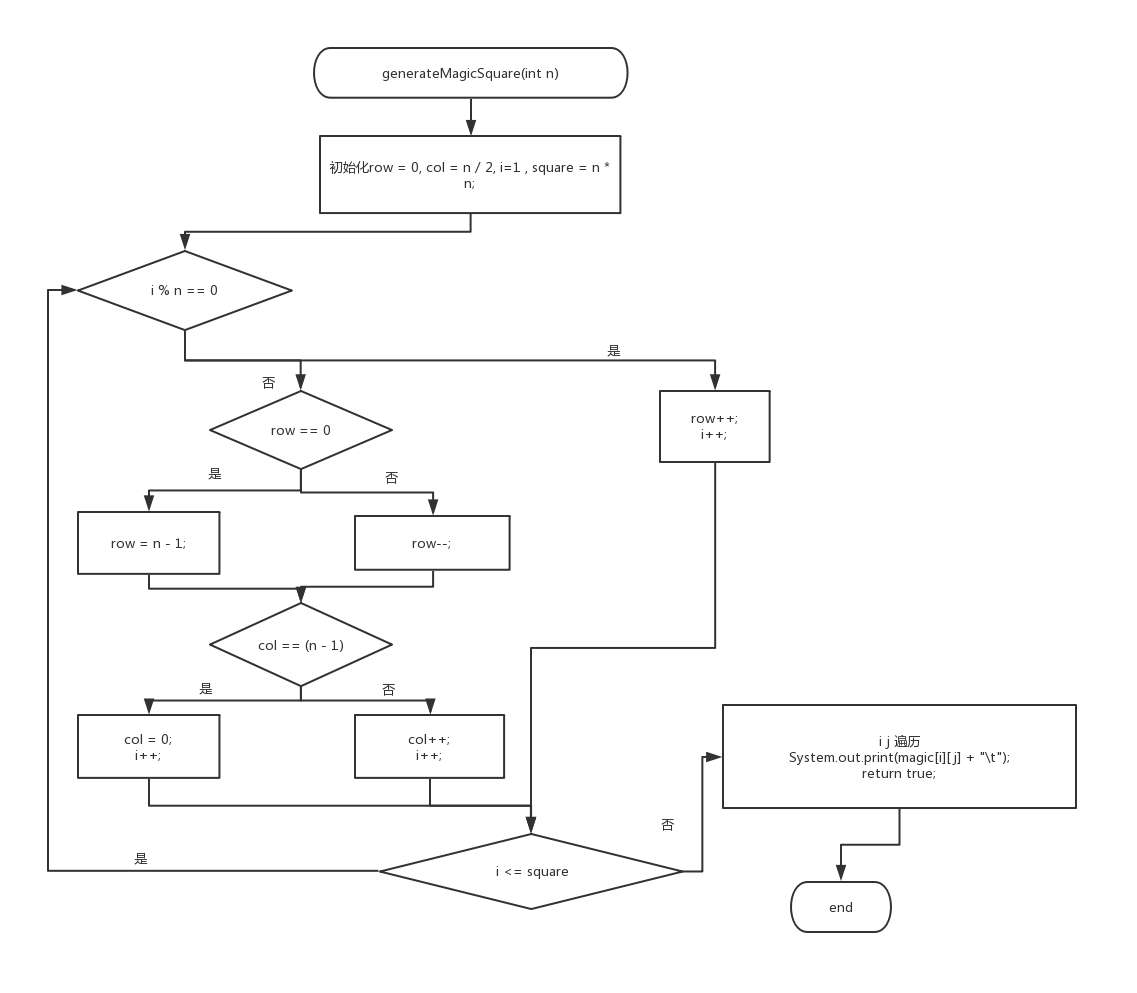
当输入的阶数为负数时，会出现上面的报错，通过查询可以知道错误信息意思为数组长度为负数，或者是数组下标为负数，这个问题的原因很好找，根据下面的代码：



可以看到一开始创建的magic数组是通过输入的n来定义大小的，只要通过限定输入的n的大小就可以避免上面问题的出现，下面是修改之后的结果：



该函数的流程框图如下：



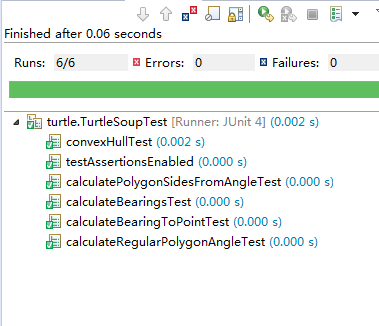
## Turtle Graphics

在这里简要概述你对该任务的理解。

该任务主要是让我们从github上面获取代码，在本地建立git仓库，然后自己写代码解决一系列的问题：

1. 问题3主要是让我们实现forward和turn这两个功能。
2. 问题5是让我们实现calculateRegularPolygonAngle，drawRegularPolygon，并且第一次尝试使用Junit进行测试。
3. 问题6是让我们实现calculateBearingToPoint和calculateBearings两个函数的功能，并且测试成功。
4. 问题7是让我们首先了解凸壳的概念，并使用Gift wrapping algorithm算法对一系列的点进行计算，返回正确的点通过测试。
5. 问题8是让我们自己使用forward和turn的两个功能并使用for循环等画出自己 的设计的图案，并可以对线条进行代码中有的颜色的改变。

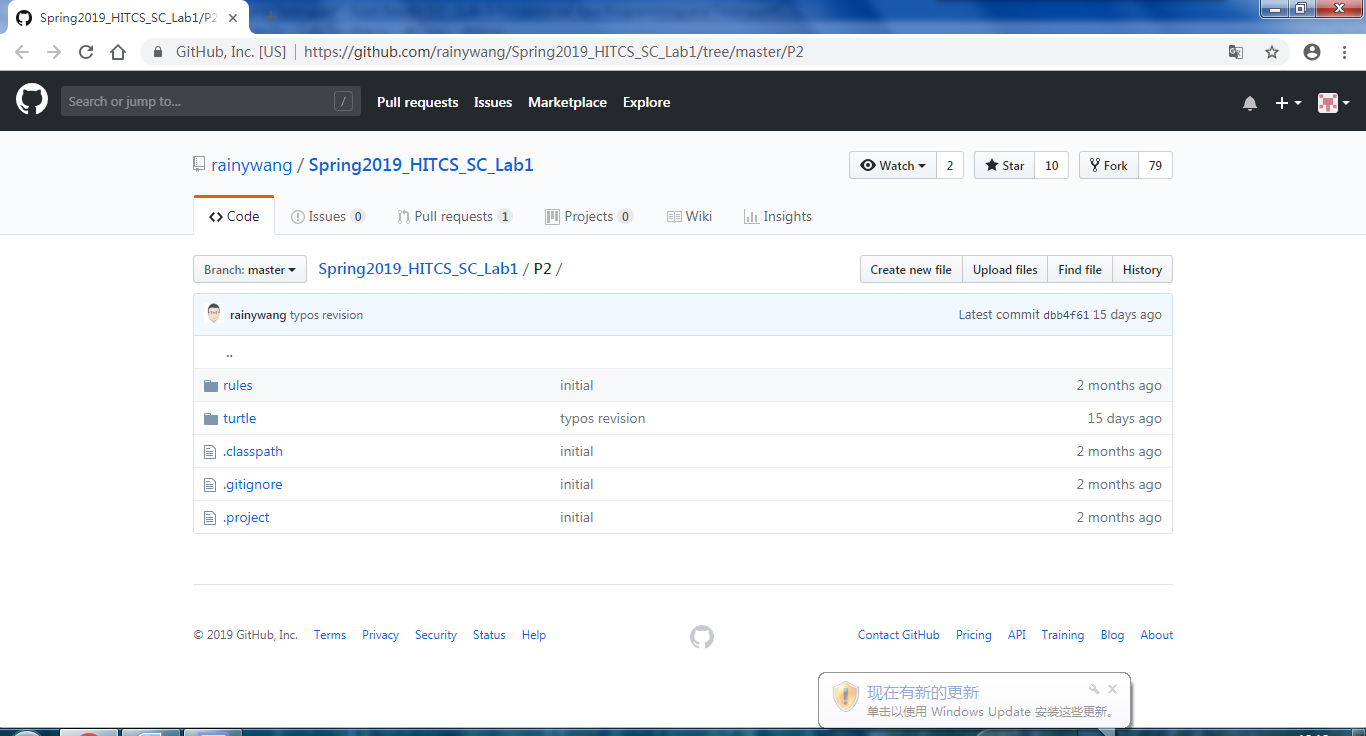
下图是通过测试的截图：



### 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。

从下面获取代码



打开git bash

输入命令git clone +网址将github上面的仓库克隆到本地

下面是上传文件的命令

git add .        （注：别忘记后面的.，此操作是把Test文件夹下面的文件都添加进来）

git commit  -m  ”提交信息”

git push -u origin master   （注：此操作目的是把本地仓库push到github上面，此步骤需要输入帐号和密码）

### Problem 3: Turtle graphics and drawSquare

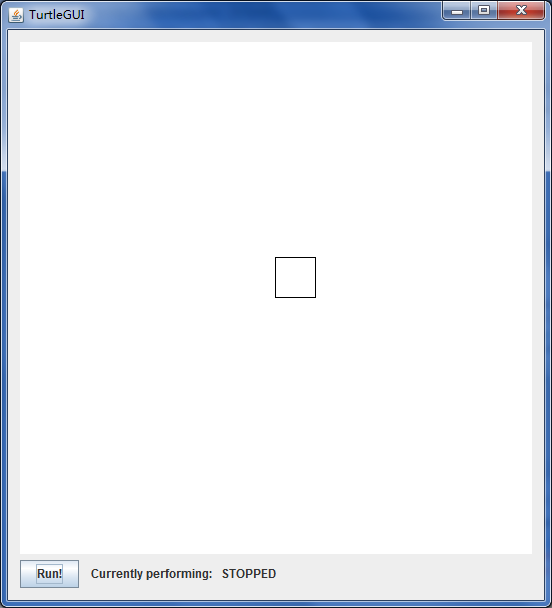
本题需要首先了解Turtle的工具包，然后根据所给的参数画出一个正方形。

首先先用turtle.draw();给出一个画布，然后用turtle.forward(sideLength)

表示线条向当前的方向前进sideLength个长度，之后使用turtle.turn(90.00);

表示方向向顺时针方向旋转90度，依次使用上面的功能就可以画出一个正方形，具体见代码。

代码实现图如下：



### Problem 5: Drawing polygons

实现下面的函数：

public static double calculateRegularPolygonAngle(int sides)

从上面可以知道函数需要根据给定的参数sides返回该多边形的内角。

该函数需要通过多边形的外角和是360度来计算。具体见函数。

实现函数

public static void drawRegularPolygon(Turtle turtle, int sides, int sideLength)

该函数是需要通过给出的sides sideLength两个参数，通过上面实现的函数来实现该多边形的画出。

首先通过calculateRegularPolygonAngle函数计算出当前多边形的内角，之后通过问题3 的画多边形的经验画出。

### Problem 6: Calculating Bearings

该问题首先需要实现 public static double calculateBearingToPoint(double currentBearing, int currentX, int currentY,int targetX, int targetY)函数

该函数需要首先通过比较currentX和targetX两个的值，

如果两个相同的情况下m被赋值为90度，然后比较currentY和targetY两个的值，

如果两者相同就证明两个点是重合的，那么就返回10000000这一个特定的值来特殊标记这一情况。

如果Y对应的不相同的话，通过函数来计算出两个点练成的直线的斜率计算出直线与x正方向的夹角，赋值给m，然后通过目标点是在当前点的左边还是右边计算出相应的转动的角度，最后通过 判断将角度控制在0-360度之间，返回该角度。

然后实现下面函数：

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

该函数需要通过每次取List中的x与y，然后计算出相应的角度，把所有的角度放到List中，作为返回值。这里的主要功能在上个函数就实现了，只需要依次取出xy，然后反复调用上面的函数，将返回的转动的角度加入到一开始声明的数组里就可以了，这里需要注意的是每次调用函数时，当前点的角度的变化。这里需要第一次初始化为0，之后每次需要加上对应的角度，如果角度大于360度，就需要减一个360度。

### Problem 7: Convex Hulls

该任务需要实现下面的函数：

public static Set<Point> convexHull(Set<Point> points)

该函数的功能是需要通过给定的一系列的点，算出组成凸包的点，作为返回值返回。

实现功能的步骤如下，首先判断所给点的个数，如果没有点，直接返回空的，

否则判断所给点集合的个数，如果点小于等于3个点，直接将三个点返回，

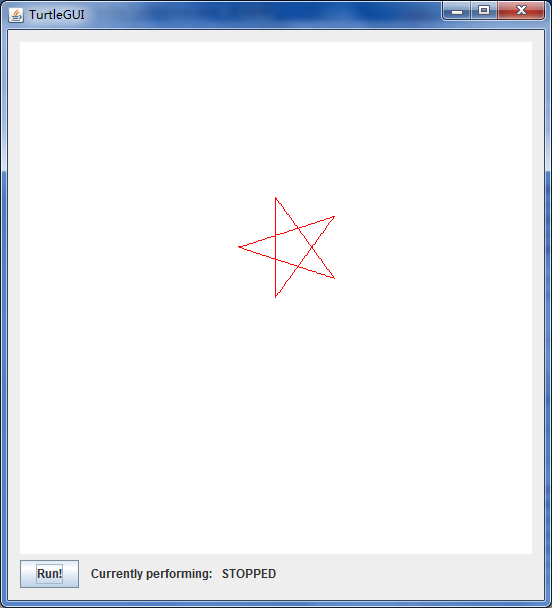
否则寻找一个xy坐标都是最小的点，方向为0度，然后遍历一遍所有的点，利用之前写好的函数计算出偏移的角度，保留偏移角度最小的一个点，当遇到两个点的偏移角度一样大小的时候，说明此时三个点都在一条直线上，通过二维平面的距离计算公式计算出这两个点到当前点距离，保留距离较大的点。

将保留的点加入到返回的点集中，然后将这个点作为当前点，继续循环进行上面的操作，循环的跳出条件是当前点和最初的点相同，此时点集就添加完毕，将这个点集返回就是所求。

### Problem 8: Personal art

这里我自己设计的是一个用红笔描绘的五角星，首先利用.color把颜色设置为红色，然后用for循环重复五次以下操作，turtle.forward(100);

turtle.turn(144.00);，就可以画五角星了。



### Submitting

如何通过Git提交当前版本到GitHub上你的Lab1仓库。

git add .        （注：别忘记后面的.，此操作是把Test文件夹下面的文件都添加进来）

git commit  -m  ”提交信息”

git push -u origin master   （注：此操作目的是把本地仓库push到github上面，此步骤需要输入帐号和密码）

## Social Network

在这里简要概述你对该任务的理解。

该任务需要我自己新建一个Person类,并且构造有向图实现对于人际关系的存储，具体的功能包括对人名的添加，对两个人关系的添加，对于两个人关系距离的计算等等。

### 设计/实现FriendshipGraph类

这个类主要包括addVertex，addEdge，getDistance

addVertex函数通过参数name，首先判断name是不是null，是的话返-1，不是的话添加进去，返回1

addEdge函数通过先判断两个参数是不是null，只要有一个是，就返回-1，否则在第一个人名的friends中添加第二个人。

getDistance函数初始化distance=0判断两个名字是否相同，相同直接返回，否则的话用广度优先搜索算法，每计算一层，distance++,看看是在第几层找到的，返回distance

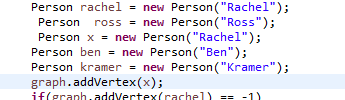
### 设计/实现Person类

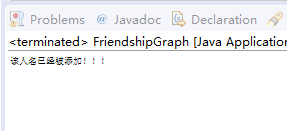
Person中包括name，visited，friends三个参数，并且撰写set,get方法，其中name是人名，visited表示这个人是否被访问过，friends表示这个人的朋友，用list存储。

### 设计/实现客户端代码main()

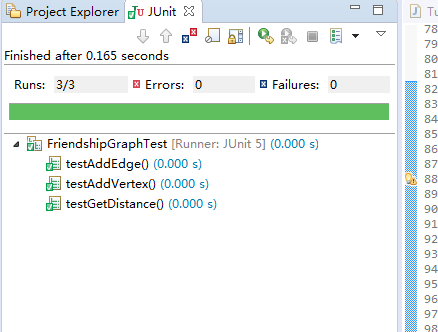
main函数就是自己写一些用例进行计算输出，看看结果对不对。

### 设计/实现测试用例





从上面可以看出，当执行graph.addVertex(x);的代码时，因为Rachel已经添加过了，所以会输出错误提示，程序结束。下面是其他的测试用例测试结果



## Tweet Tweet

任务四主要让我们了解Tweet这一个类，并通过它解决以下的一些问题：

1. 根据tweets寻找出最早的时间和最晚的时间
2. 根据tweets中的text信息，找到符合规则的@后的人名
3. 根据给的tweets以及name,返回该name所写的tweets的全部信息
4. 根据给的tweets以及时间段，返回该时间段的tweets的信息
5. 根据给的tweets以及字符串，返回包含该字符串的tweets的信息
6. 根据给的tweets的信息，建立社交map

### Problem 1: Extracting data from tweets

1. getTimespan:在该函数中找到最早的时间点和最晚的时间点，返回由这两个时间点构成的Timespan
2. getMentionedUsers:在完成该函数之前，先写一个函数isNumeric，确定给定的字符串是否符合实验中所要求的构成，是的话返回true，不是的话，返回false。然后将tweets中的text内容按照空格分隔开，对每个部分寻找是否存在@符号，存在的话将@后面的字符调用上面的isNumeric函数，如果返回值是true,添加该字符串到set中的，完成后返回。

### Problem 2: Filtering lists of tweets

1. writtenBy：遍历tweets，找到和所给name一样的author,将这条tweets添加到list中。
2. inTimespan：遍历tweets，将所给的最早时间和最迟时间和tweets的时间比较，如果在该时间段里面，将该tweets添加到list中，然后返回list
3. containing:遍历tweets，将text的内容用空格分开，判断每个部分和所给words是否相同，如果相同，添加该条tweets，最后返回list。

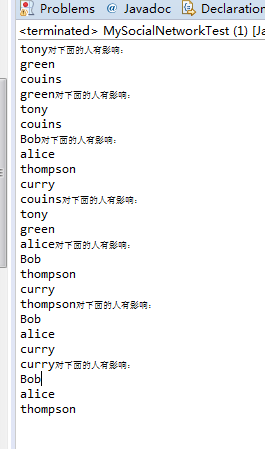
### Problem 3: Inferring a social network

1. guessFollowsGraph：该函数需要声明一个map，遍历tweets，在循环中，判断该tweets的text的@之后是否是一个人名，是的话保存当前的字符串，添加到set中，循环一次之后map进行一次put操作。最后返回该map。
2. influencers：根据所给的map新建一个map, Map<String,Integer>，integer中保存该人的map对应的values的size,然后对integer进行排序操作，排序完成之后利用ArrayList<String>的进出顺序不变的原理，依次放入。

### Problem 4: Get smarter

给函数guessFollowsGraph1参数Map<String, Set<String>>，前面的String是发tweets或者转发tweets的人，后面的是该用户的转发该tweets的列表信息，返回值是Map<String, Set<String>>，前面的String是发推或者转发tweets的人，后面是因为前面该用户而受到影响的用户的人名。

guessFollowsGraph1该函数需要先把发tweets或者转发的人的列表添加进去，然后将该列表再遍历重复一遍上面的操作，最后将信息添加到map中，重复遍历上面的操作，就可以得到结果。



# 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 |
| 2019-03-02 | 08:00-11:30 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按时完成 |
| 2019-03-02 | 14:00-17:30 | 完成问题1的要求二 | 提前一小时完成 |
| 2019-03-02 | 19:00-21:00 | 完成问题二的前面一部分，能做多少算多少 | 按时完成 |
| 2019-03-03 | 08:00-11:30 | 完成问题二 | 未完成，拖到下午 |
| 2019-03-03 | 13:30-16:30 | 完成问题二 | 凸包卡住未完成，留到以后 |
| 2019-03-03 | 18:00-21:00 | 对问题三有了解，简单做一点 | 按时完成 |
| 2019-03-04 | 13:45-22:00 | 完成问题三 | 按时完成 |
| 2019-03-05 | 18:30-23:30 | 做问题四前两部分 | 按时完成 |
| 2019-03-06 | 18:00-22:30 | 写做完的实验报告 | 未按时完成 |
| 2019-03-09 | 8:00-20:00 | 解决剩下的所有问题，完成报告 | 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 |
| 在问题一的要求二的报错分析中有一个是数组越界的问题，当时把我认为的方法：if中的row++之后写判断代码if(row==n)，报错返回，结果不管输入什么n，奇数偶数都报错了 | 通过使用一个3阶矩阵的实例进行分析之后，发现因为最大的数一定在矩阵的最后一行，满足所有条件，一定会出现row=n的，之所以之前的没有报错是因为代码执行到row=n时，正好最大的值也被赋值了，不会再次进入循环了，也就没有数组越界的情况发生了，因此我之后只能在赋值语句的前面加上判断是否row=n的条件来判断数组是否已出！ |
| 问题二的git指令暂时不太会用 | 个人计划周六日的整个时间先写代码，等周一之后有较短时间简单学习git |
| 在做问题二的问题六时，遇到问题就是我认为我写的思路没与任何问题，结果算的也对，但是就是过不了测试 | 向同学求助，在与他们的讨论过程中发现自己把0度的方向看做是x轴的正方向了，所以计算错误，回过头来看，是自己的审题不认真导致的，在题目中的例子没有好好看，才导致自己一直卡在这个地方，浪费了很长的时间。 |
| 在做凸包问题的时候，junit的测试一直过不了，一开始我以为计算的结果有问题但是从右图可以看到，除了顺序，跑出来的结果和测试用例一样 | 经过与同学讨论发现测试代码中比较的点的地址，而我在函数过程中一开始的起始点我new了一个点，把new的点添加进去了，因此地址不一样，从而没通过。 |

# 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

进过本次的实验，让我对于java的编程有了基本的回顾，不了解的地方也通过查相关的资料弄明白了，受益匪浅。在实验过程中有许多我自己未料到的错误卡了我很长时间，比如说那个point比较的是地址等等，但是这些问题的出现也让我更加深入的了解了本次的实验！增长了自己的知识！

## 针对以下方面的感受

1. Java编程语言是否对你的口味？
2. 关于Eclipse IDE
3. 关于Git和GitHub
4. 关于CMU和MIT的作业
5. 关于本实验的工作量、难度、deadline
6. 关于初接触“软件构造”课程

个人感觉java的编程还是很有意思的，eclipse我之前就一直在用，很习惯这个ide，git我之前没有接触，了解他我花费了一些时间，CMU和MIT的题目让我深有感触，感觉这些外国的题目虽然因为我个人英语水平有限，了解题目意思有些困难，但是这些题目在锻炼我们编程能力的同时还与数学等知识相结合，也有写自己创意的题目，让我做起来趣味横生！回味无穷!说实话这个课程难度还是挺大的，需要课下花费大量的时间，但是因为deadline比较晚，时间还算充裕。软件构造这门课初次接触，知识也没什么了解，暂时没什么想说的。