

**2021年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 沈城有 |
| 学号 | 1190200526 |
| 班号 |  |
| 电子邮件 |  |
| 手机号码 |  |

**目录**

[1 实验目标概述 1](#_Toc72568393)

[2 实验环境配置 1](#_Toc72568394)

[3 实验过程 2](#_Toc72568395)

[3.1 Magic Squares 2](#_Toc72568396)

[3.1.1 isLegalMagicSquare() 2](#_Toc72568397)

[3.1.2 generateMagicSquare() 3](#_Toc72568398)

[3.2 Turtle Graphics 5](#_Toc72568399)

[3.2.1 Problem 1: Clone and import 5](#_Toc72568400)

[3.2.2 Problem 3: Turtle graphics and drawSquare 6](#_Toc72568401)

[3.2.3 Problem 5: Drawing polygons 7](#_Toc72568402)

[3.2.4 Problem 6: Calculating Bearings 8](#_Toc72568403)

[3.2.5 Problem 7: Convex Hulls 9](#_Toc72568404)

[3.2.6 Problem 8: Personal art 9](#_Toc72568405)

[3.2.7 Submitting 10](#_Toc72568406)

[3.3 Social Network 10](#_Toc72568407)

[3.3.1 设计/实现FriendshipGraph类 11](#_Toc72568408)

[3.3.2 设计/实现Person类 12](#_Toc72568409)

[3.3.3 设计/实现客户端代码main() 12](#_Toc72568410)

[3.3.4 设计/实现测试用例 13](#_Toc72568411)

[4 实验进度记录 13](#_Toc72568412)

[5 实验过程中遇到的困难与解决途径 14](#_Toc72568413)

[6 实验过程中收获的经验、教训、感想 14](#_Toc72568414)

[6.1 实验过程中收获的经验和教训 14](#_Toc72568415)

[6.2 针对以下方面的感受 14](#_Toc72568416)

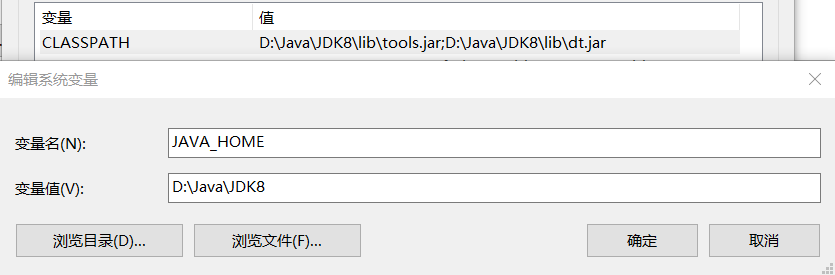
# 实验目标概述

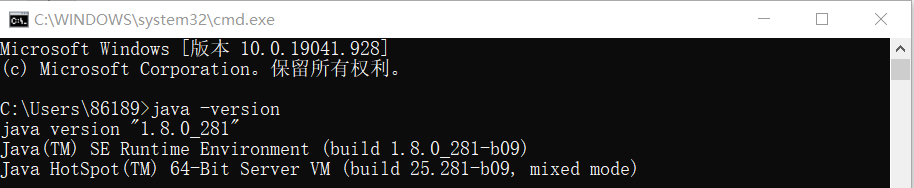
本次实验通过求解三个问题，训练基本Java编程技能，能够利用Java OO开发基本的功能模块，能够阅读理解已有代码框架并根据功能需求补全代码，能够为所开发的代码编写基本的测试程序并完成测试，初步保证所开发代码的正确性。另一方面，利用Git作为代码配置管理的工具,学会Git的基本使用方法。

# 实验环境配置

寒假自学期间按照我校计算学部团委QQ公众号发布的《Java设计预习第一弹：教你如何下载配置Java编译环境》进行了Java环境及Eclipse的安装及基本配置；从Git官网上下载了Git并简单学习了其使用方式；GitHub使用之前注册的账号。

在配置Java环境结束后检查是否成功时，发现了环境变量错误问题，此问题导致在cmd中输入java -version报错。经检查为JAVA\_HOME路径错误，修改后解决。下图为修改成功后的环境变量及cmd运行截图：





GitHub Lab1仓库的URL地址：

<https://github.com/ComputerScienceHIT/HIT-Lab1-1190200526>

# 实验过程

## Magic Squares

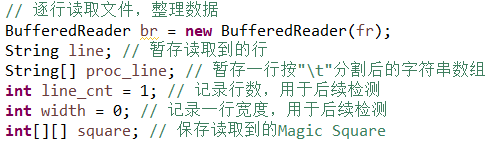
该任务要求我们首先了解什么是Magic Square（幻方），即按正方形排列的一组数，其满足每行、列、对角线上的数的和均相等。任务主要分为两部分，其一是编写isLegalMagicSquare()函数检查输入是否为Magic Square，二是理解并测试 generateMagicSquare()函数。

### isLegalMagicSquare()

首先使用传入函数的文件路径创建一个FileReader对象，利用此FileReader对象创建BufferedReader对象用于逐行读取文件内容，如下图：



随后利用循环逐行读取文件中数据至二维数组，过程与实验手册中提示的过程类似，此处不再展示源代码。下图为读取过程中所声明的变量及类：



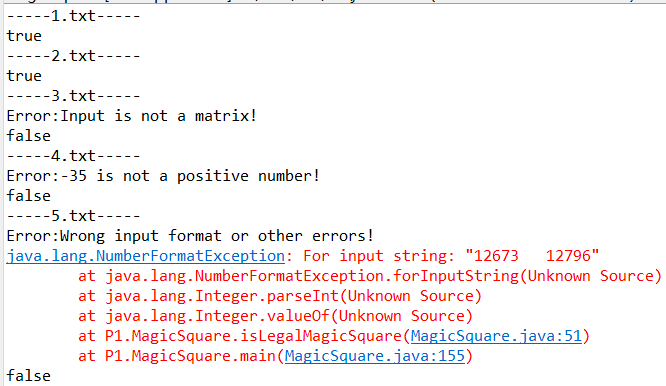
在读取过程中及读取完成后要检查并处理一些特殊情况：

1. 不符合定义情况1（行数大于列数）；
2. 不符合定义情况2（存在非正整数->输出Error提示，或格式不正确输入->exception）；
3. 不符合定义情况3（并非矩阵）；
4. 不符合定义情况4（行数小于列数）；
5. 空文件情况。

以上情况中，除b)以外，均通过编程检测并输出对应提示信息，b)中除检测到负整数外均利用异常输出相关的错误提示信息。遇到以上任何一种情况，程序均会在输出错误提示信息后返回false。

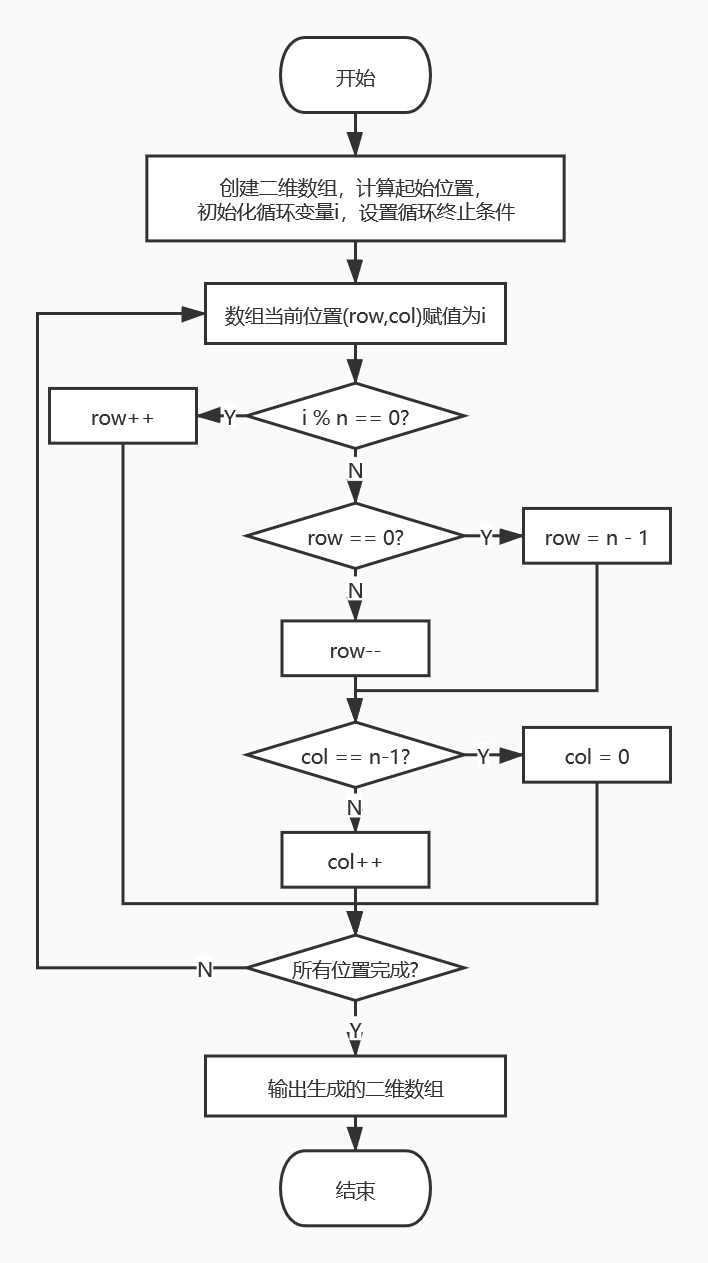
最后，如果数据通过了以上检查，则检查各行、列、对角线和是否相等。我的思路是：首先单独计算一行的和，存入变量sum，随后利用for循环逐行、列、对角线计算和并与sum比较，如果出现不相等则立即停止循环返回false，否则返回true。

按照以上设计思路及过程编写的代码运行结果如下图：



### generateMagicSquare()

下图为程序流程图：



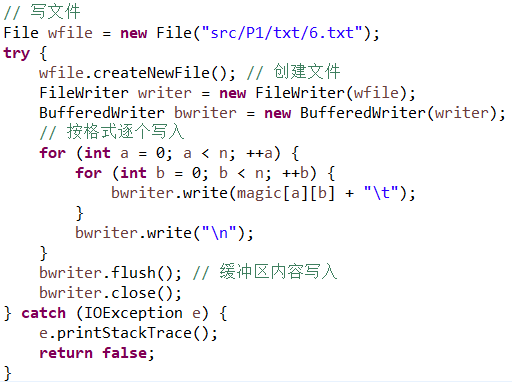
函数使用奇数参数n，首先计算初始位置（0，n/2），赋值为1，之后每次取当前位置的右上的位置，同时考虑超出范围的处理：如果当前行是第一行，则下一行为最后一行，如果当前列是最右边的列，则下一次取左边第一列（超类似于循环的思想）设置的值每次加1，如此重复n\*n次，就完成了对整个矩阵的赋值，而且达到了每行、列及对角线之和均相同的效果。

如果输入的参数n为偶数，变量row的值会出现等于n的情况，造成数组越界访问，从而抛出ArrayIndexOutOfBoundsException异常，程序终止。

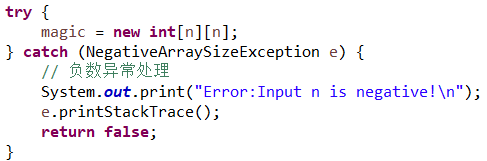
如果输入的参数n为负数，数组索引值会为负值，同样也是对数组的非法访问，从而抛出NegativeArraySizeException异常，程序终止。

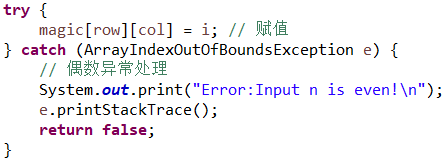
此函数中文注释见源代码文件，此处不再展示。

函数扩展1——写文件部分如下图（按格式逐个写入）：



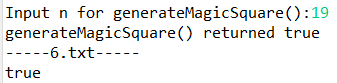
函数扩展2——输入n不合法异常处理（主要分为n为负数、偶数等情况，我选择设置try-catch来处理，也可以直接检测退出避免异常）：

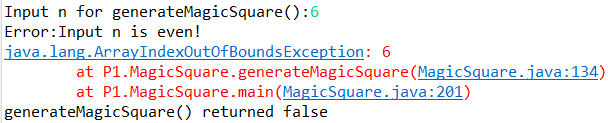


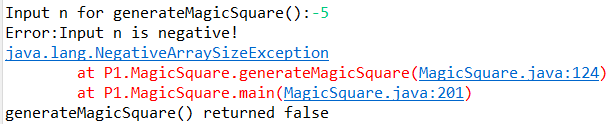


主函数测试部分见源代码文件。

运行结果：







## Turtle Graphics

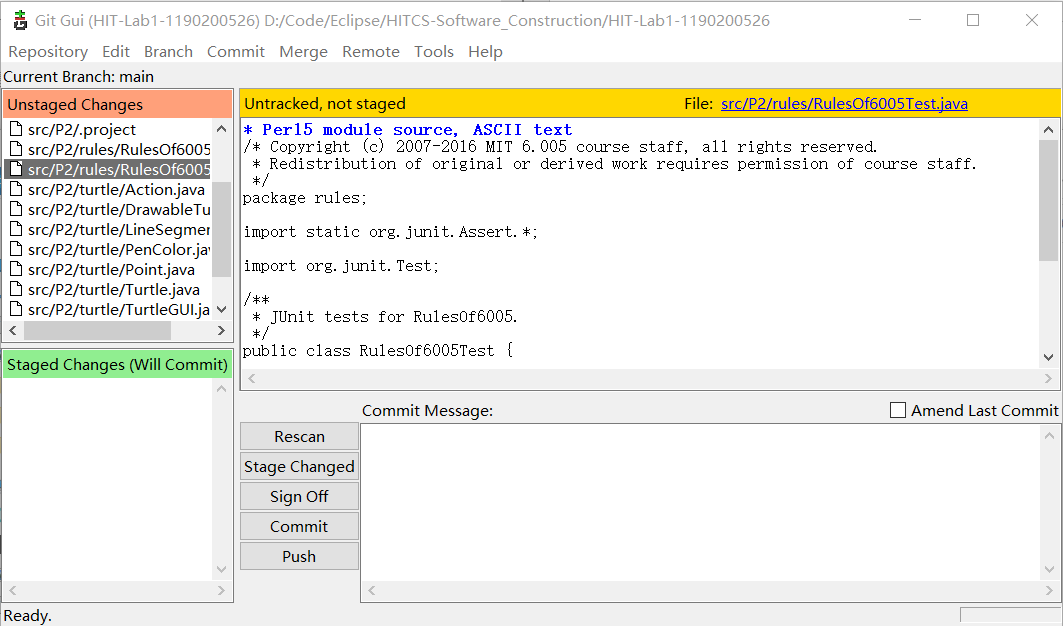
该任务的主要目标是练习使用Git的基本功能、学会在Eclipse中导入项目、熟悉Turtle Graphics中的一些函数接口及简单使用Junit进行程序测试。主要任务是实现简单的绘图及计算。

### Problem 1: Clone and import

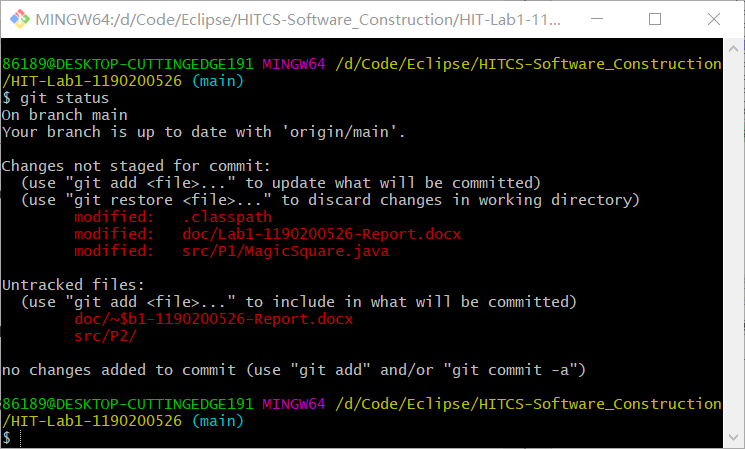
按实验手册要求，我从GitHub网页端下载了该任务代码的ZIP压缩包，并提取其中所需内容至之前已经创建好的本地Git仓库。

本地Git仓库通过git clone命令（SSH地址）将GitHub上的Lab1库克隆至本地指定文件夹，克隆的过程也确定了本地库与远程库的关联关系。

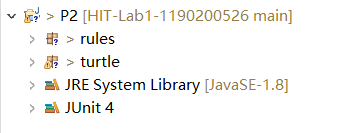
打开本地库文件夹，右键菜单选择Git Gui Here，可以可视化地查看库的状态并执行操作。界面如下图：



右键菜单中也有Git Bash Here的选项，使用此工具我们可以直接通过键入命令执行对库的操作，界面如下图：



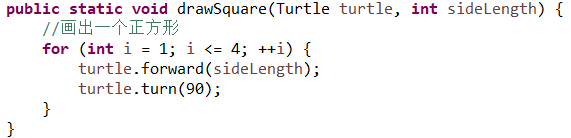
在开始P2的编程部分前，在Eclipse中按MIT实验指导的步骤完成了项目的导入，结果如下图：



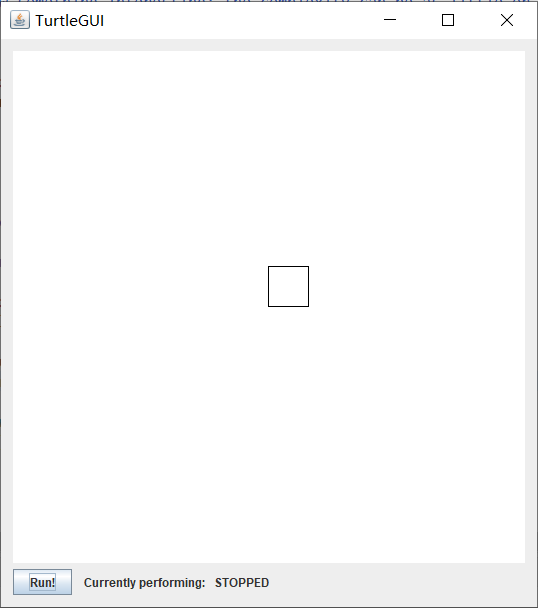
### Problem 3: Turtle graphics and drawSquare

要求：查看TurtleSoup.java，使用方法forward和turn实现drawSquare()函数来画出一个正方形。

执行四次forward和turn即可。代码如下：



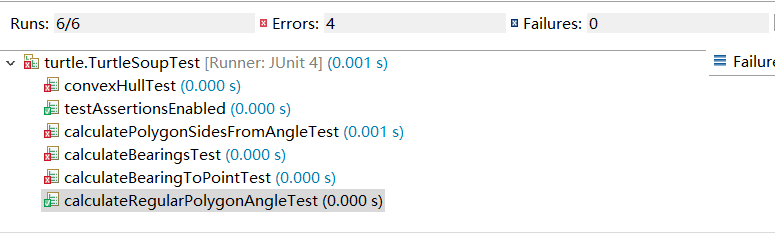
右键选择Run as -- 1 Java Application得到下图结果：



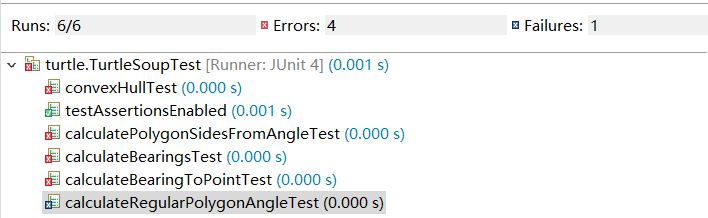
### Problem 5: Drawing polygons

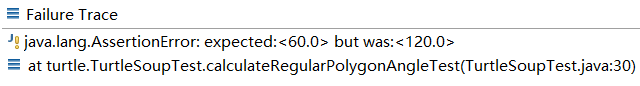
首先实现函数calculateRegularPolygonAngle()，即计算正n边形一个内角的大小，可使用(sides – 2) \* 180.0 / sides来计算。

随后使用Junit测试，结果如下图：



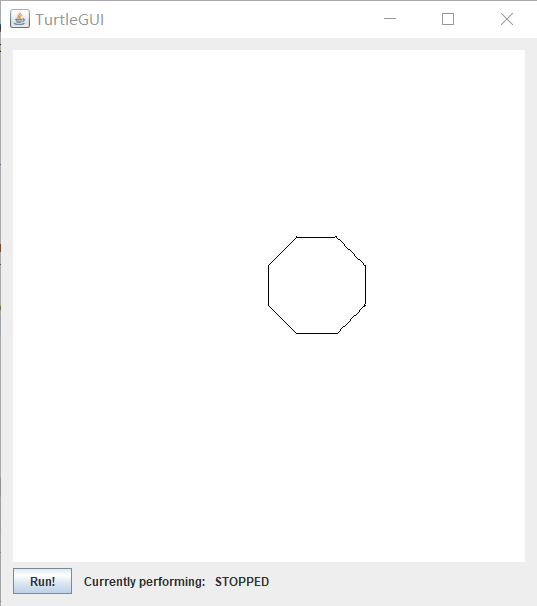
将实现的函数改为错误的，查看信息，了解JUnit：





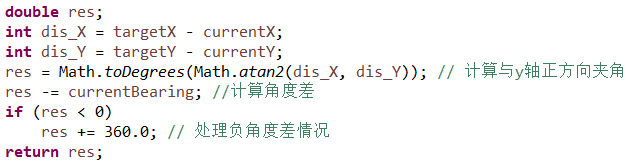
最后修复函数，实现drawRegularPolygon()。我的思路是利用循环执行forward和turn，循环边数次，每次转的角度为180.0 calculateRegularPolygonAngle()的计算值。代码此处略。

结果如下图（正8边形，边长40）：



### Problem 6: Calculating Bearings

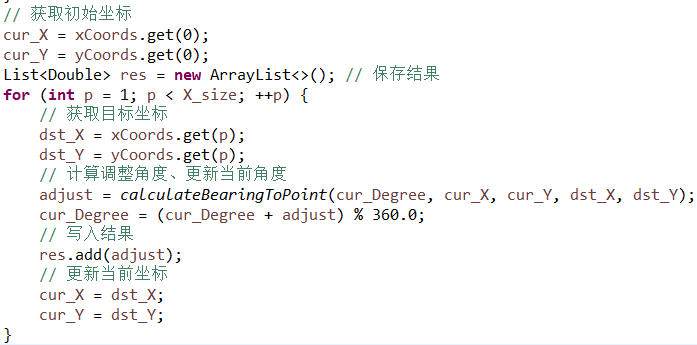
首先实现calculateBearingToPoint()函数。此函数根据当前位置、方向及目标位置计算前往目标位置需调整的角度。思路为使用arctan(x/y)的原理计算夹角，将夹角转换至所需范围，最后计算角度差，也要转换一次范围。代码及注释如下：



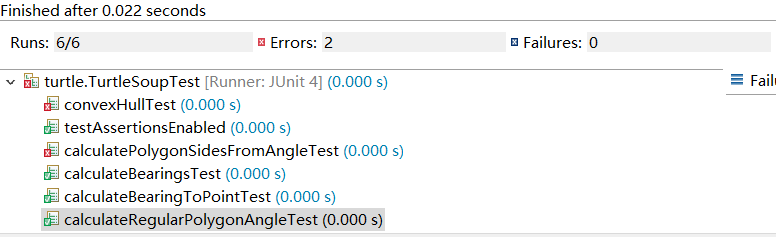
随后实现calculateBearings()函数。此函数调用calculateBearingToPoint()函数计算List中每相邻两点间要调整的角度，并将结果写入一个List<Double>对象输出。

主要实现思路为首先获取初始坐标、目标坐标及默认方向，通过函数calculateBearingToPoint()计算调整角度写入结果List，并更新当前坐标、目标坐标及当前方向，如此重复直至遍历完所有坐标。

主体部分代码如下图：



最终测试结果如下图：



### Problem 7: Convex Hulls

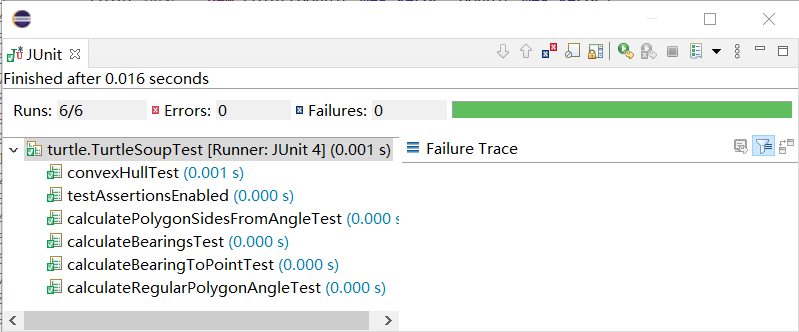
要求我们实现convexHull()函数，此函数计算并输出构成凸包周长顶点的输入点的最小子集。由于对此问题的定义及解决算法完全不熟悉，我首先在网上查找了相关资料。

此问题最常用的算法为Graham扫描法和Gift wrapping算法，此处我采用较容易实现的Gift-Wrapping算法。具体实现步骤如下：

1. 首先在所有点中选取y坐标最小的一点当作基点。如果存在多个点的y坐标都为最小值，则选取x坐标最小的一点；
2. 然后遍历点集，计算与基点连接后相对于y轴正方向偏转最小的点P，偏转均最小时选距离最大的，则P对应线段HP一定在凸包上，即点H、P在结果集合中；
3. 将点P设为当前点，遍历点集，按照b)重复执行；
4. 最终再次走到基点时结束。

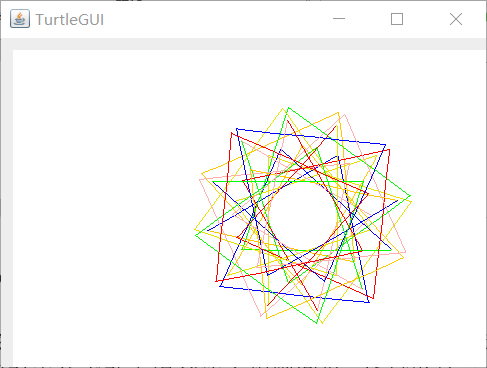
计算偏转角可使用重载的calculateBearingToPoint方法。具体代码实现较长，见源代码文件。

Junit测试结果如下图：



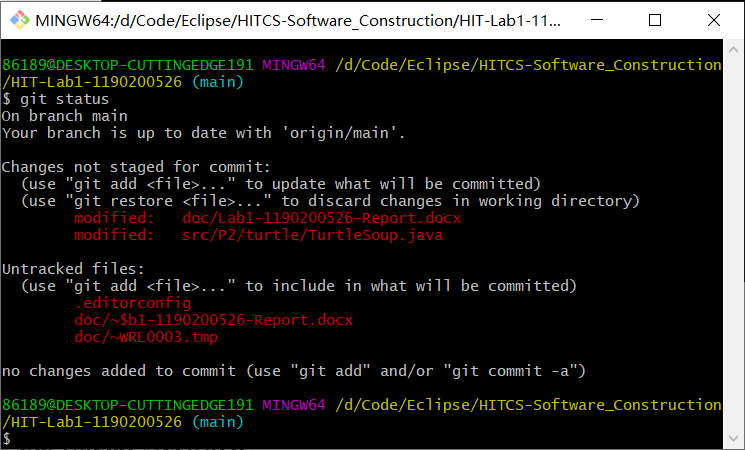
### Problem 8: Personal art

要求使用Turtle Graphics自行设计绘图。我的设计主要利用了Problem 5开始时由于忘记对内角求补而得到的不规则图形，通过循环画出一组这样的图形，并在作图过程中不断修改颜色可得到下图：



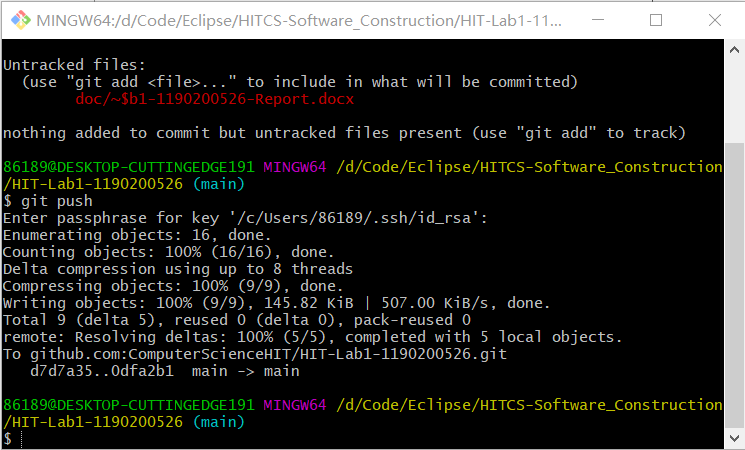
### Submitting

打开本地Git库所在文件夹，右键菜单选择Git Bash Here，进入Git Bash，使用git status命令，查看当前库状态，如下图：



使用git add \*命令，将所作所有更改暂存，之后使用git commit -m + 描述 将更改提交至本地库（图略）。

最后使用git push命令，将commit推送至GitHub。如下图：



## Social Network

该任务要求利用一些数据结构实现和测试FriendshipGraph类和Person类，这两个类将以无向图形式刻画简单的社交网络模型，同时也要求设计的类支持未来扩展到有向图。

### 设计/实现FriendshipGraph类

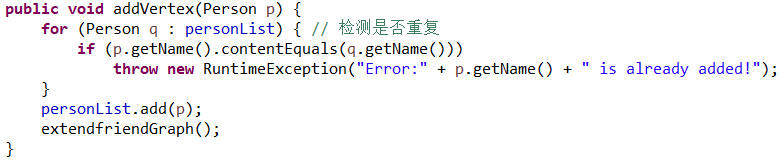
根据实验手册要求，我定义了以下两个private对象：



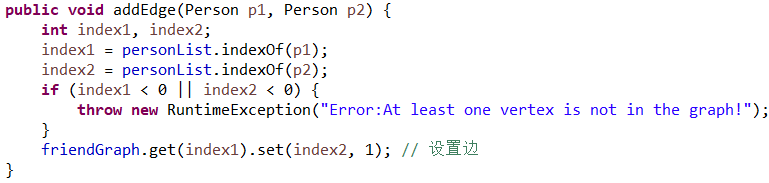
第一个用于保存关系网中的所有Person对象，第二个用于保存社交关系，原理类似于邻接矩阵。这些private对象将在构造时被初始化。

按照要求，公共方法主要有三个：addVertex，addEdge及getDistance。

对于addVertex方法，我的思路为：正常情况下，程序将新顶点加入personList，并调用私有方法扩展friendGraph（）；如果发生姓名重复，则抛出一个异常。具体实现如下：



对于addEdge方法，首先检测传入的参数是否都在personList中，如果是则设置friendGraph中对应位置为1，否则抛出一个异常。具体实现如下：



对于getDistance方法，开始时先判断两个参数是否相等，相等则返回0，若不相等，检测传入的参数是否都在personList中，如果不是，抛出异常；通过以上两个检测后，使用广度优先搜索的思想求出最短路径返回。

此方法核心在于BFS，此部分代码如下：



实现过程与C++类似。通过使用Java中与C++相似的Queue类及Map类中的HashMap，可以较为方便地实现BFS。

此类中私有方法extendfriendGraph较为简单，此处略去。

### 设计/实现Person类

Person类作为实现的社交网络中人的代表，只需要简单的属性、构造函数及公共方法用于设置、保存和获取人的姓名。具体实现如下：



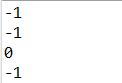
由于在实验手册客户端代码示例中没有有关删除、修改个人信息的操作，故Person类的实现较简单。

### 设计/实现客户端代码main()

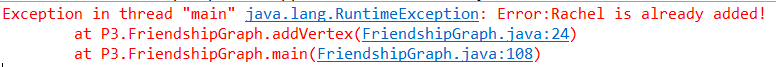
将手册中的代码复制后运行，程序输出与要求一致，如下图：



如果将graph.addEdge(rachel, ross);这条语句注释掉，则rachel和ross之间只存在单向的社交关系rossrachel，且rachel与ben之间变为不可达，故第14-17行应输出-1, -1, 0, -1。实际结果如下，与预期一致：

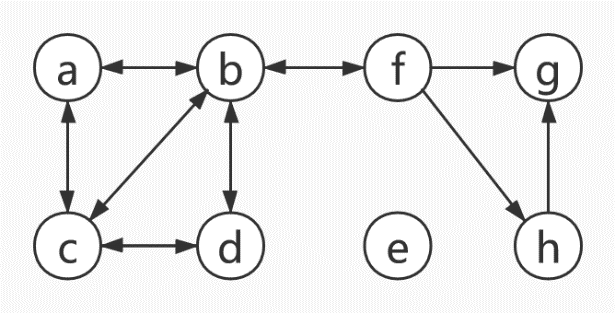


如果将第3行引号中的“Ross”替换为“Rachel”，则违背了每个人名字不同的前置条件，我的程序会抛出异常，效果如下：

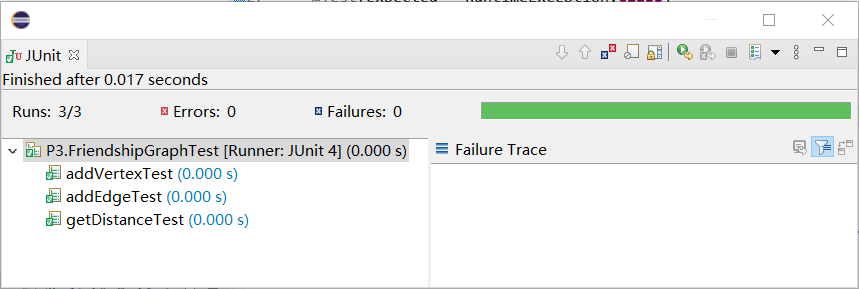


### 设计/实现测试用例

测试用例覆盖了各公共方法涉及到的非法输入处理（抛出异常），并创建了一个较实验手册稍复杂一些的社交网络图（见下图）来检测getDistance()方法在各种情况下的正确性。



具体测试代码此处略。测试结果如下图：



# 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 |
| 2021-05-13 | 14:00-15:30 | 学习Java相关语法知识，尝试编写问题1的isLegalMagicSquare()函数 | 基本完成，但不够完善 |
| 2021-05-13 | 18:30-21:00 | 完成isLegalMagicSquare()函数的编写和测试，撰写对应部分的报告 | 按计划完成 |
| 2021-05-15 | 16:00-18:30 | 完成P1及对应部分的报告 | 按计划完成 |
| 2021-05-16 | 15:00-17:30 | 完成P2 Problem1 - 4及对应部分的报告 | 按计划完成 |
| 2021-05-16 | 18:00-20:00 | 完成P2 Problem5、6及对应部分报告 | 按计划完成 |
| 2021-05-17 | 20:00-22:40 | 完成P2剩下的编程任务 | 基本完成 |
| 2021-05-18 | 16:00-17:00 | 完成P2对应报告内容，完善程序代码 | 按计划完成 |
| 2021-05-18 | 19:00-20:30 | 学习Java类的相关知识，初步尝试设计P3的两个类 | 按计划完成 |
| 2021-05-19 | 16:00-17:00 | 继续进行P3编程 | 部分方法需完善 |
| 2021-05-19 | 18:30-22:00 | 完成P3除测试用例以外部分的编程及对应报告内容 | 按计划完成 |
| 2021-05-20 | 14:00-15:30 | 完成P3测试用例的设计 | 按计划完成 |
| 2021-05-20 | 18:30-21:00 | 完成实验全部任务 | 按计划完成 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的困难 | 解决途径 |
| 不熟悉Java的输入输出流，特别是读取文件操作 | 阅读网络上相关文章：  <https://blog.csdn.net/caidewei121/article/details/89426032>  借阅书籍：《Java程序设计》、《Java核心技术卷Ⅱ》 |
| Eclipse项目导入与配置问题 | 通过尝试和求助其他同学最终成功导入工程，完成相关配置，也对Eclipse项目结构有了一定的了解。 |
| 不了解凸包问题及其算法 | 查看维基百科上的关于凸包的内容：  <https://zh.wikipedia.org/wiki/%E5%87%B8%E5%8C%85> |
| 不熟悉如何编写JUnit测试 | 参考P2的Junit测试代码--TurtleSoupTest.java |

# 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

1. 学会使用Eclipse进行Java简单编程；
2. 能够设计简单的测试用例并使用Junit进行程序测试；
3. 深化了课堂上所学的关于规约、可变性与不可变性等知识。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

我对Java编程很感兴趣。Java提供了大量功能强大的类（类似于C++的STL），在具备C\C++语言的基础后能较快熟悉Java的基本使用，应用也十分广泛。

1. 关于Eclipse IDE；

开始使用时遇到了一些困难，但经过不断尝试和探索，现在使用已经没有太多障碍。

1. 关于Git和GitHub；

学习此课程前我已经使用了一段时间，已经掌握了基本的使用方法。

1. 关于CMU和MIT的作业；

作业要求具体，步骤明确，完成后很有收获。

1. 关于本实验的工作量、难度、deadline；

此次实验工作量较大，难度也不小，但deadline较为宽松。

1. 关于初接触“软件构造”课程；

课程注重实践，与实际应用紧密结合，我也有一定的兴趣。