

**2024年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 奔佳达 |
| 学号 | L20L020801 |
| 班号 | 2003102 |
| 电子邮件 | [prakatpin@outlook.com](mailto:prakatpin@outlook.com) |
| 手机号码 | 15244755982 |

**目录**

[1 实验目标概述 1](#_Toc97648154)

[2 实验环境配置 1](#_Toc97648155)

[3 实验过程 1](#_Toc97648156)

[3.1 Magic Squares 1](#_Toc97648157)

[3.1.1 isLegalMagicSquare() 1](#_Toc97648158)

[3.1.2 generateMagicSquare() 1](#_Toc97648159)

[3.2 Turtle Graphics 1](#_Toc97648160)

[3.2.1 Problem 1: Clone and import 2](#_Toc97648161)

[3.2.2 Problem 3: Turtle graphics and drawSquare 2](#_Toc97648162)

[3.2.3 Problem 5: Drawing polygons 2](#_Toc97648163)

[3.2.4 Problem 6: Calculating Bearings 2](#_Toc97648164)

[3.2.5 Problem 7: Convex Hulls 2](#_Toc97648165)

[3.2.6 Problem 8: Personal art 2](#_Toc97648166)

[3.2.7 Submitting 2](#_Toc97648167)

[3.3 Social Network 2](#_Toc97648168)

[3.3.1 设计/实现FriendshipGraph类 2](#_Toc97648169)

[3.3.2 设计/实现Person类 2](#_Toc97648170)

[3.3.3 设计/实现客户端代码main() 2](#_Toc97648171)

[3.3.4 设计/实现测试用例 3](#_Toc97648172)

[4 实验进度记录 3](#_Toc97648173)

[5 实验过程中遇到的困难与解决途径 3](#_Toc97648174)

[6 实验过程中收获的经验、教训、感想 3](#_Toc97648175)

[6.1 实验过程中收获的经验和教训（必答） 3](#_Toc97648176)

[6.2 针对以下方面的感受（必答） 3](#_Toc97648177)

# 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。完成本次实验，学生将了解基本的Java OO编程、基于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

本人使用Windows 10 64位操作系统的电脑，利用了IntelliJ IDEA 2024.1 (Community Edition) 综合开发环境来编程和运行代码（链接：https://www.jetbrains.com/idea/download/?section=windows），从git官网下载git安装包（git-bash包含在内，链接：https://git-scm.com/download/win），Java开发工具包（JDK 22版本，链接：<https://www.oracle.com/sg/java/technologies/downloads> ）；至于junit4.13和hamcrest-core-1.3的“.jar”包文件可在IntelliJ里下载。若下载和配置的过程中遇到困难可以点击[此处](https://blog.csdn.net/z1667002/article/details/137499315)去详细阅读下载、安装、配置的教程。正确安装和配着后，IntelliJ会自动识别出JDK。本人GitHub Lab 1仓库的地址：  
https://github.com/justgithubpk/HIT-Lab1-L20L020801.git

# 实验过程

## Magic Squares

实验要求使用Java语言编写一段程序（命名为MagicSquare.java） 能够对矩阵检查其：行、列、对角的个数和值，并进行判断该矩阵是否为Magic Square。

第一任务，程序能够读取给定的5个测试txt文件，并判断是否为Magic Square成立。若返回为真，本人添加了一个“Successfully read <filename>”提示。第二任务，为扩展任务，对generateMagicSquare函数进行扩展，能够产生一个Magic Square后保存到一个txt文件（6.txt）中。而且，对6.txt进行判断，该文件存放的值是否为Magic Square。如果遇到偶数阶或者负数，程序的返回为假，让用户重新输入阶数。

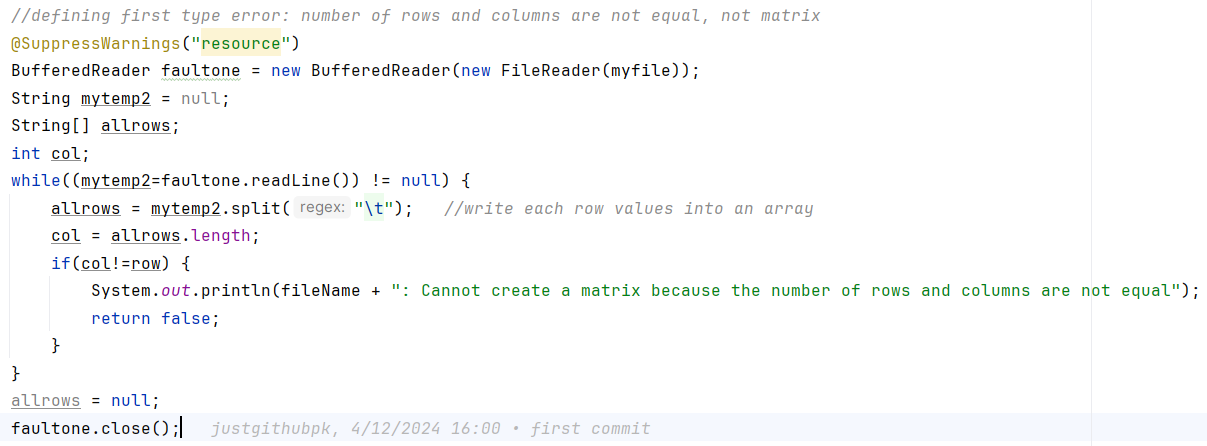
### isLegalMagicSquare()

**设计思路：**根据实验手册的要求，函数isLegalMagicSquare的主功能为判断输入的矩阵是否满足成为合格Magic Square的条件，读取txt文件里的值去存入数组中，读取成功后将从三个方面进行判断；第一，该矩阵是否为方针，即行、列、对角的个数相等；第二，矩阵值是否为正整数；第三，值和值之间的分隔符是否为“\t”。如果以上三个条件其中一条不被满足，程序将输出相应的提示并返回false；否则，开始对矩阵的每一行、每一列、对角求和，判断是否相等。如果相等了，程序将返回true，否则返回false。

**过程描述：**

1. 判断矩阵是否为方针，先调用do-while循环统计行的个数，再以“\t”为分隔符去读每一列的值统计列的个数；若行与列的个数相等，则返回true，否则返回false。
2. 抓非正整数的矩阵值，首先将以“\t”各分的字符串存入数组，再判断是否存在非法的值；若有，则返回false，否则返回true。
3. 检查分隔符，若存在两个数值不以“\t”去分隔，运行Interger.valueOfs的时候将抛出异常信息，并返回false。

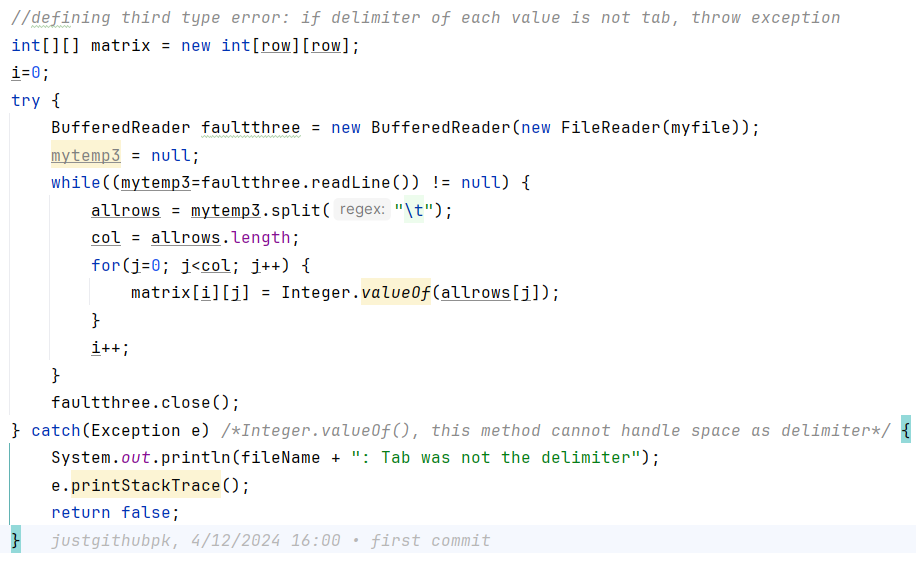
***三种错误代码***



第一种错误

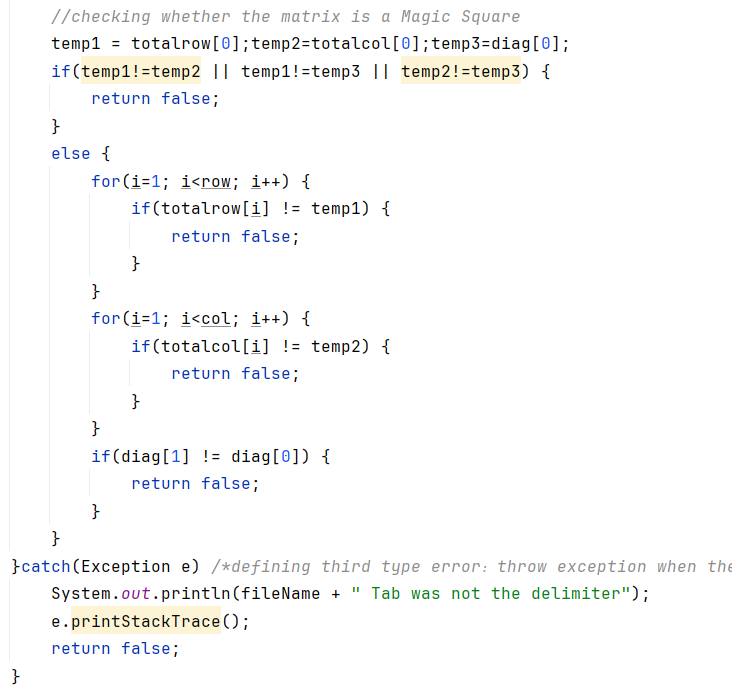


第二种错误



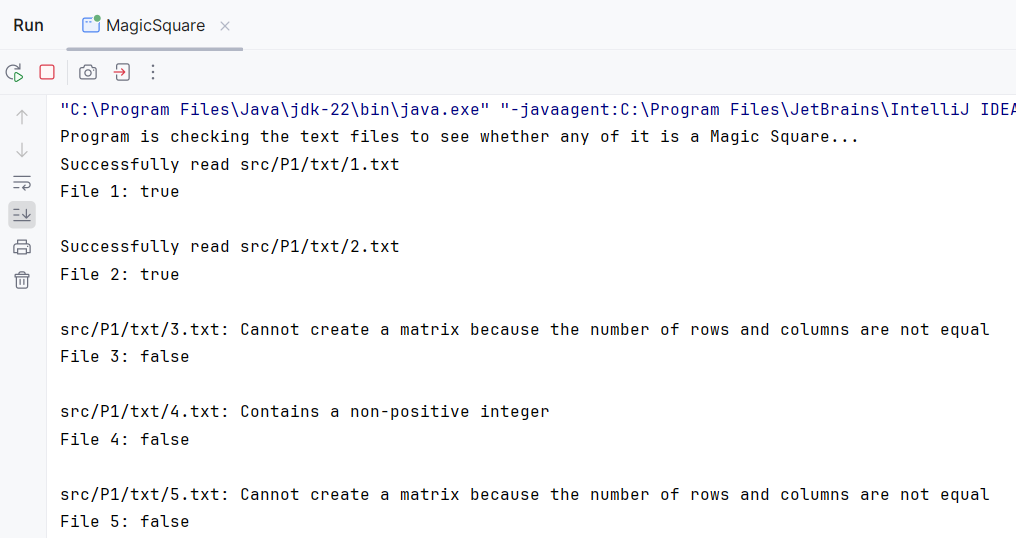
第三种错误

***Magic Square的判断***



判断矩阵是否为 Magic Square

**运行结果：**如下方图所示



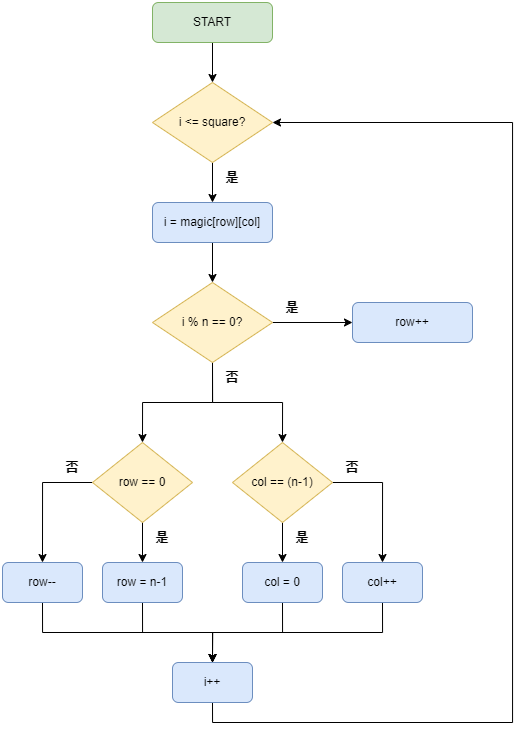
### generateMagicSquare()

**设计思路：**实验手册已列出实现generateMagicSquare函数的核心代码，为了能够让程序满足实验要求，本人给添加了如下功能：

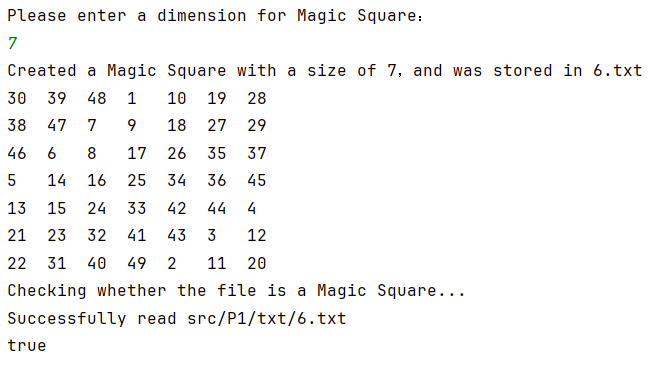
1. 写出功能，即将计算机生成的Magic Square结果写到txt文件里（6.txt）；
2. 检查结束，当用户输入的阶数n为非法阶数（如n为偶数，n为负数）时，程序将跳出提示并退出函数让用户重新输入合格的阶数；

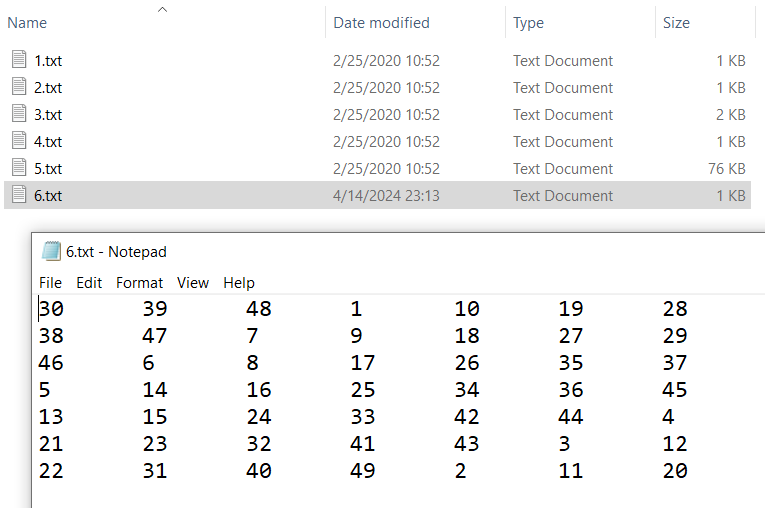
于是，我们需要判断阶数n是否满足Magic Square的矩阵阶数；若满足，开始生成Magic Square并写到对应的txt文件里。

**过程描述：**



运行结果：





## Turtle Graphics

海龟图形是1967年发展的标志性几何绘图工具的实现，该工具最初在Logo中引入。海龟可以使用重复简单移动的程序绘制复杂的形状。海龟图形提供了一个物理“海龟”的表示（一个带有笔的小型机器人），它在地板上的纸张上绘制。这是学习者接触编程概念并与软件交互的有效和经过验证的方法，因为它提供即时可见的反馈。它还提供了方便的图形输出访问。海龟绘图最初是作为教育工具创建的，供教师在课堂上使用。对于需要产生一些图形输出的程序员来说，这可以是一种无需引入更复杂或外部库的开销来完成的方法。

### Problem 1: Clone and import

**从GitHub获取该任务的代码：**

1. 绑定GitHub到本地开发工具，打开Git Bash

1、获取密钥：ssh-keygen -t rsa -C "your\_email@example.com"

2、往GitHub网站，复制公钥添加到GitHub上

3、回到Git Bash界面，输入 ssh -T git@github.com

4、git config –global user.name “your GitHub username”

5、git config –global user.email “your\_email@example.com”

**使用git在本地创建git仓库和管理本地开发:**

\*先使用cd指令进入到本地项目路径

1、git init

2、git add .

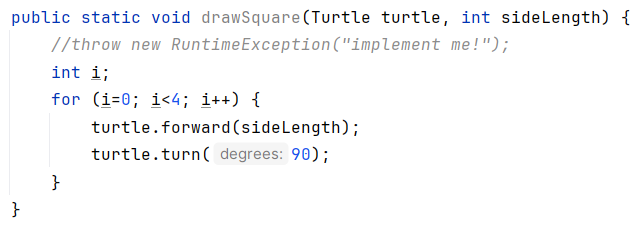
3、git commit -m “first commit”

4、git remote add origin *link to your GitHub repository*

5、git push origin master

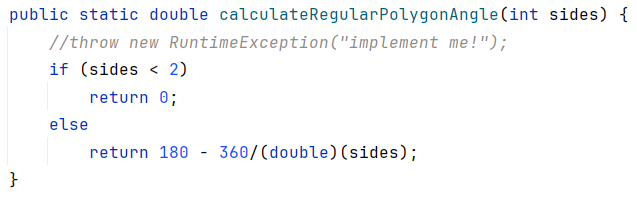
### Problem 3: Turtle graphics and drawSquare

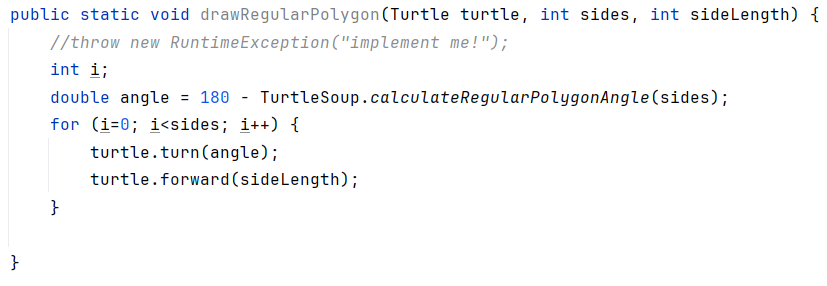
乌龟向前走sideLength步，然后转90度，重复4次得到一个正方形



### Problem 5: Drawing polygons

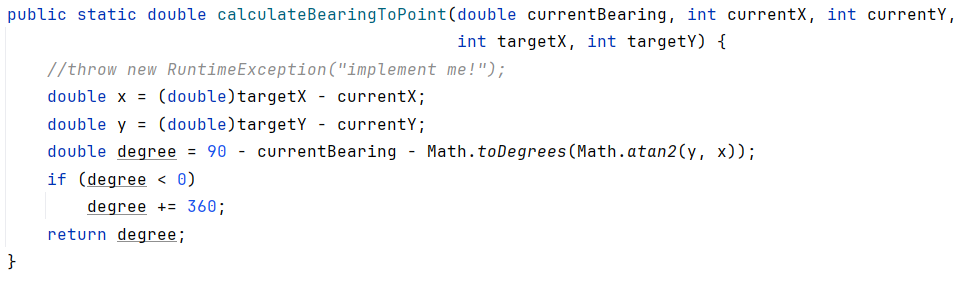
计算正多边形内角只需根据正多边形 内角=(边数-2)/边长  
写出calculateRegularPolygonAngle (int sides) 然后画正多边形直接调用此函数

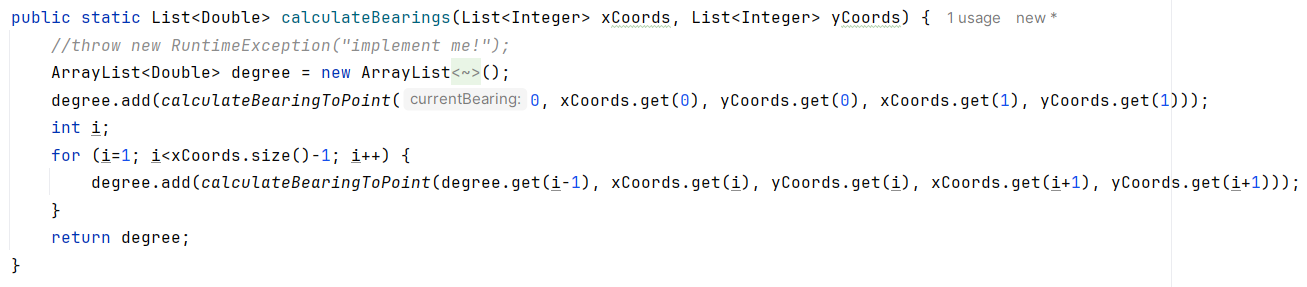




### Problem 6: Calculating Bearings

调用atan2函数得到弧度计算前进方向的偏转角，然后调用toDegree转换成角度；如角度为负值，则需加上360度。

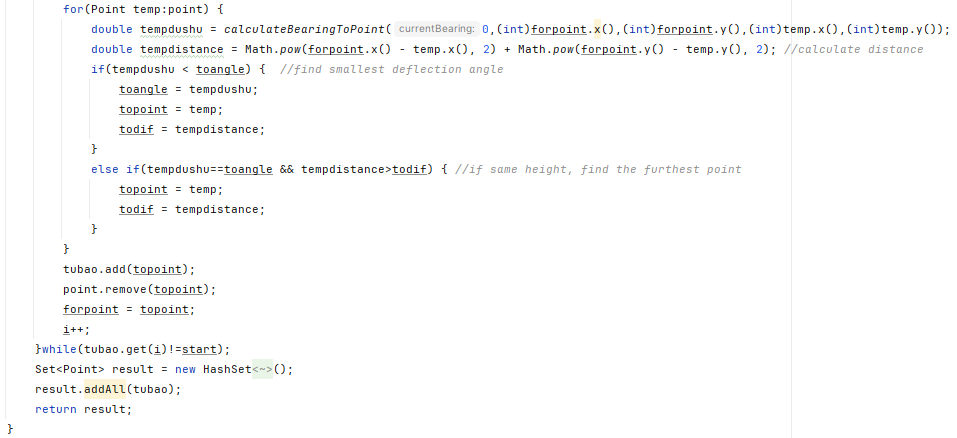




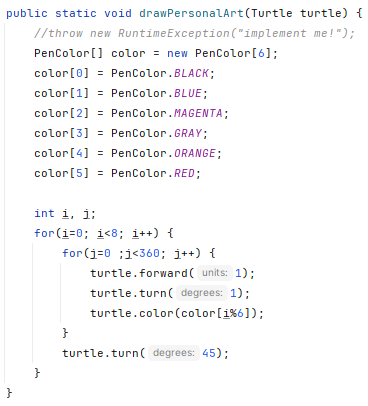
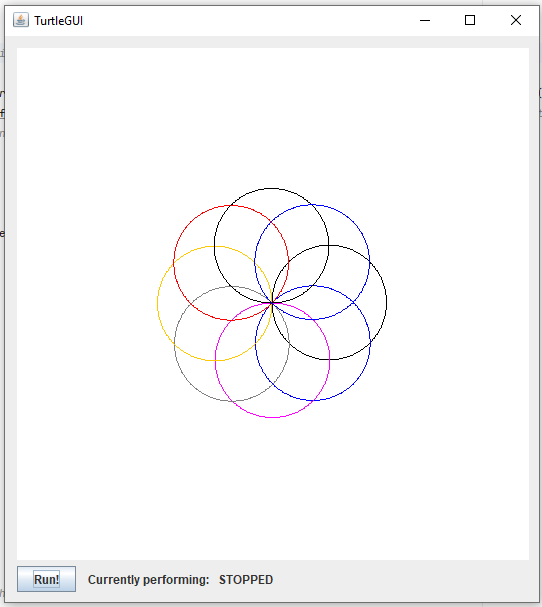
### Problem 7: Convex Hulls

利用Gift Wrapping Algorithm (GWA) 算法，计算给定集合中的凸包。首先找到最左下角的点加入集合，再比较剩余点到此点的偏转角，然后找到偏转角最小的加入集合；当偏转角相同时，需找到最长的一条边的点加入集合，最后得到凸包的点集。



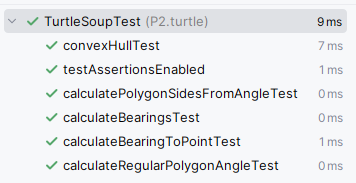


### Problem 8: Personal art



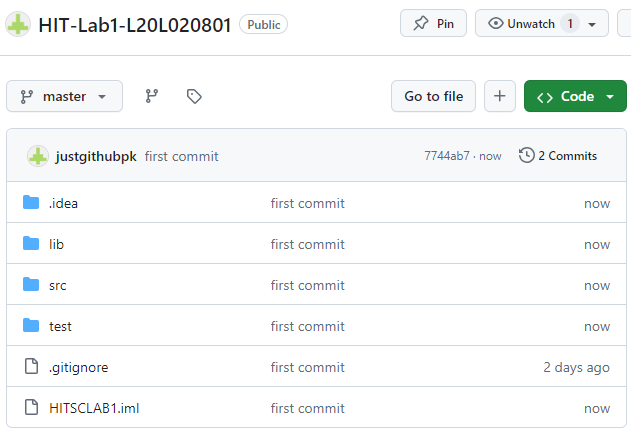
### Submitting

Junit测试结果如下：



通过Git提交到GitHub：

1. 打开Git Bash;
2. cd /IdeaProjects/HITSCLAB1
3. git add .
4. git commit -m “first commit”
5. git push origin master



## Social Network

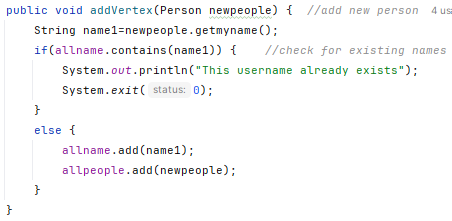
第三部分实验要求构造Person类和Friendship Graph类，调用FriendshipGraph构建Person之间的关系来模拟社交网络上的关联，可以计算出两个Person之间最短路径。

### 设计/实现FriendshipGraph类

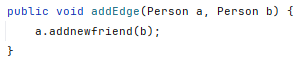
设计思路：对于FriendshipGraph类中，我们要实现，添加一个Person（addVertex）,增加一条边（addEdge）,就可以计算两个人之间的最短距离了。主要挑战为第三功能的实现相当复杂，决定采用广先遍历找出最短路径。



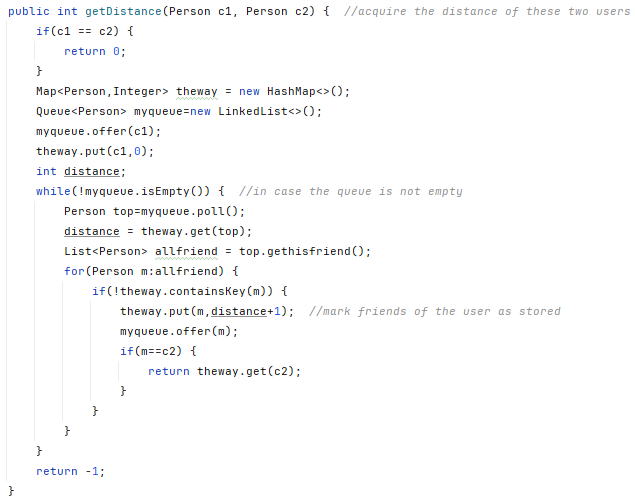
功能1：addVertex()，首先判断增加的Person对象是否已经存在，需要在开始判断allname集合中是否已经有了待增加的Person对象的名字；如果已存在，则输出“This username already exists”并结束程序，若没有，则将待增加的新的Person对象加入allpeople集合中，并且将对象的名字加入allname集合中。



功能2：addEdge()，直接调用Person类的addnewfriend方法



功能3：getDistance()，首先需要知道如果两个Person对象为同一个，则返回0。再定义一个Map集合theway和一个Person队列myqueue，队列myqueue用来储存广搜的遍历结果，theway的Map集合用来储存广搜的所有元素及他们与第一个元素的距离。具体实现方法为首先将第一个元素c1入队，并且把第一个元素c1和下标0入集合，当队列非空时，弹出队首元素top并将top在集合theway中的下标距离，然后调用gethisfriend函数获得队首元素的所有朋友allfriend，只要allfriend中的元素与c2不同就把这些元素全部入队。将这些元素及下标distance+1放入集合。如果队列非空，继续执行以上步骤，直到找到某个元素与c2相同并且返回这个元素在集合theway中的下标。如果直到队列为空还没找到c2，则返回-1。

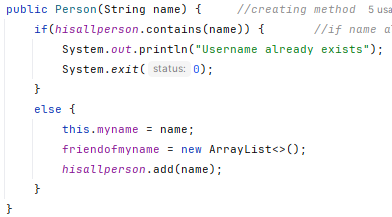


### 设计/实现Person类

在Person类中我们要完成的功能为：增加本人的新朋友addnewfriend，得到本人的名字getmyname，得到本人的朋友列表getthisfriend。定义一个String变量myname，储存名字，定义一个Person的集合friendofmyname，定义一个私有静态String集合hisallperson（用来储存所有已经存在的名字，防止定义相同名字的Person对象）。



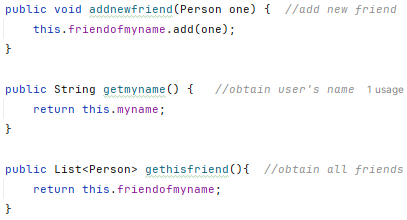
构造方法：若名字已经在hisallperson中已经存在则输出“此名字已存在”并结束程序，否则初始化这个名字，并且将这个名字加入hisallperson中。



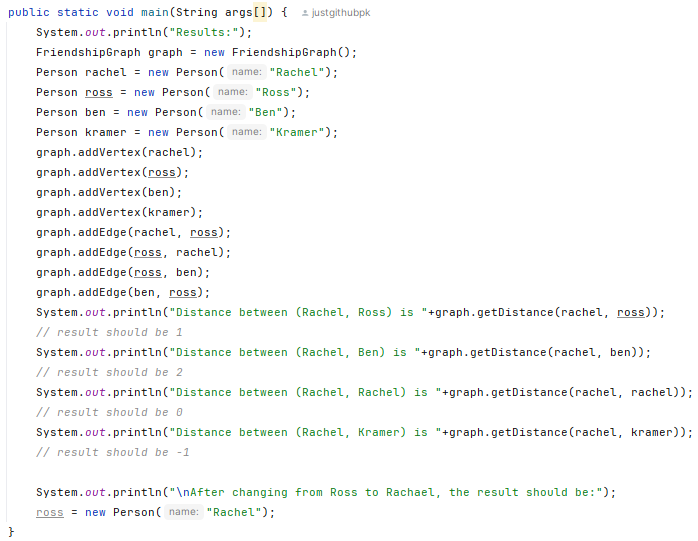
增加本人新朋友addnewfriend

得到本人的名字getmyname

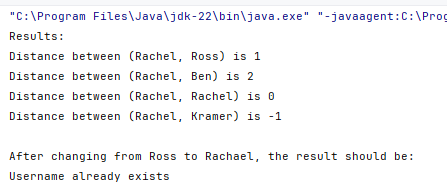
得到本人的朋友列表getthisfriend



### 设计/实现客户端代码main()



运行主函数得到的结果



### 设计/实现测试用例

# 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 |
| 2024-04-09 | 19:00-21:30 | 编写P1的isLegalMagicSquare函数 | 按计划完成 |
| 2024-04-10 | 20:00-22:30 | 编写P1的generateMagicSquare函数并写完测试 | 按计划完成 |
| 2024-04-12 | 09:00-11:30 | 完成P2的1-6 | 按计划完成 |
| 2024-04-12 | 13:30-18:00 | 完成P2的凸包和个人画图并用自带的测试类完成了测试 | 出现一点问题  但按计划完成 |
| 2024-04-13 | 09:00-12:00 | 完成P3的Person类 | 按计划完成 |
| 2024-04-13 | 14:00-17:00 | 完成P3的FriendshipGraph类 | 按计划完成 |
| 2024-04-13 | 19:00-21:00 | 完成P3的FriendshipGraphTest类 | 不能运行 |
| 2024-04-14 | 08:30-10:30 | 在检查代码后找到了P3测试类的问题，修改正确 | 按期完成 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的困难 | 解决途径 |
| 第一次接触到Git | 通过上网查询相关资料和教程视频一步步解决问题 |
| FriendshipGraphTest跑不了 | 尚未解决 |
| 人类的构造 | 参考CSDN |

# 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

* 第一次使用Java，可以读得懂，而且很快就掌握Java语言的一部分；
* 对Java的OOP获得一定的理解；
* 编程的过程学到很多各种各样的新知识，印象深刻。

## 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？

Java语言很有趣，作为第一次使用，但我觉得很好用；只不过它有分public和private类让我有点困难。

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

IntelliJ IDEA 是一款流行的集成开发环境（IDE），主要用于Java开发，虽然它通过插件支持各种编程语言。优点：丰富的功能集、智能代码辅助、可扩展性等；缺点：资源消耗大、语言支持有限等。

1. 关于Git和GitHub，是否感受到了它在版本控制方面的价值；

通过Git和GitHub，团队成员可以轻松地合作开发项目，跟踪代码的变化，解决问题，并保持代码库的整洁和组织。这两者的结合为软件开发过程带来了很大的便利性和效率提升。

1. 关于CMU和MIT的作业，你有何感受；

完成本次实验收获挺大，了解很多新的知识。

1. 关于本实验的工作量、难度、deadline；

考虑到上这门课的学生也上其他课程，老师布置的作业和实验难免有冲突，为了提升学生的心理健康和人体健康，可以考虑到简化实验、直观给出实现方法，毕竟做实验的目的是为了让学生们体验到Java的精彩。学生看到了、感受到了才更有兴趣并保持focus在这门课上。

1. 关于初接触“软件构造”课程；

编写准确高效的Java程序