

**2022年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 王煜菲 |
| 学号 | 120L030319 |
| 班号 | 2003010 |
| 电子邮件 | [1015487798@qq.com](mailto:1015487798@qq.com) |
| 手机号码 | 13683802098 |

**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：

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

Java JDK 11：由于需要官网登录太麻烦，死活整不好，采用了CSDN上某镜像。

IntelliJ IDEA：之前重装系统下JB顺手下了个，配置环境安装Chinese、Git、Junit等插件。但是测试时由于没有充分了解java的语言特性和IDEA正确的使用方式，helloworld反复失败。

在这里给出你的GitHub Lab1仓库的URL地址:

https://github.com/ComputerScienceHIT/HIT-Lab1-120L030319

# 实验过程

## Magic Squares

检验是否为合法输入：不能是负数；不能是小数；数与数以‘\t’分割。检验是否为合格幻方：行列数相等；行、列、左右对角线都和相等。采用文件输入输出流对已定txt进行检验测试。

用给定方法输出一个幻方txt，并对其进行检验。

### isLegalMagicSquare()

打开文件，并进行相应的异常处理。

判断是否为合法输入。首先，通过spilt.(“\t”)，一方面对数据进行处理，刨除无用信息，一方面对数据输入格式进行检验，如果用其他方式分割，得到的输入并非纯数字。将处理过的字符串转换为int类型，转化失败说明输入格式不对，例如有“ ”等分割数字符号，异常抛出。判读数字是否是负数。

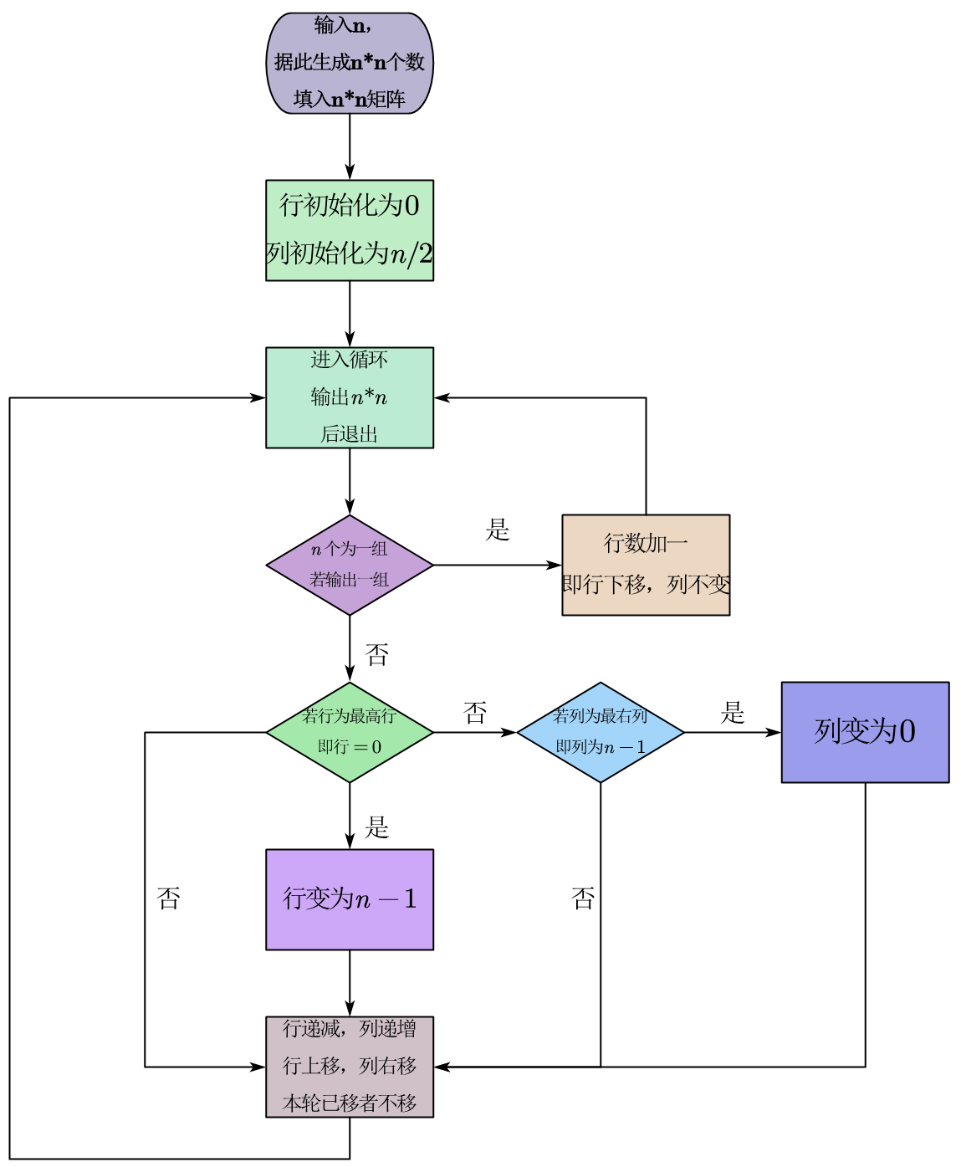
按行读取，计算第一行（已取得）的长度与总和，作为基准值。下面依次读取行，由于幻方应当是n\*n的矩阵，所以其列数与行数相等，依次读取判断是否输入合法，并将处理过的字符串存储入二维数组num，用于后续处理。在此过程结束后，如果发现还有未读取行，说明行数小于列数，如果i<j，说明列大于行，分别输出提醒后返回false。

利用二维数组num分别对行、列、左右对角线值进行考察，并分别输出错误提醒。 若经历检验为格式规范的合格幻方，返回true。

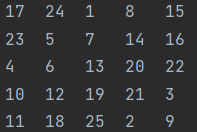
### generateMagicSquare()

判断是否为奇数，若为负数或偶数给出提示并返回false。

相对路径创造txt文件并写入。幻方构建思路如下。



按顺序输出从1至n\*n，通过不断改变输出的位置构造幻方。初始位置为（0，n/2）,行上移，列右移，以n个为一组，为避免重复，每n个调整一次初始位置，即在该回到初始位置时下移一位。当遇到矩阵边界时，保持原有趋势并绕至另一边。一个输出案例如下：



矩阵位置安排结束后，依次输出矩阵元素至txt中，并以“\t”分割。

## Turtle Graphics

学习使用turtl graphics，进行基本练习，如画正方形、多边形；进行复杂练习，通过对基本功能如计算角度等方法的一步步构建，完成计算数列转向角、计算凸包、画personal art等任务。

### Problem 1: Clone and import

从链接下载好P2文件，放入LAB1中src下。修改package名字，导入junit工具包，使文件能够正常使用不再飘红。

### Problem 3: Turtle graphics and drawSquare

初始方向为y轴正方向，以参数sideLength为步长，每画完一步旋转九十度，结束。

### Problem 5: Drawing polygons

完成方法calculateRegularPolygonAngle的构建，计算多边形每个角的大小为（边数-2）\*180/边数。

计算出正多边形单个角度，需要注意的是，我们画图需要的是外角。画图思路同上，画边，旋转。

### Problem 6: Calculating Bearings

完成方法calculateBearingToPoint的构建，通过Math.atan2现在坐标、目标坐标，可计算两点间夹角，再减去当前朝向，可以得出在当前点，从当前朝向转到目标点，需要多少角度。若角度小于0，则加上360。

处理输入数据，建立列表存储两两点间朝向转移角。调用calculateBearingToPoint函数计算朝向转移角，存入列表中。前朝向初始值为0，每轮循环中加上转移角进行更新，最后返回角度列表。

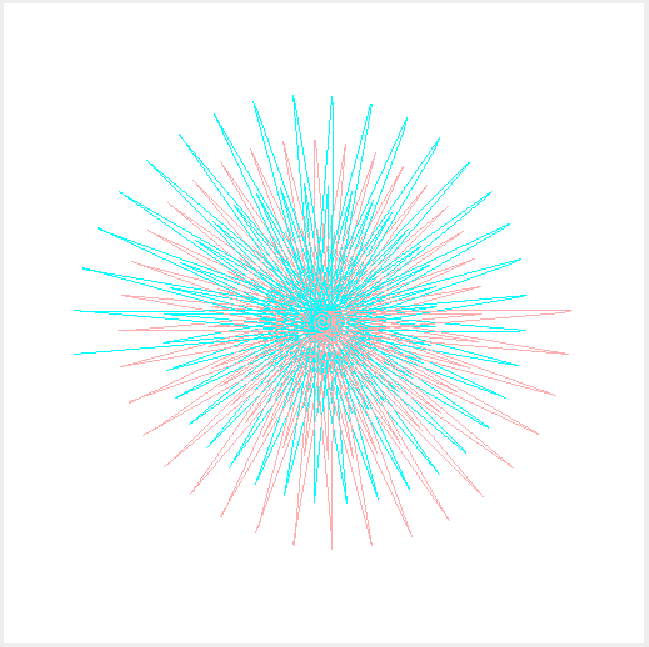
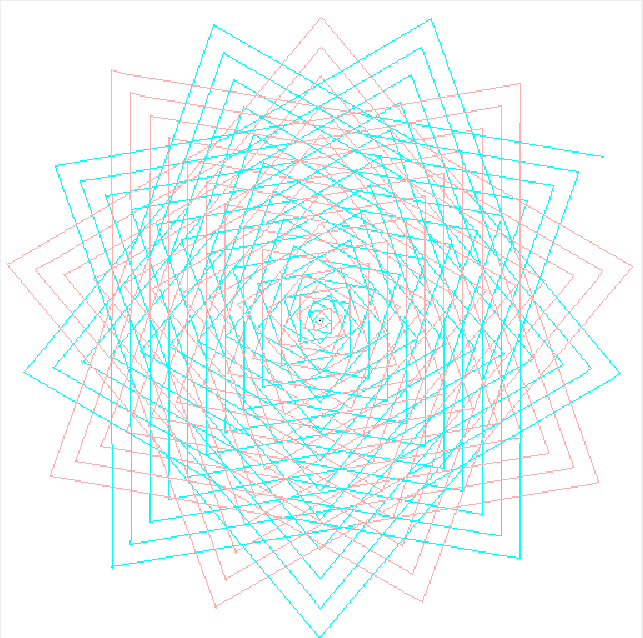
### Problem 7: Convex Hulls

参数为一系列点。计算点的凸包。

采用的算法思路如下：遍历选取左上角点，调用Calculating Bearings函数计算偏转角最小的点，更新基准点为该点，并更新当前转向。若有多个最小偏转角点，选取距离最大的。当待加入的点为已加入的点时，结束循环，说明已成功构建凸包。

### Problem 8: Personal art

思路为每走一定步长偏转一定角度，并可通过switch-case循环改变颜色。实际上整出了很多好玩的图案，有些还会出现先顺后逆的视觉错觉效果。

### Submitting

如何通过Git提交当前版本到GitHub上你的Lab1仓库。

## Social Network

刻画一个社交网络有向图，根据给出的案例和架构设计相应的类和方法。学习如何面向对象编程。

### 设计/实现FriendshipGraph类

#### 变量

private Map<Person, List<Person>> map = new HashMap<>();  
private List<Person> persons = new ArrayList<>();

分别记录图和节点。

#### addVertex

通过persons.contains检验是否是已有的结点，否则加入persons，为ta创建friends列表，并加入到图中。

#### addEdge

检查两个节点是否加入到图中，如果没有就提示退出。如果有，判断n2是否在n1关系中，若已有则不重复加入，没有则加入。

#### getDistance

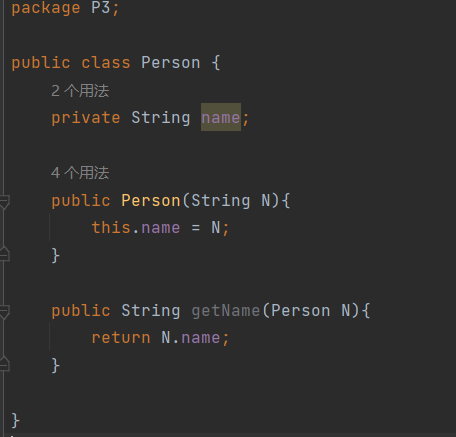
判断n1、n2是否都在图中，不在则报错退出。判断n1、n2是否为同一人，为则提示结束，返回距离为1。

采用深度优先遍历图，以深度衡量距离并返回。建立队列，首先将n1推进。进入循环，队列为空时退出，每轮取队列首节点，height初始化为0，每轮加一。遍历与当前点有联系的点，如果不在队列中则推入，如果等于n2，则返回height并退出。如果遍历结束后仍未退出，则从getDistance返回，提示两者无关系，返回-1。

while(!vsQueue.isEmpty()){  
 np = vsQueue.poll();  
 f\_length = map.get(np).size();  
 height++;  
 for(int i=0; i < f\_length; i++){  
 if(n2 == map.get(np).get(i)){  
 return height;  
 }  
 if(!sought.contains(map.get(np).get(i))){  
 sought.add(map.get(np).get(i));  
 vsQueue.offer(map.get(np).get(i));  
 }//视情况入队列  
 }  
}

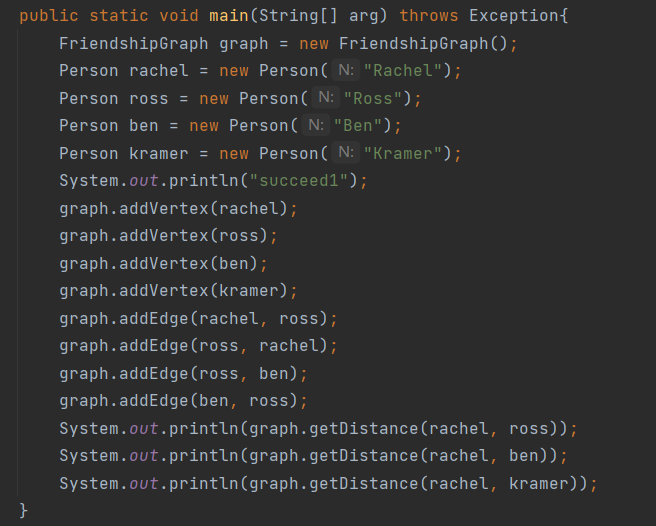
### 设计/实现Person类

此项大多根据FriendshipGraph的需求而设计。定义变量name，具有两个方法。



### 设计/实现客户端代码main()

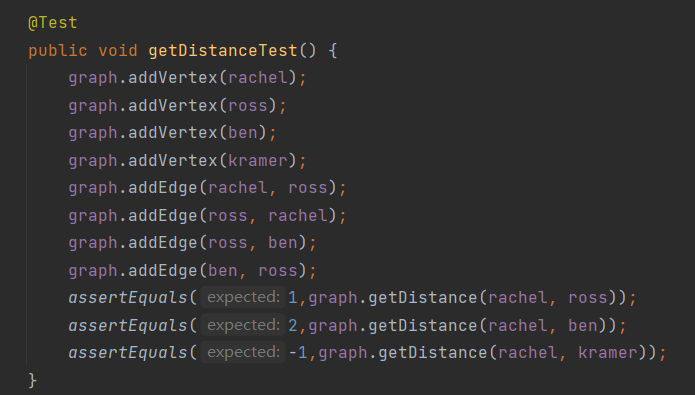
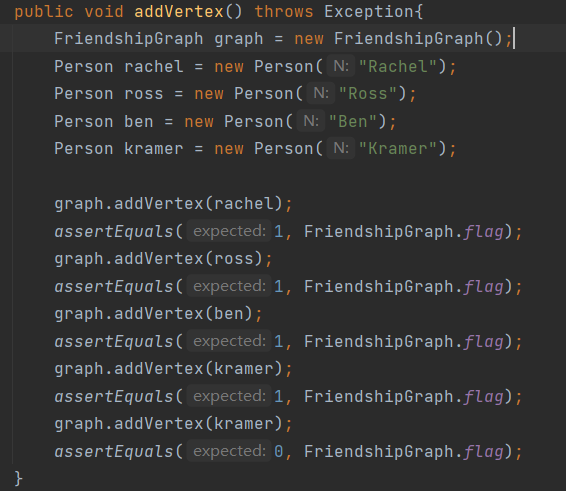
根据给定数据进行测试。



### 设计/实现测试用例

首先利益IDEA搭建好测试源根。

在进行测试时，发现一般的测试思路assertEquals不适用addVertex和addEdge，因为这两个函数返回值为void。查询学习可得，可通过引入全局变量并改变其在函数不同位置的值，通过对该全局变量assertEquals可进行测试，遂引入。



写到addEdge，发现初始化graph时，无法进行addVertex等操作，也就无法初始化，于是将graph挪到外面变成了全局变量。但addEdge和getDistance依然不通过，于是猜测是否因为测试内对Point重命名导致其不能识别，于是挪到外面，失败。进入addEdge内部，发现map值为0，发现是测试内赋值无效，又因为不能脱离方法使用函数，故只能选择每个测试重复命名。

# 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 |
| 2022-05-04 | 9:45-11:35 | 完成幻方性质代码任务，并调试；完成generateMagicSquare代码 | 按计划完成，稍稍推迟 |
| 2022-05-04 | 15:00-17:05 | 完成问题二除凸包任务 | 由于学习了一些知识延迟了半小时，但感觉值 |
| 2022-05-04 | 19:35-20:40 | 完成凸包任务 | If优先级逻辑问题导致debug多de了20分钟 |
| 2022-05-04 | 10：00-0：20 | 完成问题二个人设计；规划、完成问题3部分任务 | 遭遇List危机，延期半小时完成 |
| 2022-05-05 | 10:03-11:20 | 编写问题3getDistance函数，并进行测试、debug；加强对addEdge、addVertex函数的输入合法性检验。 | 遇到小问题，但如期解决 |
| 2022-05-05 | 19:00-21:48 | 重申代码，写测试案例 | 测试大危机（ |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的困难 | 解决途径 |
| 问题二凸包输出点始终不理想 | 一步步调试，首先发现偏转角异常，接着发现是初始点异常，最后发现是if优先级以及逻辑问题错误导致的异常问题，两次修改后输出正常。 |
| 问题3定义变量始终报错，不能识别List | 百度查询问题始终无果，大部分建议是配置jdk和junit但我已经配置。尝试不用List但发现实在太麻烦了，尝试后放弃。  又查询List适用范围，改为import java.io.\*，问题解决 |
| 问题3getDistance函数实验结果不理想 | 查阅资料，复习了queue.poll和queue.peek的区别，同样是取队列首元素，但一个弹出一个不弹出，影响很大。 |

# 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）

在本次实验中，较为全面的学习了java，从基本语法、文件输入输出流、使用工具（调包）、面向对象编程等角度学习了Java，让我意识到之前做的准备是不充分、不深入的，并使我发现自己在面向对象编程方面的能力缺陷，需要多加补足。

本次实验还让我意识到整体架构的重要性，对课堂上所学的理论知识有了进一步的了解。在本次实验中，我还渐渐养成了注释的重要性。以及可以说一万遍的注意细节。

## 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？

目前我还不能习惯这种面向对象+类的编程方式，写第三个问题时如何构建模块一度成为一大问题。但不可否定的是这种方法确实具有很强的安全性、规范性、模块化。就我浅薄的理解而言，Java的某些特性使他在处理某些问题时显得麻烦、绕弯，而某些语法又不是那么的简洁方便。

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

我用的是IDEA。

优势就是人性化，细心方便，很多我的疏漏之处它能帮助我很快发现改正，且界面用户友好，操作便捷。

缺点大概是强制飘红，有些地方必须要按照它的规章制度走。而且编译运行速度比较慢，对于我这种小项目有点杀鸡焉用牛刀了。

1. 关于Git和GitHub，是否感受到了它在版本控制方面的价值；
2. 关于CMU和MIT的作业，你有何感受；

挺有水平的。作业设计很好地兼顾了教学和实践，能让学生在教学中充分领悟Java的某些特性，并通过精巧设计拓宽学生对Java的理解和学习。拿我举例，在后两个实验中被迫学习了List，Map、Point、Set等数据结构及其相关函数知识。并学习了turtle，挺好玩的。

1. 关于本实验的工作量、难度、deadline；

由于本周比较忙而且我个人事情比较多，本来想周二周三写完然后复习考试结果周四了还弄不完…确实是很大负担。对于我这种没接触过Java，学习全靠自学和上课听课的小白来说，最难的不是代码书写（第三题除外，那个考的就是思想），而是如何合理利用Java特性，编写出Java体制下的合适代码。这很有挑战性，挑战性也是使他有趣的一个原因，但由于其难度和任务量使它在很多时候成为我一大负担，使我感到疲惫又无可奈何。我不能透彻的学习，很多时候往往是出现问题了反过来根据问题去针对性学习。这可能更点对点，但是也浪费了时间，并使我失去了获得整体性视野机会。

而且接触新事物新工具本身的难度也是一大障碍。

1. 关于初接触“软件构造”课程；

很有挑战性，值得好好学。