

**2020年春季学期  
计算机学院《软件构造》课程**

|  |  |
| --- | --- |
| 姓名 | 海龙 |
| 学号 | 1180300513 |
| 班号 | 1803005 |
| 电子邮件 | 769365437@qq.com |
| 手机号码 | 15765521978 |

**目录**

[1 实验目标概述 1](#_Toc29325521)

[2 实验环境配置 1](#_Toc29325522)

[3 实验过程 1](#_Toc29325523)

[3.1 Magic Squares 1](#_Toc29325524)

[3.1.1 isLegalMagicSquare() 2](#_Toc29325525)

[3.1.2 generateMagicSquare() 3](#_Toc29325526)

[3.2 Turtle Graphics 3](#_Toc29325527)

[3.2.1 Problem 1: Clone and import 3](#_Toc29325528)

[3.2.2 Problem 3: Turtle graphics and drawSquare 3](#_Toc29325529)

[3.2.3 Problem 5: Drawing polygons 4](#_Toc29325530)

[3.2.4 Problem 6: Calculating Bearings 4](#_Toc29325531)

[3.2.5 Problem 7: Convex Hulls 5](#_Toc29325532)

[3.2.6 Problem 8: Personal art 5](#_Toc29325533)

[3.2.7 Submitting 5](#_Toc29325534)

[3.3 Social Network 5](#_Toc29325535)

[3.3.1 设计/实现FriendshipGraph类 6](#_Toc29325536)

[3.3.2 设计/实现Person类 6](#_Toc29325537)

[3.3.3 设计/实现客户端代码main() 6](#_Toc29325538)

[3.3.4 设计/实现测试用例 6](#_Toc29325539)

[4 实验进度记录 7](#_Toc29325540)

[5 实验过程中遇到的困难与解决途径 8](#_Toc29325541)

[6 实验过程中收获的经验、教训、感想 8](#_Toc29325542)

[6.1 实验过程中收获的经验和教训 8](#_Toc29325543)

[6.2 针对以下方面的感受 8](#_Toc29325544)

# 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Gi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 环境：JDK13

IDE 软件：Eclipse 2019.12

<https://github.com/1180300513/Lab1-1180300513>

实验1之前的准备工作，需要配置环境变量，安装jdk，eclipse以及git。

三者的下载没什么难度，lab0的实验说明上给出了下载地址，对于第一次接触这些软件的人来说比较生疏，所以虽然设置的时候参考了许多教程，但还是没什么把握能够成功运行。

英语不好，所以仅看实验说明和指导的部分还是很难完成这些工作，最后还是百度筛选了一些觉得靠谱的文章完成了实验环境的构造。

# 实验过程

## Magic Squares

本问题要求实现判断一个nxn矩阵是否是一个幻方，并且设计一个n为奇数的幻方的生成算法。

该问题分为两个部分完成。

第一部分是读取5个txt文件中的数据判断是否为幻方，包括将txt中的内容存入int型的二维数组中以及判断是否满足幻方的定义（包括数据是否为整数，是否以“\t”分隔，行，列，对角线之和的判断），

第二部分是对给定的用来生成幻方的函数进行分析然后将生成的矩阵写入文件，主要包括分析函数的参数n为奇数、偶数以、负数的情况以及产生的异常。

该问题主要涉及了读写文件的一些操作以及数组还有数据类型之间的转换以及循环结构和分支结构的应用等一些基本的Java语言的运用，通过这些基础的Java语言的应用，熟悉Java语言。

1.读文件，按行读取。

2.写文件

3.按行从文件中读入矩阵的内容，对于每一行，按照"\t"进行split，得到一个字符串数组,然后把得到的所有数组保存到一个String容器内（容器的每一项都是一个字符串数组）。

4.判断异常的模板

### isLegalMagicSquare()

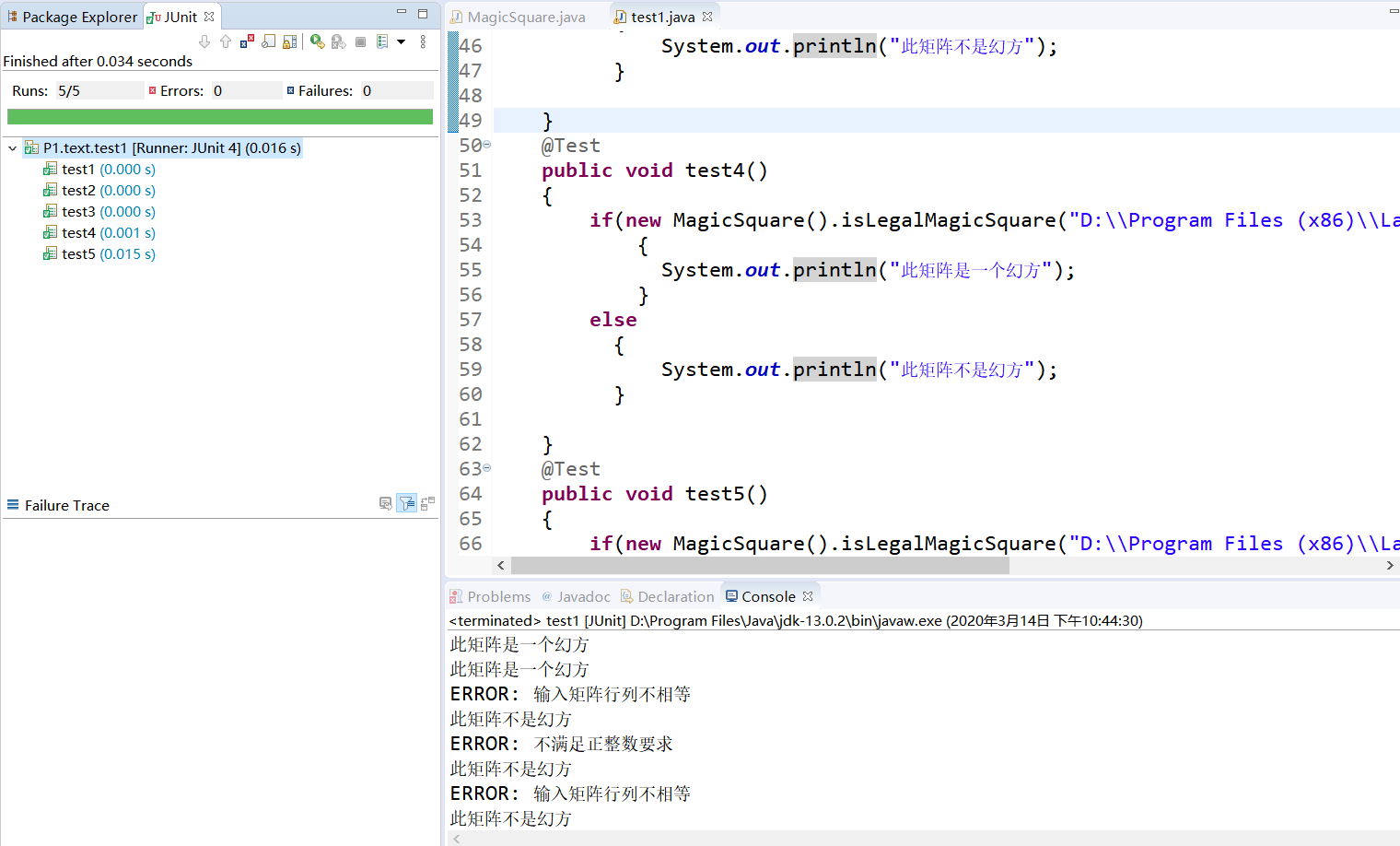
1.使用java输入流按行读入文件

对于每行尝试使用\t进行分割，对于分割后的每一项，如果发生Exception， 有两种可能原因，一是该矩阵不满足使用"\t"分隔，一是该项包含非法数字。

2.将第一行的数字个数作为基准数目n，在读入的过程中判断该矩阵是否是n\*n矩阵。

3.使用三次for循环分别检查行、列、对角线数字的和是否相等。

结果：

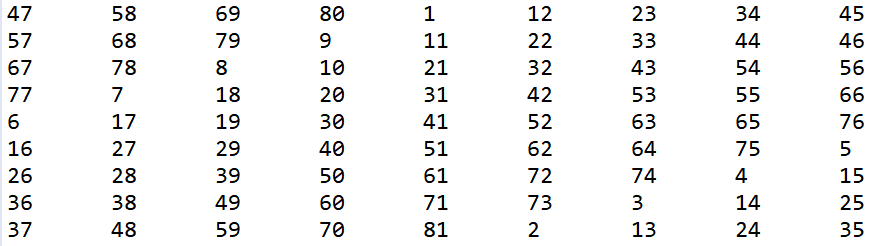


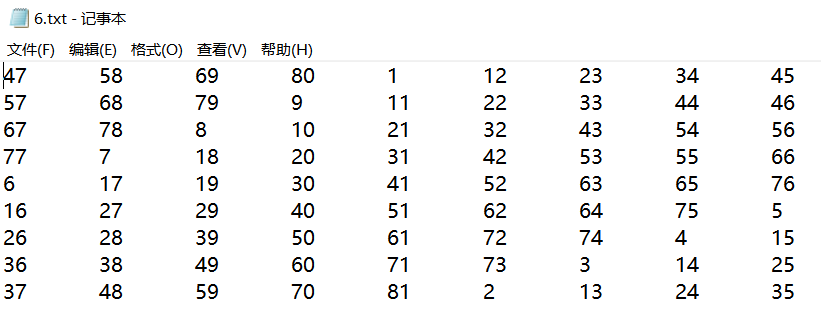
### generateMagicSquare()

思路：使用罗伯法构造一个n为奇数的nxn幻方。

生成的文件写入6.txt中。







## Turtle Graphics

这个实验是为了训练使用java自带的画图库。主要考点：计算几何基础知识，正多边形的内角外角，java函数调用绘制多边形，计算向量之间的夹角，计算凸包，使用Junit进行单元测试。

### Problem 1: Clone and import

代码在GitHub中点击按钮就可以下载到本地，也可以使用URL下载到本地。 从本地创建git仓库：

1. git init初始化本地仓库
2. 2.git add remote origin 添加远程库源
3. 3.在远程仓库创建master分支
4. 4.git pull origin master将远程仓库同步到本地
5. 5.git add->git commit->git push 将本地文件加入本地仓库，将本地仓库同步到远程仓库。

此时git仓库本地与远程可以同步。

### Problem 3: Turtle graphics and drawSquare

问题思路：这个方法考察的是基本的turtle的forward和turn函数，。向前走再转90度，每次转向都是一个方向即可。

1. forward (steps) 首先根据当前坐标和朝向计算走steps步之后的坐标，根据起始点生成线段类lineSeg，根据朝向以及lineSeg构造Action添加到ActionList中等待执行。
2. turn(degrees) 更改当前朝向，向ActionList中添加turn的Action。
3. drawSquare，调用drawRegularPolygon(turtle,4,sideLength)

### Problem 5: Drawing polygons

问题思路：这个任务包含了几个方法

1 calculatePloygonSidesFromAngle

 给定内角度数要求计算出边数。

2 drawRegularPolygon

根据给定的长度和边数画出多边形，根据我们上面的函数我们可以很快的完成。

3 calculateRegularPolygonAngle

给定边数要求计算出来边数。

calculateRegularPolygonAngle 计算正多边形的内角，（边数-2）/sides。

drawRegularPolygon ：调用calculateRegularPolygonAngle（sides）计算正多边形的内角为rotation，调用turtle.forward和turtle.turn进行前进和转向。

### Problem 6: Calculating Bearings

问题思路：

1 calculateHeadingToPoint

这个函数是计算给定两个向量之间的方向角，通过给定的点进行处理。

2 calculateHeadings

函数计算边构成的向量之间的夹角，得出转动的角度的大小，使用上一个函数计算出角度。

### Problem 7: Convex Hulls

问题思路：使用gift-wrapping算法实现convexHull

1. 首先根据坐标选出最左下角的点作为点B。添加一个点A，A位于B点稍下的位置。将(A,B)作为向量Vec1。
2. 扫描所有的不处于凸包中的点，计算A点与该点之间的夹角，夹角通过两向量点击与向量的模计算cos然后通过acos函数获得，找到形成夹角最小的点，如果有多个选择距离当前**直线距离最远**的那个点（其实可以选择距离前一个点最远的点）。将点B作为新的A，点C作为新的B，再次循环。
3. 当新找到的节点是最左边的初始节点时结束循环。

### Problem 8: Personal art

使用下面代码实现drawPersonalArt：

turtle.color(PenColor.PINK);

for(int i = 0; i < 15; i++){ // i < 15 to limit size of shapes

drawRegularPolygon(turtle, i, i\*10);

drawRegularPolygon(turtle, i\*10, i);

}

### Submitting

先用git clone [URI 将在github](https://github.com/1170300814/Lab1-1170300814.git%E5%B0%86%E5%9C%A8github)上创建的库克隆到本地，然后将要上传的文件移到目录下。

然后使用命令

Git add -A

Git commit -m “lab1”

Git pull –rebase origin master

Git push origin master

就能将本地仓库同步到云端

## Social Network

实现Person和FriendshipGraph两个类，模拟社交网络，判断一个人的社交活动和影响力。主要考点：java类的实现，java基本数据结构，BFS求最短路。

### 设计/实现FriendshipGraph类

设计思路：

1. 存储结构：使用List<Person> persons存储所有节点。Map<String,Integer> namePool 用于记录所有的用户名字，从而提示重名错误。
2. adjGraph ：表明这是一个图，统计点的个数
3. addVertex：添加节点，向persons List中添加一个person对象
4. addEdge：将存在图中的两个人联系起来。
5. getDistance：特殊判断stPerson是否与edPerson相等，相等则返回0。使用Queue与Map实现BFS求最短路。Map用来存放距离源点的最短路同时作为当前点是否已经访问过的标志。

### 设计/实现Person类

设计思路：

存储结构：Person方法将名字赋给类中的name

1. String nameString存放当前Person的名字，List<Person> neighborList存放当前节点为起始点的所有出边的另一端的Person。
2. addNeighbor：向neighborList中添加一个联系人。
3. getName()：获得Person的name的getter
4. getNeighList()：获得Person的相邻节点列表的getter

### 设计/实现客户端代码main()

Main函数代码如下：

public static void main(String[] args)

       {

             FriendShipGraph graph = new FriendShipGraph();

           Person rachel = new Person("rachel");

           Person ross = new Person("ross");

           Person ben = new Person("ben");

           Person kramer = new Person("kramer");

           graph.addVertex(rachel);

           graph.addVertex(ross);

           graph.addVertex(ben);

           graph.addVertex(kramer);

           graph.addEdge(rachel, ross);

           graph.addEdge(ross, rachel);

           graph.addEdge(ross, ben);

           graph.addEdge(ben, ross);

           System.out.println(graph.getDistance(rachel,ross));

           System.out.println(graph.getDistance(rachel,ben));

           System.out.println(graph.getDistance(rachel,rachel));

           System.out.println(graph.getDistance(rachel,kramer));

       }

### 设计/实现测试用例

测试思路：

1. 测试单边联通的特殊情况
2. 测试重名的情况
3. 测试不能到达的情况
4. 测试到达路径有多条的情况

# 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 |
| 2020-3-6 | 14:00-15:30 | P1 | 自学java |
| 2020-3-11 | 全天 | P1 | 完成 |
| 2020-3-12 | 10：00-19:00 | P2 | 遇到困难，未完成 |
| 2020-3-13 | 12：00-22:00 | P2 | P2未完成 |
| 2020-3-14 | 10：00-22:00 | P3 | P2完成，P3未完成 |
| 2020-3-15 | 10：00-12:00 | P3 | P3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 |
| P1实验中测试文件的读取以及生成矩阵写入文件 | 采用相对项目根目录读取的方式，用new File(“”).getCanonicalPath()获取项目的根目录，然后拼接src下具体文件路径 “/src/P1/text/1.txt” |
| P2 Convex Hulls中对凸包上同一条直线出现多个点情况的处理 | 如果出现最少角度的直线上出现多个点，则先将这些点加入一个暂存的集合，循环完毕，遍历该集合，找出该集合中这些点到当前点的距离最大的那个点，即为下一个要找的点。 |
| 不太会Java | 通过看Java编程思想和看mooc完成 |

# 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

这次的实验需要java而我是第一次接触，所以在完成上遇到了很大的困难。参考了很多的资料。仍然在完成上遇到了困难，但是我相信情况会越来越好。

利用一些工具如：Git、eclipse、Junit单元测试等能够显著的提高编程效率，熟悉工程的管理、了解Git的运作方式也至关重要。

编写完程序之后并不是结束，最好使用Junit进行单元测试。

程序测试一定要综合考虑多种边界情况，输入的标准与否是程序首先应该要确定的。

Java中文件输入输出流需要最后close否则会造成内存泄漏。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

是

1. 关于Eclipse IDE

不是很好用

1. 关于Git和GitHub

之前没有使用过Git，对其还是相对陌生的，但是实验要求不高，只需要 掌握git的几个比较常用的命令即可；github平时接触过，但运用也不是特别熟练，还需要课下勤于使用。

1. 关于CMU和MIT的作业

虽然是第一次试验，但是对JAVA新手来说设计挺难。

1. 关于本实验的工作量、难度、deadline

实验工作量不轻松，因为是第一次使用JAVA和eclipse及git，所以操作起来很有难度。

1. 关于初接触“软件构造”课程

希望老师能讲一些JAVA语言中的重难点，以及怎么与C映照着学习JAVA。