

**2018年春季学期  
计算机学院大二软件构造课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 王世川 |
| 学号 | 1163710227 |
| 班号 | 1637102 |
| 电子邮件 | 597944330@qq.com |
| 手机号码 | 17766520412 |

**目录**

[1 实验目标概述 1](#_Toc506282558)

[2 实验环境配置 1](#_Toc506282559)

[3 实验过程 1](#_Toc506282560)

[3.1 Magic Squares 1](#_Toc506282561)

[3.1.1 isLegalMagicSquare() 1](#_Toc506282562)

[3.1.2 generateMagicSquare() 1](#_Toc506282563)

[3.2 Turtle Graphics 1](#_Toc506282564)

[3.2.1 Problem 1: Clone and import 2](#_Toc506282565)

[3.2.2 Problem 3: Turtle graphics and drawSquare 2](#_Toc506282566)

[3.2.3 Problem 5: Drawing polygons 2](#_Toc506282567)

[3.2.4 Problem 6: Calculating headings 2](#_Toc506282568)

[3.2.5 Problem 7: Personal art 2](#_Toc506282569)

[3.2.6 Submitting 2](#_Toc506282570)

[3.3 Social Network 2](#_Toc506282571)

[3.3.1 设计/实现FriendshipGraph类 2](#_Toc506282572)

[3.3.2 设计/实现Person类 2](#_Toc506282573)

[3.3.3 设计/实现客户端代码main() 2](#_Toc506282574)

[3.3.4 设计/实现测试用例 2](#_Toc506282575)

[3.4 Tweet Tweet（选作，额外记分） 3](#_Toc506282576)

[4 实验进度记录 3](#_Toc506282577)

[5 实验过程中遇到的困难与解决途径 3](#_Toc506282578)

[6 实验过程中收获的经验、教训、感想 3](#_Toc506282579)

# 实验目标概述

本次实验通过求解四个问题（其中一个可选），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

安装了eclipse，GitHub Editor，Jre9，Jdk9

本次实验在 GitHub Classroom 中的 URL 地址为：

https://github.com/ComputerScienceHIT/Lab1-1163710227

由于已经有相关经验所以安装没有出现问题。

# 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（但千万不要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

该任务有两个要求，要求一考察编程能力，针对一类事物（幻方）抽象出判断幻方是否合法的方法，并且锻炼了异常处理能力、容错性健壮性编程能力，学会让程序遇到错误时优雅地提示错误信息并退出。要求二则考察了对他人代码的阅读、理解、学习、复用能力。

### isLegalMagicSquare()

思路：

根据题目要求，整体的流程分为三大部分，一是**读取和存储数据**，二是**判断是否为幻方**，三是**处理各种错误**。

过程：

步骤1：

因为将要读取的数据的大小未知，所以应当使用动态数组完成。

同时为了熟练面向对象编程，本次存储数据的动态数组是我**自行创建的MyList类**，MyList类用于存储整数类型数组，能够任意扩增容量，实现增删查改和输出，能够**基本满足动态数组的需求**。

读取数据可以通过将文件流重定向到Scanner，然后按照一般的读取方法读取，文件不存在时，需要报错并返回false退出函数。

步骤2：

**判断是否为幻方**，首先需要判断行列数是否一致，每行数字个数是否有残缺，是否有不合乎要求的输入，这些在读取数据的过程中同时完成。

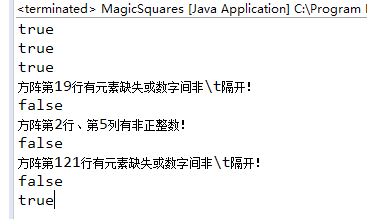
然后是判断每行相加和，每列相加和，对角线相加和是否满足幻方的条件。任何一条不和规范都可以return false，但应该注意的是，为了**保持代码的简洁可读**，应当**适当遵循单出口原则**，减少程序中的return出口。

步骤3：

最后对以上各种情况进行错误处理，当发现有不合条件的情况时，尽可能详细地提示出错误，**避免程序因为异常终止退出**，在异常发生前就应当提前做好准备。

结果：

对准备好的1.txt——6.txt进行测试，结果如下：



如上，测试通过。

### generateMagicSquare()

思路：

看到这个题目的时候，我首先上网了解了幻方生成算法，本题示例代码使用了奇数阶幻方（劳伯法）生成幻方。

过程：

流程图：

申请空间int[n][n]大小的方阵

初始化坐标到第一行中间

i=1;i<n\*N;i++

输出幻方

在当前坐标填入数字i，再将坐标移动至右上一格

坐标是否超出边界

调整坐标回到范围内。

解释：

该函数所用的填法是劳伯法。填写的方法是：

把1（或最小的数）放在第一行正中；按以下规律排列剩下的(n×n－1)个数：

（1）每一个数放在前一个数的右上一格；

（2）如果这个数所要放的格已经超出了顶行那么就把它放在底行，仍然要放在右一列；

（3）如果这个数所要放的格已经超出了最右列那么就把它放在最左列，仍然要放在上一行；

（4）如果这个数所要放的格已经超出了顶行且超出了最右列，那么就把它放在底行且最左列；

（5）如果这个数所要放的格已经有数填入，那么就把它放在前一个数的下一行同一列的格内。

错误1：

数组越界。

错误2：

定义了非法的数组大小（小于零）

步骤5：

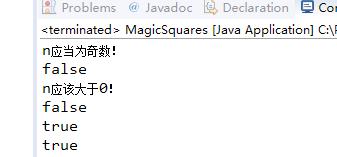
修改代码，使之能够将结果输出到文件。只需要将结果输出到文件打印流即可。

步骤6：

修改代码使之报错的时候能够优雅地退出，并详细地输出错误原因。，针对各种错误用if发现并报错即可。

结果：

分别以6，-1，5测试该函数，观察函数的返回值和报错信息，并调用上一个函数检查生成的幻方是否合法：



测试通过，正确姿势了错误的种类，成功生成的幻方也通过测试。

## Turtle Graphics

这题是MIT软件构造课的作业，从github克隆项目开始，通过已有的信息，调用别人写好的方法进行编程。模拟了多人合作编程中，通过理解和学习他人代码，完成自己部分的过程。

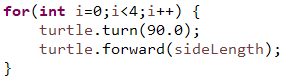
### Problem 1: Clone and import

由于我们不是MIT的学生，无法访问他们的仓库，所以选择从实验报告给定的仓库地址中，下载代码，解压并整理到自己的仓库中，代码的包名因为目录结构的改变需要稍作修改。

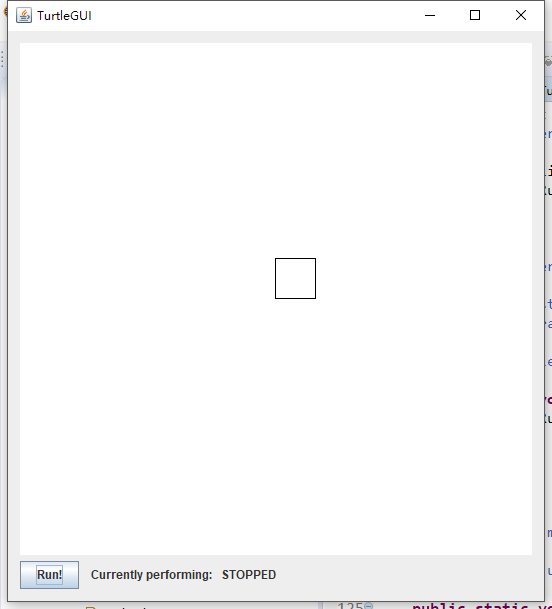
### Problem 3: Turtle graphics and drawSquare

查看Turtle.java了解各个函数的 用法，修改drawSquare方法体，调用相关函数即可。

因为是方形，所以重复调用forword走一定距离和turn转90°

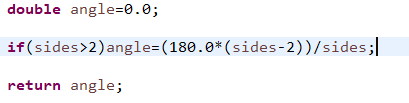


结果如下：



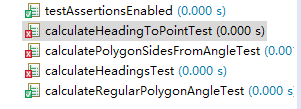
### Problem 5: Drawing polygons

**步骤1：**按照题目要求，首先实现计算多边形内角的函数。



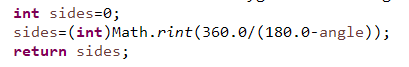
此处题目没有要求做出异常处理，但实际上，我们不能总是保证他人调用此函数时传入的边数不会不合法，所以此处采用比较温和的方式，如果不合法的话，返回角度0。

**步骤2：**运行Junit，测试函数是否通过

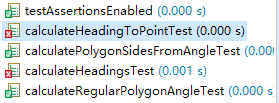


该函数测试通过

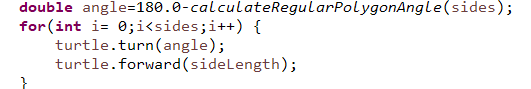
**步骤3：**编写上一个函数的逆函数，运行Junit测试该函数：



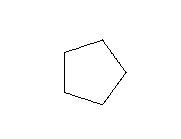
**测试如下：**



**步骤4：**完成drawRegularPolygon方法，并运行测试：



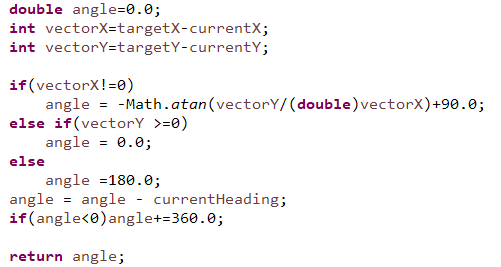
测试结果如下：



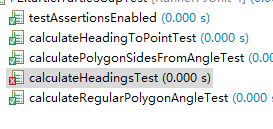
### Problem 6: Calculating headings

步骤1：先要能够计算海龟从所在位置要走向目标位置所需要旋转的角度，海龟位置可以用一对直角坐标表示，面朝向的方向用一个以北向为初始方向的夹角表示，则此处需要计算的是，海龟面向的方向和与目标位置的向量顺时针的夹角。

代码如下：

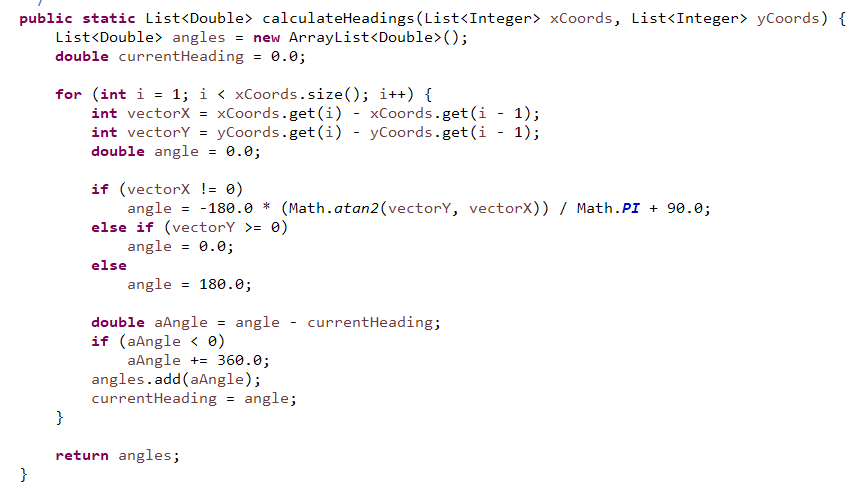


测试结果如下：

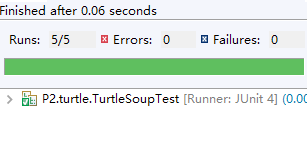


步骤2：接下来继续完成难度更高的情况，输入一组坐标，默认海龟初始的方向是正上方，返回一组角度，表示海龟遍历这些点需要做的每次转向。

通过调用上一个函数，编写代码如下：



测试结果如下：

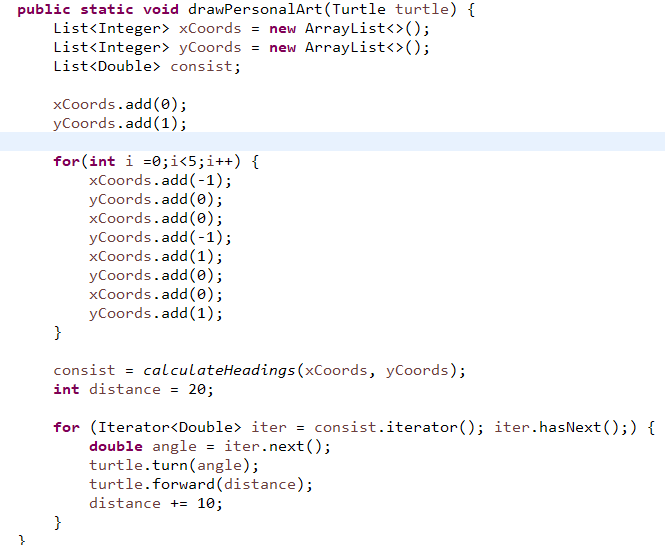


测试通过。

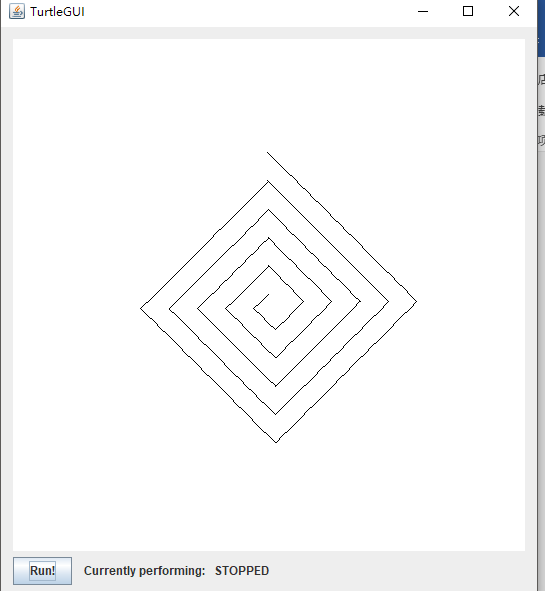
### Problem 7: Personal art

步骤一：设计一个简单的艺术图形，我设计的是一个螺旋的方形。

步骤二：完成对应的代码，操控海龟画出图形：



只需要一个初始点和四个遍历点，就可以构造出四个向量，使海龟转圈，用for循环控制转圈次数，**此处以五圈为例，测试结果如下**：



### Submitting

首先将eclipse工程文件夹中的源代码复制到本地git仓库，再检查本地git仓库的changess，然后commit提交变更，最后push到远程仓库上即可。

## Social Network

这个任务安排了一个需求，要求我们围绕这个需求设计解决方案。综合考察了软件设计、数据结构和算法设计的能力。

### 设计/实现FriendshipGraph类

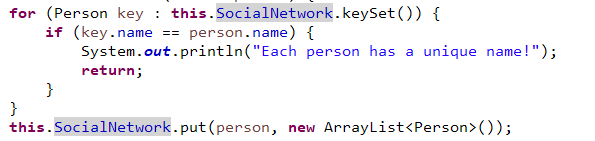
思路：

题目提示需要用到广度优先搜索和许多技巧，根据这些提示，很容易想到一键多值图的数据结构，每个person为一个键，该person的所有friends为值。

过程：

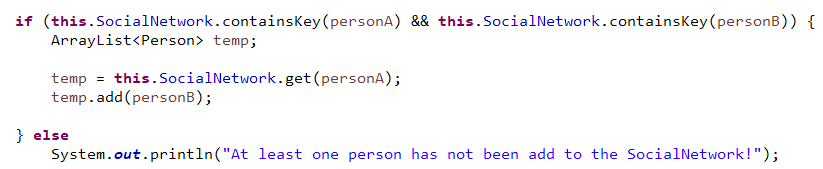
步骤1：

接下来完成将个人添加入图中的函数addVertex(Person person)，需要检查这个人是否已经出现过，确保每个人有一个确定的名字。



步骤2：

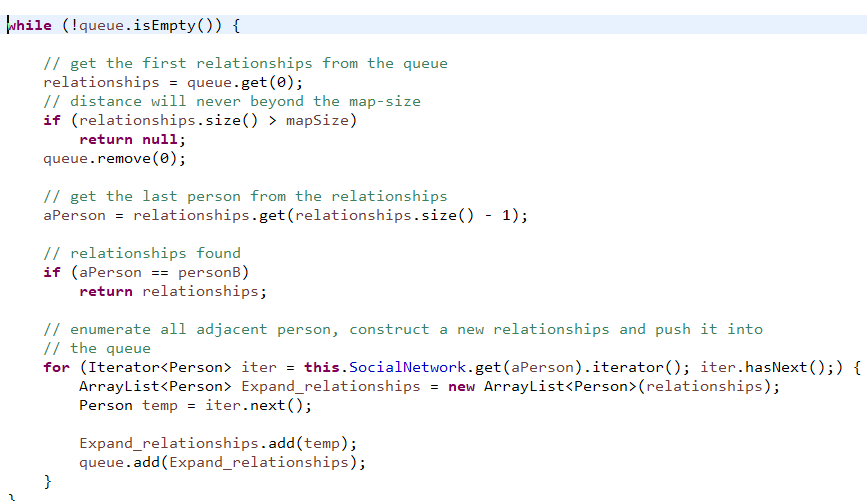
实现添加每个人关系的函数addEdge(Person personA, Person personB)，某两个人为朋友时根据题目要求，要调用该函数两次使二者互相加入对方的值中，在底层要求是是有向图。



同时需要排除比如有其中一个人并未加入图中的情况，即不能和不存在的人做朋友

步骤3：

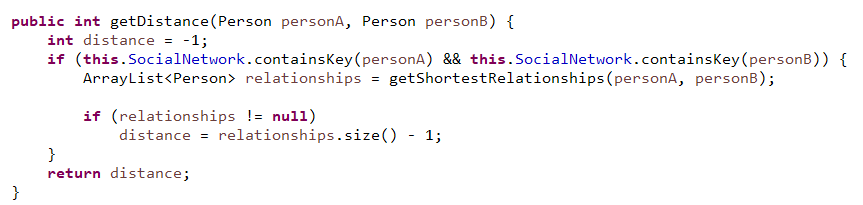
为了计算两个人之间的距离，并且在此之上，考虑到未来可能的对两人之间关系路径的索引，先构造一个可以返回二者之间最短路径的方法getShortestRelationships(Person personA, Person personB)，通过广度优先搜索，从personA开始，将A有关系的人组成路径组(A->ni),i=123…，再将每个ni有关系的mj接在ni后面组成路径组(A->ni->mj),i=123…,j=123….，知道有一个路径的最末端正好是PersonB，则搜索结束。



值得注意的是，路径的长度不会大于图中的节点数，如果大于节点数还没检索到PersonB，那么就可以确定这两个人之间没有关系。

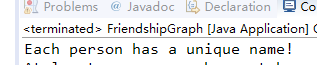
步骤4：

查询二者路径长的方法可以基于上一个函数来完成，通过调用上者，然后返回上一个函数返回的路径的长度即可



步骤5：

测试是否能做到不同人不同名字



### 设计/实现Person类

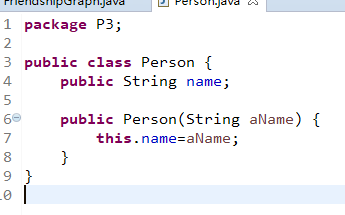
思路：

Person类至少包含了个人的名字，当然也可以拓展出更多的属性和方法。

过程：

步骤1：

实现Person类，该类需要一个类变量表示这个人的名字。



### 设计/实现客户端代码main()

思路：

简单假定几个人，插入一些关系，然后看看能否正确运行

过程：

步骤1：

按照实验报告构建一个框架：

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) + ":1");

// should print 1

System.out.println(graph.getDistance(rachel, ben) + ":2");

// should print 2

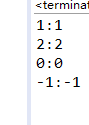
System.out.println(graph.getDistance(rachel, rachel) + ":0");

// should print 0

System.out.println(graph.getDistance(rachel, kramer) + ":-1");

// should print ‐1

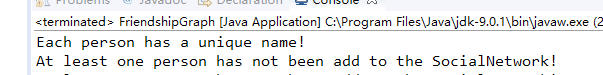
测试通过：



步骤2：

按照实验指导要求，修改掉一个人名使之重复，运行上述代码：





### 设计/实现测试用例

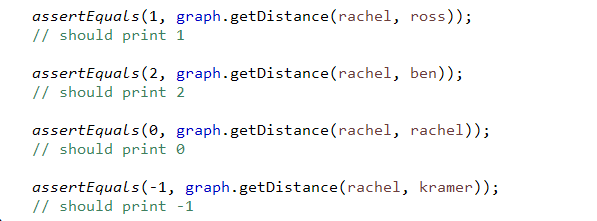
思路：

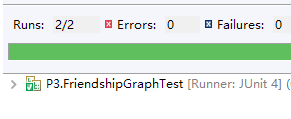
按照之前main函数编写即可。

过程：

步骤1：

编写Juint测试代码





测试通过。

## Tweet Tweet（选作，额外记分）

完成了Filter类

# 实验进度记录

请尽可能详细的记录你的进度情况。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 计划任务 | 实际完成情况 |
| 2018-02-26 | 14:00-15:30 | 实验课完成环境配置、阅读实验手册，下载实验材料。 | 按计划完成 |
| 2018-03-05 | 13:45-15:30 | 完成实验任务一 | 按计划完成 |
| 2018-03-08 | 10:00-11:45 | 完成实验任务二 | 延迟一小时完成，因为一边上课一边写分心。 |
| 2018-03-11 | 10:00-00:30(+1d) | 完成实验任务三 | 主要部分按预期完成，Junit出现错误未能排除。 |
| 2018-03-12 | 13:45-15:30 | 完成实验任务三和实验报告的撰写 | 按计划完成 |

# 实验过程中遇到的困难与解决途径

遇到的主要困难是对Java不熟悉，但是经过一段时间的学习，现在对Java很快就适应了。

# 实验过程中收获的经验、教训、感想

本节除了总结你在实验过程中收获的经验和教训，也可就以下方面谈谈你的感受（非必须）：

1. Java编程语言是否对你的口味？

个人对java语言并无好感，报错还不够人性不够具体，性能上也不算特别优秀，语法也没有很厉害的优化，写起来跟C一样感到疲惫，就从用不同语言实现相等功能的程序的实验结果看来，java花费的时间似乎比较多，虽然没有具体分析，但是总体体验上java确实不是一个可以开心写代码的语言。

1. 关于Eclipse IDE

用来写java确实比较方便，写久了现在也非常熟悉，没有特别的体验值得一提。

1. 关于Git

免费的代码仓库，还可以阅读大量的开源代码，做个人主页、做博客也非常够用，程序员神器，很早就开始用了，治百病。

1. 关于CMU和MIT的作业

难度比较大，要做好是需要花费比较多精力的，但是收获颇多，值得一做也值得写个博客分享心得和知识。

1. 关于本实验的工作量、难度、deadline

工作量比其他专业的计算机相关课程的实验内容确实是多了不少，但是在本专业的话，这样工作量的实验，也早就体验过了，能够接受。上学期的计算机系统课一个实验的安排时间都是十个小时。

1. 关于初接触“软件构造”课程

还不是很能理解这个课程要讲授的内容和目的，听课也有很多新词、英文，速度也挺快，以至于有点消化不良，需要适应。