

**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 | 陈曦 |
| 学号 | 1160300813 |
| 班号 | 1603008 |
| 电子邮件 | 1091892346@qq.com |
| 手机号码 | 18846080652 |

**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，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

简要陈述你配置本次实验所需环境的过程，必要时可以给出屏幕截图。

在本地机器安装相应的开发环境（JDK、Eclipse、Git）并配置、调用命令提示符并运用、了解Git工作流程。

注册个人 GitHub 账号 、在 GitHub Classroom 中获取实验的 private repositories 、本地 Eclipse 开发、Git 管理、提交至 GitHub 仓库 。

特别是要记录配置过程中遇到的问题和困难，以及如何解决的。

在提交代码后总是不懂更新，之后出现很多问题。就将库删掉再创建库。

对命令提示符的运用不熟练，总是会报错。

在这里给出你的GitHub Lab1仓库的URL地址（Lab1-学号）。

<https://github.com/ComputerScienceHIT/Lab1-1160300813>

按照提示建立自己的 Lab1 仓库并关联至自己的学号。

本地开发时，本次实验只需建立一个项目，统一向 GitHub 仓库提交。实验包 含的 3(+1)个任务分别在不同的目录内开发，具体目录组织方式参见各任务最后 一部分的说明。请务必遵循目录结构，以便于教师/TA 进行测试。

# 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

该任务围绕魔方进行。一方面判断它是否是一个魔方，这个部分的工作又分为两个部分，分别是魔方形式以及元素数据和三种和是否一致。另一部分是运用题目中所给的函数，实现要求的功能。整个过程中对try和catch方法的运用对我来说是一个难点。思路并不难，只需要最少化判断即可。

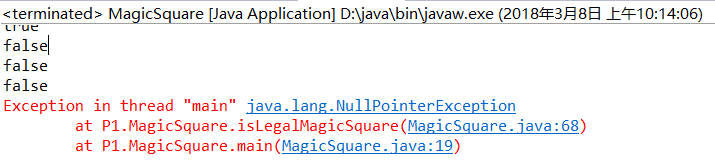
### isLegalMagicSquare()

首先设置基础，读入一行后将行中元素分开并转化为整型，将其放在我们定义的数组中。假设该行的元素个数即为魔方的阶数。此时若遇到一个负数或者偶数则报错。接下来循环直到所有的行都被读入，依次检查行中元素个数是否为我们所假设的阶数。在最后检查行数是否为假设阶数，行数用r实现标记。将读入元素转换为整型检查，遇到错误输入（负数、偶数、浮点数）即以上两种错误就报错。进行到这里时魔方格式基本正确，用同样的方法求其行元素与列元素相加是否为同一个值。之后将两个对角线的元素相加与之前结果比较。若相等则为魔方。若不相等则报错。

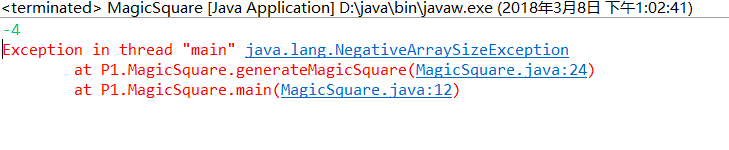
### generateMagicSquare()

该函数根据所给正整数生成一个以其为阶数的魔方。并将其写入文件。当遇到输入不合法时提示错误函数输出false结束。

当n为偶数：



当n为负数：



魔方有着如下的规律：1被填在第一行正中间。2填在最后一行往右一格的位置，向斜上方依次填入下一个数，若斜上方已经有数，就填在当前位置的下一个（实际上只有当前数为n的倍数才会发生这种情况）。若到达边缘，就向另一边延展（仿佛它们是首尾相接卷起来的）（实际上只有第一行和最后一列会发生这种情况）。题目中所给代码就是应用了这个规律的。

## Turtle Graphics

该任务的主要内容是海龟做图，海龟做图曾在python语言中略有了解。核心是forward和turn两个方法。整个过程数学基本知识简单具有趣味性。

### Problem 1: Clone and import

GitHub获取该任务的代码:

在bash里输入git clone 地址；

在本地创建git仓库:

git init；

使用git管理本地开发:

git add;

git commit;

### Problem 3: Turtle graphics and drawSquare

这里需要熟练掌握turtle.forward和turtle.turn两个函数。第一个函数的功能为以海龟朝向的方向画一条直线，直线长度为可设定的参数。第二个函数的功能为转换海龟的朝向，在这里的操作海龟是原地不动的。画一个正方形有八种画法，可以从正方形任意一点开始，分为顺时针旋转和逆时针旋转。在我写的函数中运用了右下角逆时针的方法，所以需要每次在之前的基础上旋转270°。

### Problem 5: Drawing polygons

画正多边形首先需要用之前写过的calculateRegularPolygonAngle函数根据正多边形边数算出每个角的内角度数。之后画上一条基准线，然后依次旋转正多边形内角的补角度数来画剩余的边，这样得到一个逆时针画出的图，只需要画边数减一次即可。

### Problem 6: Calculating headings

这个函数的功能是在上一个函数calculateHeadingToPoint的基础上展开的。calculateHeadingToPoint用当前坐标和目的坐标以及当前海龟朝向来计算出需要直线到达目的点时的调整角。首先设立基准，偏移角（angle）为零度，之后得到需要不断更新的当前偏移角。在这里需要区分的是每一次循环中的当前偏移角和调整角（Degrees)。我们需要从函数中返回的是每次的调整角。而当前偏移角是上一次偏移角加上调整角而来。这里需要注意的是，当前偏移角大于360°时需要我们减去360°，使它的角度保持在零度到三百六十度的范围之内。

### Problem 7: Personal art

我的想法是画一个蓝色的五角星，现在已知五角星的每个角为36°。整个过程和画正多边形类似，首先画一条蓝色的基准线，之后依次旋转36°的补角后画线，重复四次即可得到一个蓝色的五角星。

### Submitting

通过Git提交当前版本到GitHub上你的Lab1仓库:

git remote add origin ... .git

git push -u origin master

## Social Network

对于这道题首先需要构建两个类。分别用于实现一个人的社交关系和整个社交关系网。在之后求得最短路径的过程中需要运用广度优先算法。将一层中的点能一步到达的所有点称之为下一层的点，初始点单独构成第一层。这样找到目的点的层数即为两者的距离。

### 设计/实现FriendshipGraph类

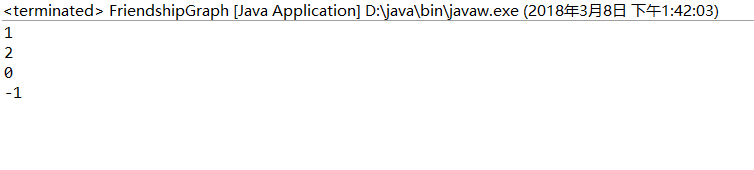
它的数据域为元素是Person类的变长数组，这样就可以完成一张图。构造方法后，用Person类中的加入联系人的方法实现FriendshipGraph类中加入一条边的方法。加入一个点的方法只需要在上述变长数组中加入元素即可。接下来实现求距离方法。以广度优先搜索为基础。一边将本层元素出队，一边将下一层元素入队，直到找出目的点即可，此时目的点所在层数即为两者的最小距离。

### 设计/实现Person类

它的数据域中包含四个部分，首先是用字符串表示的人名，接着将与它直接相连的所有人装在一个动态数组中。接下来的两个在求最短距离用到，一个布尔型标记它是否被访问，整型值判断它所在层数。构造方法用来创建一个新对象，初始化它的姓名。方法用来将一个人的联系人加入到其中。

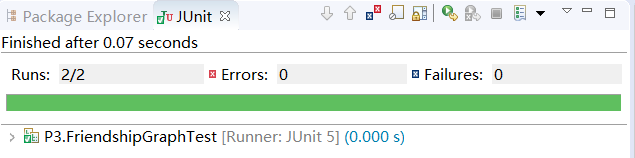
### 设计/实现客户端代码main()

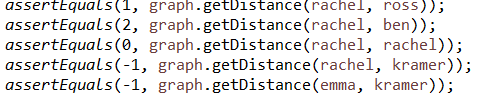
将客户端代码写在FriendshipGraph类中，用题目所给样例测试。



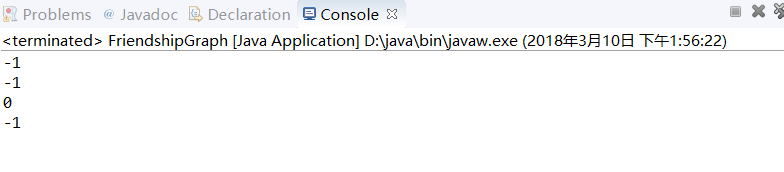
### 设计/实现测试用例

在题目所给样例的基础上扩展，在此基础上加入了一个新的点和关系。

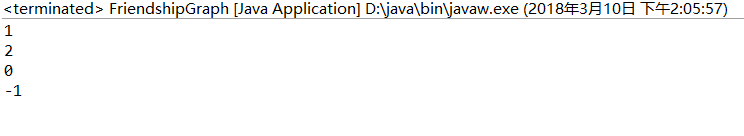


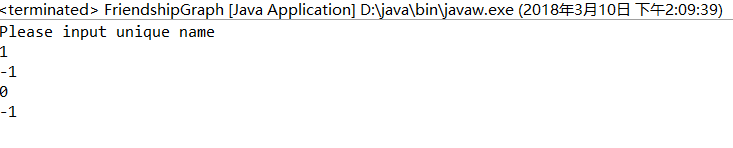


如果将上述代码的第 10 行注释掉（意即 rachel 和 ross 之间只存在单向的 社交关系 ross->rachel），所得结果与我期望的一致。



如果将第 3 行引号中的“Ross”替换为“Rachel”，你的程序会发生什么？



这其实违反了“Each person has a unique name”的约束条件。修改你的 FriendshipGraph 类和 Person 类，使该约束能够始终被满足我改FriendshipGraph类中的一个函数。在检测到相同名字后会报错并且不会把点加入图中但不会结束程序运行。

## Tweet Tweet（选作，额外记分）

自行组织目录结构。

# 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---03-1 | 没课的时候 | 学习基础Java语言、安装所需软件、课程报名等 | 通过看书看视频资料基本完成 |
| 2018-03-2---03-3 | 没课的时候 | 了解环境的配置 | 通过和室友讨论，求助学习群基本完成 |
| 2018-03-3---03-4 | 没课的时候 | 完成第一题 | 通过网上即时搜索与debug基本完成 |
| 2018-03-5---03-6 | 没课的时候 | 完成第二题 | 通过网上即时搜索与debug基本完成 |
| 2018-03-7 | 没课的时候 | 完成第三题 | 通过网上即时搜索与debug基本完成 |
| 2018-03-8 | 没课的时候 | 写实验报告 | 正在进行，即将完成 |
| 2018-03-10 | 全天 | 对实验进行补充完整 | 基本完善 |

# 实验过程中遇到的困难与解决途径

1. 在代码编写的过程中不知道有什么方法可以用来快捷高效地实现功能，通过搜索Java中定义的方法的用途和使用方法，不断修改自己的代码。
2. 实验环境配置能力薄弱，花费了大量时间也不一定正确。通过看网页资料以及和室友讨论求助学习群等解决。

# 实验过程中收获的经验、教训、感想

感受（非必须）：

1. Java编程语言是否对你的口味？

老实说之前没有学过Java语言，加上本身代码能力不强，本次实验做得并不轻松也并不很好。Java语言面向对象与可移植性高的性能非常优越，希望本学期可以借此机会熟练运用Java。

1. 关于Eclipse IDE

Eclipse IDE版面设置合理，报错人性化。虽然路径设置仍然花费了很多时间，但是依然是一个好用高效的工具。

1. 关于Git

对于Git也是初步应用，依然有很多疑惑的地方，希望在以后的实践中更好的掌握这门工具。

1. 关于CMU和MIT的作业

作业有针对性有难点又不至于无法进行，可以说是合理的实验内容了。

1. 关于本实验的工作量、难度、deadline

本次实验安排在学期开头，有大量时间。其工作量、难度、ddl都是合理的。

1. 关于初接触“软件构造”课程

软件构造本身的内容我非常感兴趣，希望可以学懂学好，也希望自己在课外进一步地掌握与运用它们。