

**2019年春季学期  
计算机学院《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 徐通 |
| 学号 | 1170300525 |
| 班号 | 1703005 |
| 电子邮件 | 2676121125@qq.com |
| 手机号码 | 18145648870 |

**目 录**

[1 实验目标概述 1](#_Toc1392798)

[2 实验环境配置 1](#_Toc1392799)

[3 实验过程 1](#_Toc1392800)

[3.1 Magic Squares 1](#_Toc1392801)

[3.1.1 isLegalMagicSquare() 1](#_Toc1392802)

[3.1.2 generateMagicSquare() 1](#_Toc1392803)

[3.2 Turtle Graphics 1](#_Toc1392804)

[3.2.1 Problem 1: Clone and import 2](#_Toc1392805)

[3.2.2 Problem 3: Turtle graphics and drawSquare 2](#_Toc1392806)

[3.2.3 Problem 5: Drawing polygons 2](#_Toc1392807)

[3.2.4 Problem 6: Calculating Bearings 2](#_Toc1392808)

[3.2.5 Problem 7: Convex Hulls 2](#_Toc1392809)

[3.2.6 Problem 8: Personal art 2](#_Toc1392810)

[3.2.7 Submitting 2](#_Toc1392811)

[3.3 Social Network 2](#_Toc1392812)

[3.3.1 设计/实现FriendshipGraph类 2](#_Toc1392813)

[3.3.2 设计/实现Person类 2](#_Toc1392814)

[3.3.3 设计/实现客户端代码main() 2](#_Toc1392815)

[3.3.4 设计/实现测试用例 3](#_Toc1392816)

[3.4 Tweet Tweet 3](#_Toc1392817)

[3.4.1 Problem 1: Extracting data from tweets 3](#_Toc1392818)

[3.4.2 Problem 2: Filtering lists of tweets 3](#_Toc1392819)

[3.4.3 Problem 3: Inferring a social network 3](#_Toc1392820)

[3.4.4 Problem 4: Get smarter 3](#_Toc1392821)

[4 实验进度记录 3](#_Toc1392822)

[5 实验过程中遇到的困难与解决途径 3](#_Toc1392823)

[6 实验过程中收获的经验、教训、感想 4](#_Toc1392824)

[6.1 实验过程中收获的经验和教训 4](#_Toc1392825)

[6.2 针对以下方面的感受 4](#_Toc1392826)

# 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

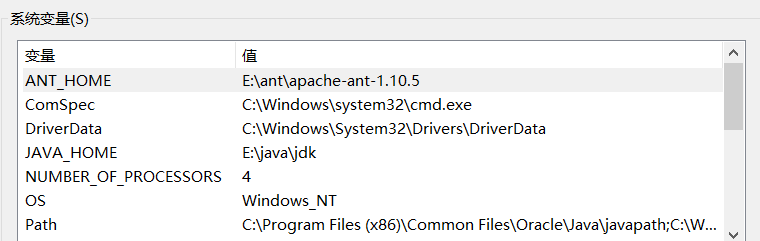
⚫ 基于 Eclipse IDE 进行 Java 编程

⚫ 基于 JUnit 的测试

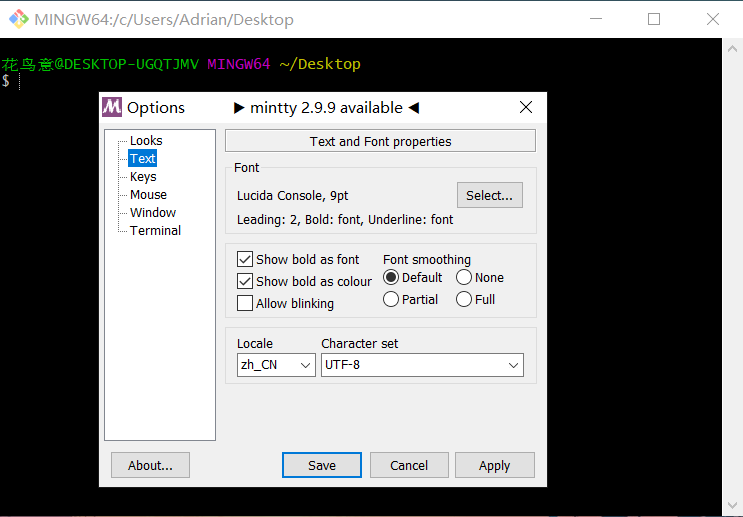
⚫ 基于 Git 的代码配置管理

# 实验环境配置

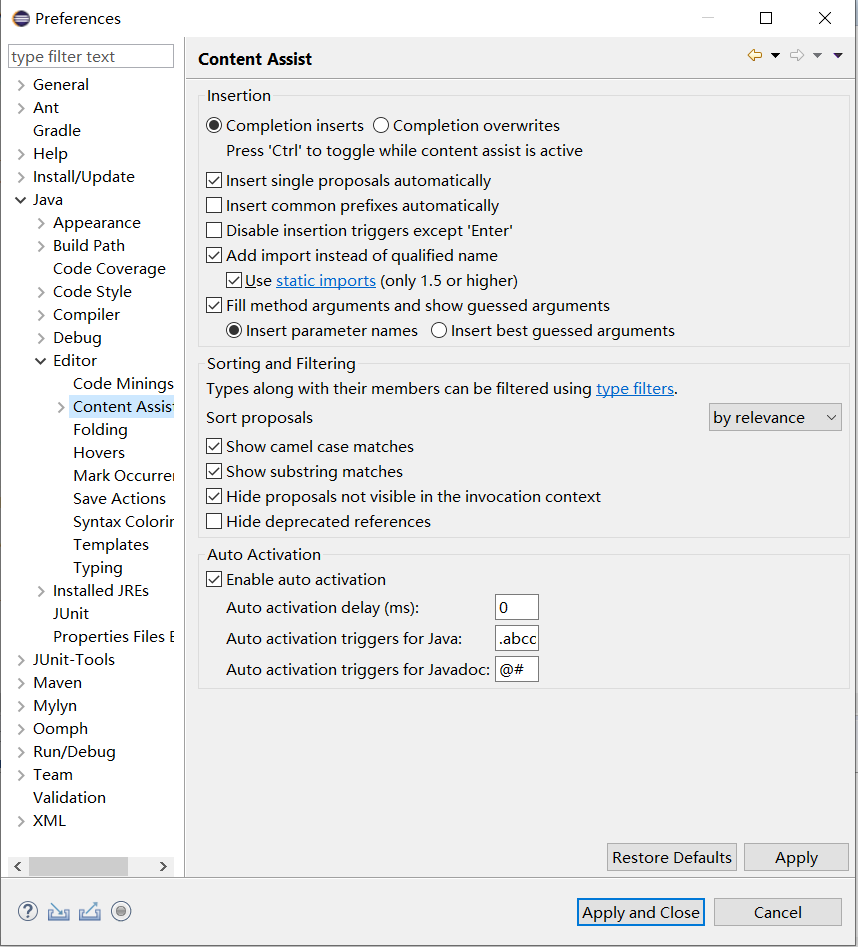
Jdk, jre装好后查询百度配置路径：



Git乱码：

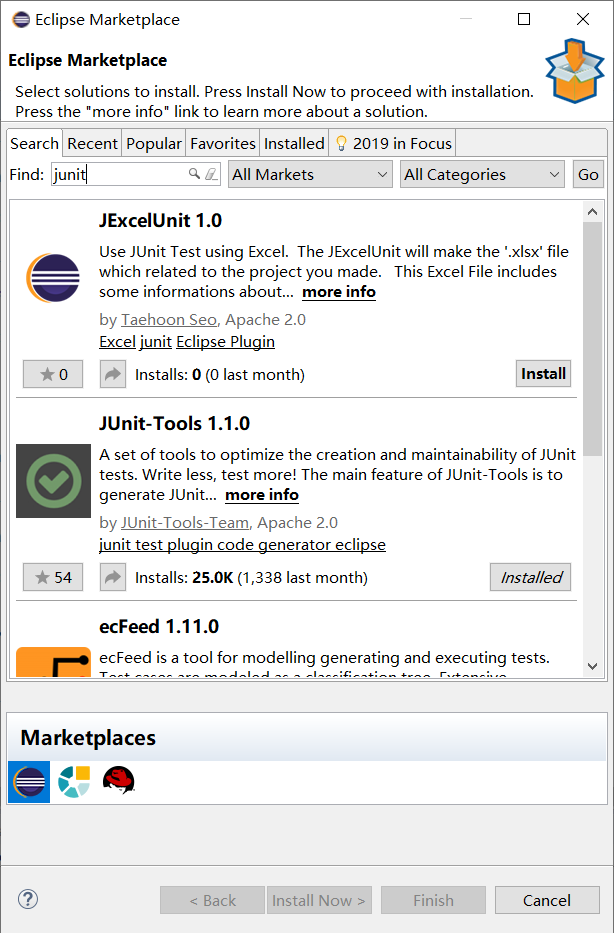


Eclipse配置自动补充：



将.改为.abcdefghijklmnopqrstuvwxyz

Junit下载:



Git查询不到文件（原因：用户文件夹为中文名）：



用另一个管理员用户对该文件夹名进行更改。

GitHub Lab1仓库的URL地址：<https://github.com/ComputerScienceHIT/Lab1-1170300525>

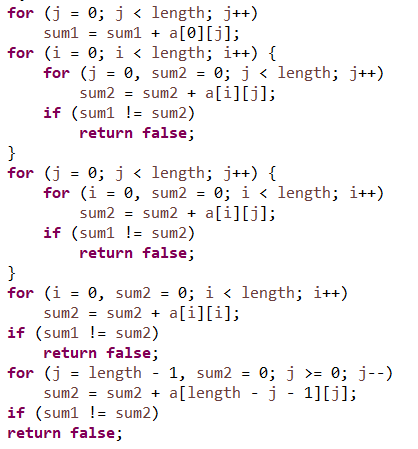
# 实验过程

## Magic Squares

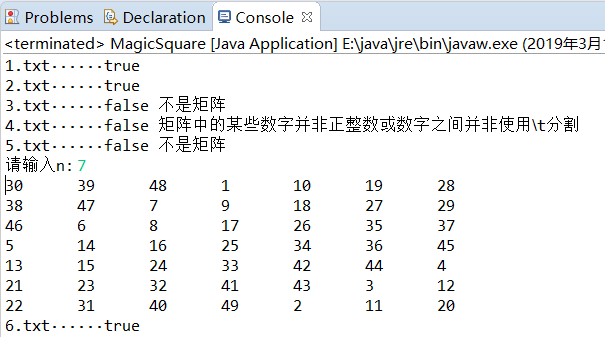
该任务需使用到文件读写，从文件中读取矩阵，根据其行列及对角线各数之和是否相同判断该矩阵是否为magic square。并使我们了解到了一个创造行列数为奇数的magic square的算法。

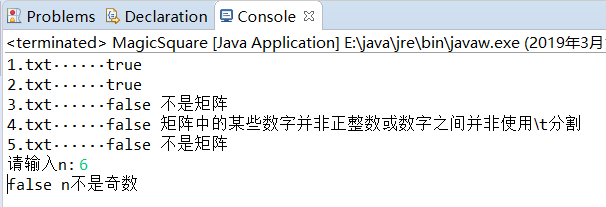
### isLegalMagicSquare()

首先计算出第一行的各数之和，再分别计算其余行，所有列及对角线的各数之和并比较，若全部都相等则称其为一个magic square。其余情况的判断由文件读写时完成传参，交由本函数输出。



结果：





### generateMagicSquare()

该函数代码已给出，需要对其进行调用及评测，该算法大概的思路是：

1. 将第一个数放在第一行正中，将每一个数放在前一个数的右上方；
2. 如果将要放数的格非空，将数放在上一个数的正下方；
3. 如果已经放到顶行，下一次从底行放；
4. 如果已经放到末列，下一次从起始列放，否则就放在上一个数的右上角。

## Turtle Graphics

该任务是对给出的turtlesoup.java进行编写，实现一个简单的用命令控制的绘图工具，从绘制正方形，绘制多边形，计算偏转角，给出一个边界点集，到最终实现自己的个人艺术，层层递进，难度递增。

### Problem 1: Clone and import

从<https://github.com/rainywang/Spring2019_HITCS_SC_Lab1> 获取该任务的代码。

git init

建立本地仓库。

git remote add origin <url>

与github链接

git pull

git add .

. 表示添加全部文件，可以换为对应文件。你已把它们添加到缓存区。

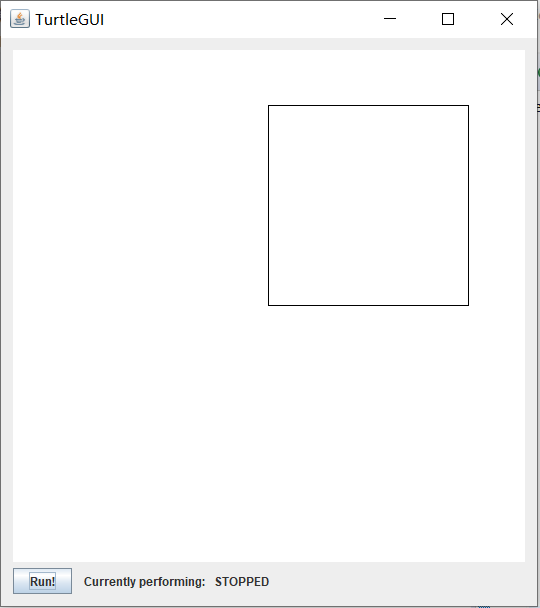
git commit -m “你对此份文件的说明”

git push -u origin master

上传到主分支。

### Problem 3: Turtle graphics and drawSquare

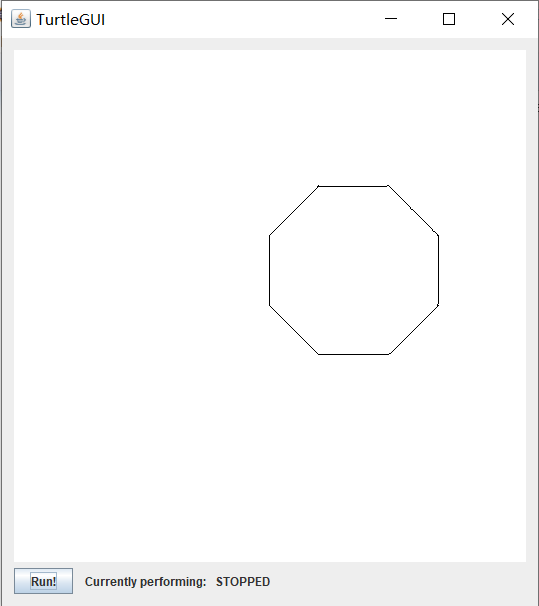
利用自带的转向和移动函数画出一个给定边长的正方形，移动和转向各用四次即可。



### Problem 5: Drawing polygons

利用自带的转向和移动函数画出一个给定边数和边长的多边形，移动和转向各用边数次即可，每次转动的角度由180 - calculateRegularPolygonAngle(sides)（计算正多边形的内角度数的函数）给出。

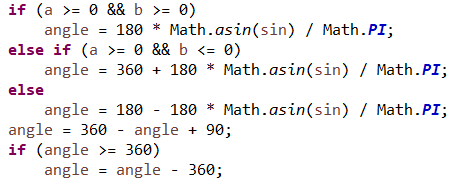
正八边形：



### Problem 6: Calculating Bearings

本函数由一组给定的点计算出每到下一个点需要转的角度，期间需要调用calculateBearingToPoint函数，计算两点之间要转的角度。这更像是一个数学问题，以Y轴正向为参考方向，求出偏转角对其的偏转量，再与本来的方向夹角做差得到真正的角度。

角度计算代码：a,b分别为横纵坐标。



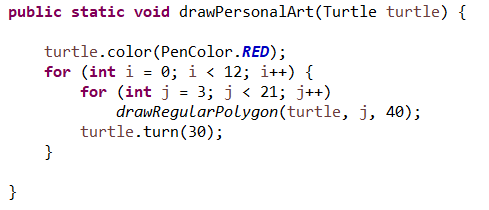
### Problem 7: Convex Hulls

计算该点集我采用了礼品包装算法，每次前往下一个点时，要找到偏转角最小的点并连接。此时只是完成了初步实现，还要考虑多种边界问题，如多点共线，偏转角为0度，一个点也没有或仅有一个点，都要考虑，比较困难。

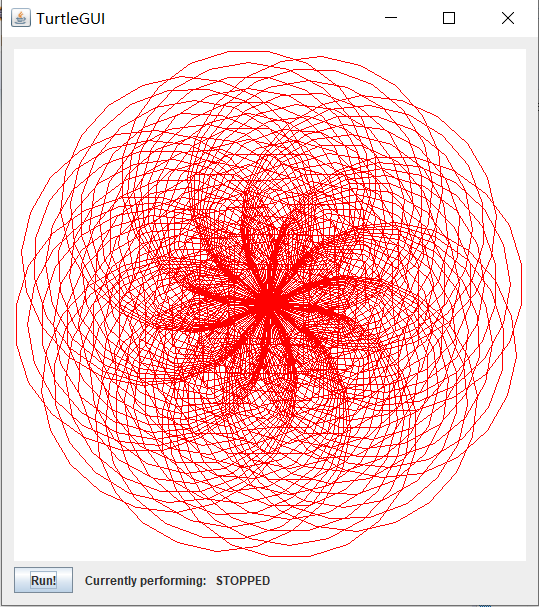
### Problem 8: Personal art

本想画一个代码比较多的图，可是无意中画出了一个不错的图（不停画圈），便不想换了。

代码：



图：



### Submitting

git add P2

git commit -m “TurtleSoup.java”

git push -u origin master

## Social Network

该任务是用邻接表或邻接矩阵建立一个朋友关系图，人名及其关系由用户进行输入，并采用弗洛依德或BFS算法判断两人间的关系为多远，并且本实验需要编写测试用例，来对程序进行测试。

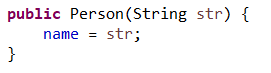
### 设计/实现FriendshipGraph类

在FriendshipGraph类中，主要实现了addVertex(增加人)， addEdge（建立两人的朋友关系），getDistance（得到两人之间的距离）几个函数，涉及到了对图增加顶点，增加边，对图进行BFS搜索。

### 设计/实现Person类

本类最主要是main函数及构造函数的编写。

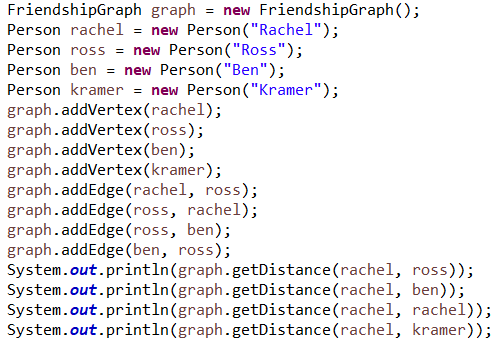
构造函数：



### 设计/实现客户端代码main()

按照题目要求，根据给定的信息进行编写。

代码：



### 设计/实现测试用例

在测试用例中，添加了10个顶点，每次addVertex添加顶点后对顶点个数进行验证，每次addEdge添加边后对边数进行验证，在随机对几组人getDistance后对其最短路径进行验证。

## Tweet Tweet

本任务要实现由多篇tweet推测人物影响力的程序，分模块化构建这个程序，先寻找tweet中@的人（即关注的人），再由多篇tweet确定关注的人的网络，再对这个网络评估，按影响力从高到低对其排序，最终要使你的程序变得更智能，不只局限于谁@谁。

### Problem 1: Extracting data from tweets

在Extract类中，实现了getTimespan（计算一些tweet的时间跨度，即找到日期最早的tweet和日期最晚的tweet），getMentionedUsers（找到tweet的作者@的人，即其关注的人），这个函数中要注意@前面不能跟题目中要求的有效字符，否则就不是一个关注的人。

### Problem 2: Filtering lists of tweets

在Filter类中要实现的几个函数比较简单，分别为writtenBy（找出给定人物所写的tweet） ,inTimespan(找出一个限定时间段之内的tweet) ,containing(找出文中包含给定词的tweet)。

### Problem 3: Inferring a social network

在SocialNetwork类中要实现两个函数。第一个是guessFollowsGraph，要调用Extract类中的getMentionedUsers找到所有发tweet的人他们所关注的人，并建立起一个关注网络。第二个是influencers，需要对第一个函数得到的关注网络进行分析，找到每个人被多少人关注，并得到影响力排行。

### Problem 4: Get smarter

Get smarter是在问题3的基础上对guessFollowsGraph进行改进，使其变得更智能，不只局限于谁@谁。我增加的功能是对RT(转发)的分析，若a关注b，b关注c，c有一篇文章被b转发了，那么a可以看到c的文章，即c对a有一个间接的影响。

# 实验进度记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 2019-03-03 | 18:40-22:00 | 练习文件读写并编写P1的读文件部分 | 按时完成 |
| 2019-03-04 | 15:45-17:30 | 编写P1的两个主要函数 | 按时完成 |
| 2019-03-05 | 19:00-22:00 | 编写P2的drawSquare，calculatePolygonSidesFromAngle，calculateRegularPolygonAngle，drawRegularPolygon函数 | 按时完成 |
| 2019-03-06 | 19:00-22:00 | 编写P2的calculateBearings，calculateBearingToPoint函数 | 按时完成 |
| 2019-03-07 | 13:00-19:00 | 编写P2的convex hulls | 按时完成 |
| 2019-03-09 | 13:00-18:00 | 编写P3及其测试用例 | 延时一小时 |
| 2019-03-11 | 15:45-17:30 | 编写P4的Exact类及Filter类 | 按计划完成 |
| 2019-03-13 | 18:45-23:00 | 编写P4的SocialNetwork类，MyguessFollowsGraph函数及测试用例 | 按计划完成 |
| 2019-03-14 | 10:00-11:35 | 编写P2的person art部分 | 延时30分钟 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的难点 | 解决途径 |
| P2的convex hulls部分输出与测试用例一摸一样可是怎么也过不去 | 经过检查后发现少考虑了一种边界情况 |
| P4的gettimespan函数不是很能理解题目要求，思考错误，无法通过 | 询问同学改题目的意图 |
| 使用git bash时我的用户文件夹为乱码，无法获取其中的文件信息 | 求助百度，创建了一个管理员用户将该文件夹重命名为英文 |

# 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

1.要大力提高自己的英文水平，不然一大部分时间都浪费在了阅读题目和理解题目意思上，过于浪费时间。

2.对于java的掌握不熟练，其中的很多数据结构不是很精通，需要查阅参考书及上网学习。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

提供了很多底层的数据结构，不需要自己重写，少了指针，也少了很多bug。

1. 关于Eclipse IDE

很不错，提供代码的自动补全和纠错，找bug更容易了。

1. 关于Git和GitHub

初次使用git不是很熟练，后来感觉可以掌握。

1. 关于CMU和MIT的作业

选自CMU和MIT的实验相对来说更有趣，同时也更难一些。

1. 关于本实验的工作量、难度、deadline

工作量略大，难度适中，deadline对大部分人来说很合理。

1. 关于初接触“软件构造”课程

实验量略大，基本一直处于写代码的状态。