

**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 | 闫钊 |
| 学号 | 1170300814 |
| 班号 | 1703008 |
| 电子邮件 | 1241390826@qq.com |
| 手机号码 | 18145640397 |

**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

这次实验通过解决求解四个问题，训练基本的Java编程技能，并且利用JavaOO开发基本的功能模块，阅读已有的代码框架并且根据需求补全代码，能够为所开发的代码编写基本的测试程序并完成测试。学会使用GIT。

# 实验环境配置

安装jdk 随意选择目录 只需把默认安装目录 \java 之前的目录修改即可

安装jre→更改→ \java 之前目录和安装 jdk 目录相同即可

安装完JDK后配置环境变量  计算机→属性→高级系统设置→高级→环境变量

系统变量→新建 JAVA\_HOME 变量 。

变量值填写jdk的安装目录

系统变量→寻找 Path 变量→编辑

在变量值最后输入 %JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;

系统变量→新建 CLASSPATH 变量

变量值填写   .;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar（注意最前面有一点）

系统变量配置完毕

URL：https://github.com/1170300814/Lab1-1170300814.git

# 实验过程

## Magic Squares

Magic Squares就是一个矩阵横竖斜都的和都一样就算Magic Squares。这个实验考验了打开文件和处理从文件中获得数据的能力。

### isLegalMagicSquare()

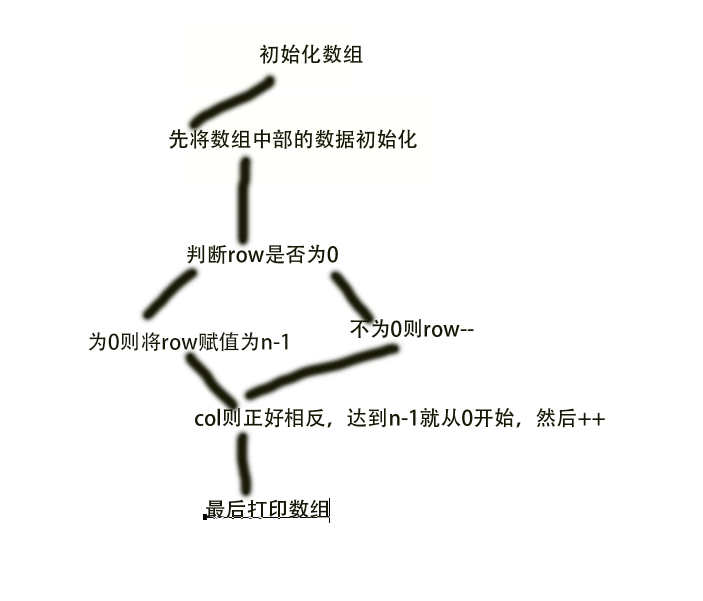
从文件中读取的矩阵，如果有负数或小数，行和列不一样再或者和有一个和不一样就返回false。可以运用求和和判断类型来排除不是幻方的文件。

### generateMagicSquare()

生成一个幻方，然后用上面写的函数判断是否是幻方，只能生成为基数的矩阵，如果输入为负就报错。

如果输入偶数则返回：

生成的文件写入6.txt中。



## Turtle Graphics

完善TurtleSoup.java文件，这个实验是为了训练使用java自带的画图库。

每个方法上都有注释让我们了解这个函数应该完成什么样的功能。再加上MIT给的实验指导很清晰，让我们有了很清晰的方向。

### 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，还有GitHub的客户端进行管理。客户端的功能很全面，能实现克隆代码上传等功能。

### Problem 3: Turtle graphics and drawSquare

**这个方法考察的是基本的turtle的forward和turn函数，实现起来很简单。向前走再转90度，每次转向都是一个方向即可。**

### Problem 5: Drawing polygons

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。

根据数学上的技巧可以解出答案。

### Problem 6: Calculating Bearings

**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。

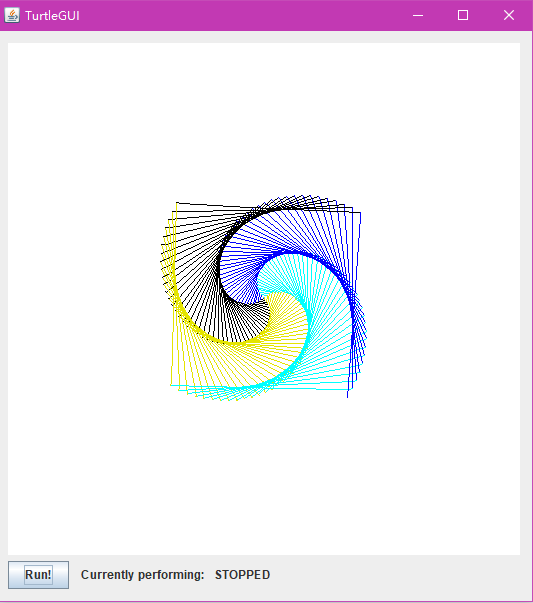
### Problem 7: Convex Hulls

计算凸边，使用递归和分治的方法。

将这些点集分为两个部分。如果点的个数小于两个直接返回，否则根据递归法则将点分成小一点的集合。

### Problem 8: Personal art

我画出了一个渐近线



### Submitting

先用git clone <https://github.com/1170300814/Lab1-1170300814.git将在github>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

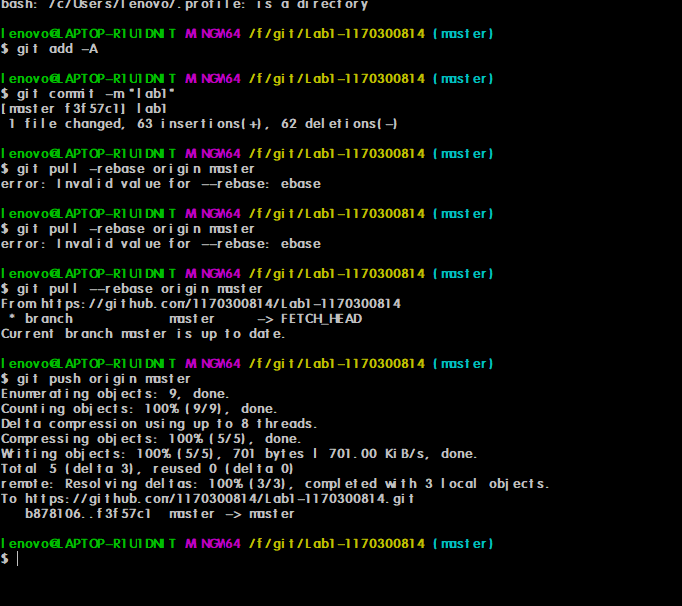
Git add -A

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

该任务是考验我们使用图的功夫，然后建立社交网，判断一个人的社交活动和影响力。

### 设计/实现FriendshipGraph类

这个类是实现了主函数要求的功能。

主要要求的方法有：

adjGraph 是表明这是一个图，统计点的个数

addVertex添加一个人进入图中

addEdge将存在图中的两个人联系起来

getDistance获取两个联系人之间的距离

BFS广度搜索来获取距离

### 设计/实现Person类

Person方法将名字赋给类中的name

linkListAdd在SocialList中添加一个联系人

netWorkSize获取联系人的大小

edgeExist判断边是否存在

getSocial判断联系人的社会位置

### 设计/实现客户端代码main()

public static void main(String[] args)

{

// TODO Auto-generated method stub

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

测试用例涵盖了非法情况，就是考验我们代码的健壮性。

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));

我就用的是实验报告中的实验用例。

## Tweet Tweet

### Problem 1: Extracting data from tweets

获得发tweet的时间和发tweet的人

public static Timespan getTimespan(List<Tweet> tweets)

获取时间间隔

public static Set<String> getMentionedUsers(List<Tweet> tweets)

获取用户，通过截断字符串来实现。

### Problem 2: Filtering lists of tweets

public static List<Tweet> writtenBy(List<Tweet> tweets, String username)

获取特定用户的tweet

public static List<Tweet> inTimespan(List<Tweet> tweets, Timespan timespan)

获取特定时间段发的tweet。

public static List<Tweet> containing(List<Tweet> tweets, List<String> words)

获取包含关键词的tweet

### Problem 3: Inferring a social network

检索用户可能感兴趣的联系人和计算影响力

计算影响力时建立一个联系人和数字对应的哈希表，这样再找出最大的数字即可。这是一个很聪明的方法。

### Problem 4: Get smarter

这个方法中我实现了查找互相关注的人的一个图。

public static Map<String, Set<String>> getSmarter(List<Tweet> tweets) {

String name=null;

String[] word=null;

Map<String, Set<String>> MAP = guessFollowsGraph(tweets);

Map<String, Set<String>> Map = new HashMap<>();

for(Tweet onlyuser: tweets)

{

word=onlyuser.getText().split("[^0-9a-zA-Z-\_@]");

for(String hi:word)

{

if(hi.startsWith("@")&&hi.length()>1)

{

name=hi.substring(1);

for(Tweet user: tweets)

{

if(Map.containsKey(user.getAuthor()))

{

Map.get(user.getAuthor()).add(name);

}

else {

Set<String> tt=new HashSet<>();

tt.add(name);

Map.put(user.getAuthor(),tt);

}

}

}

}

}

for(String username:MAP.keySet())

{

Set<String> followers=new HashSet<String>();

followers=MAP.get(username);

for(String ttt:followers)

{

if(RTMap.containsKey(ttt))

{

MAP.get(username).addAll(Map.get(ttt));

}

}

}

return MAP;

}

# 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 |
| 2019.3.6 | 15:45-17:30 | 编写问题1的isLegalMagicSquare函数并进行测试 | 开始查找资料，开始写P1 |
| 2019.3.10 |  | 写P1 | P1完成 |
| 2019.3.11 |  | 写P2 | P1未完成 |
| 2019.3.13 |  | 写P2 | P2完成 |
| 2019.3.14 |  | 写P3 | P3未完成 |
| 2019.3.15 |  | 写P4 | P3完成，P4未完成 |
| 2019.3.16 |  | 写P4 | P4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 |
| 不太会Java | 通过看Java编程思想和看mooc完成 |
| 一些函数不太理解 | 通过百度来完成 |

# 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。

开始不懂junit测试的时候，按照网上的方法一步一步来结果有一个初始化的问题，查遍全网发现是测试的函数不能带有static字样，修改后才正确，这让我意识到java有一套严格的权限控制体系。

## 针对以下方面的感受

因为以前也安装过类似的IDE，所以并不算困难。初次接触软件构造，希望能学到一些使用的软件编程技巧。