

**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 | 林之浩 |
| 学号 | 1170300817 |
| 班号 | 1703008 |
| 电子邮件 | 630073498@qq.com |
| 手机号码 | 18065053516 |

**目 录**

[1 实验目标概述 1](#_Toc3736323)

[2 实验环境配置 1](#_Toc3736324)

[3 实验过程 1](#_Toc3736325)

[3.1 Magic Squares 1](#_Toc3736326)

[3.1.1 isLegalMagicSquare() 1](#_Toc3736327)

[3.1.2 generateMagicSquare() 3](#_Toc3736328)

[3.2 Turtle Graphics 4](#_Toc3736329)

[3.2.1 Problem 1: Clone and import 4](#_Toc3736330)

[3.2.2 Problem 3: Turtle graphics and drawSquare 5](#_Toc3736331)

[3.2.3 Problem 5: Drawing polygons 6](#_Toc3736332)

[3.2.4 Problem 6: Calculating Bearings 6](#_Toc3736333)

[3.2.5 Problem 7: Convex Hulls 6](#_Toc3736334)

[3.2.6 Problem 8: Personal art 6](#_Toc3736335)

[3.2.7 Submitting 8](#_Toc3736336)

[3.3 Social Network 8](#_Toc3736337)

[3.3.1 设计/实现FriendshipGraph类 9](#_Toc3736338)

[3.3.2 设计/实现Person类 10](#_Toc3736339)

[3.3.3 设计/实现客户端代码main() 10](#_Toc3736340)

[3.3.4 设计/实现测试用例 11](#_Toc3736341)

[3.4 Tweet Tweet 11](#_Toc3736342)

[3.4.1 Problem 1: Extracting data from tweets 11](#_Toc3736343)

[3.4.2 Problem 2: Filtering lists of tweets 12](#_Toc3736344)

[3.4.3 Problem 3: Inferring a social network 13](#_Toc3736345)

[3.4.4 Problem 4: Get smarter 14](#_Toc3736346)

[4 实验进度记录 14](#_Toc3736347)

[5 实验过程中遇到的困难与解决途径 14](#_Toc3736348)

[6 实验过程中收获的经验、教训、感想 15](#_Toc3736349)

[6.1 实验过程中收获的经验和教训 15](#_Toc3736350)

[6.2 针对以下方面的感受 15](#_Toc3736351)

# 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。

# 实验环境配置

在qq群下载了Eclipse和JDK并安装， 在GitHub网站下载安装了Gitbush，分为GUI（图形）和CMD（命令行）两种模式，并简单学习了两种模式的使用。

在配置环境变量的过程中，遇到了环境变量JAVA在电脑关机重启后就会失效的问题，在CSDN上找到了解决方案，参考文章：<https://blog.csdn.net/zhijingmu/article/details/80647174>

把PATH变量分成数行加入后问题解决。

GitHub Lab1仓库的URL地址

<https://github.com/ComputerScienceHIT/Lab1-1170300817>

# 实验过程

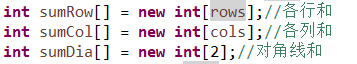
## Magic Squares

首先要了解什么是Magic Squares，Magic Squares也就是幻方是一种将数字安排在正方形格子中，使每行、列和对角线上的数字和都相等的方法。

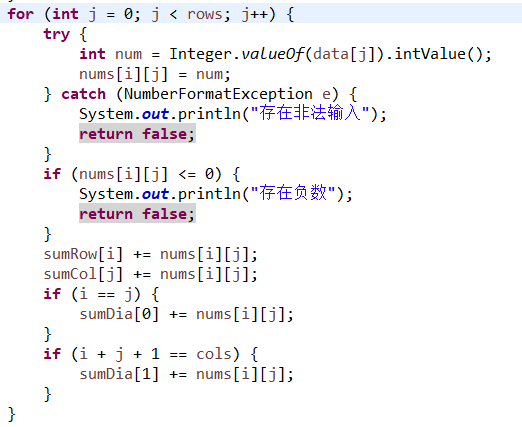
### isLegalMagicSquare()

本题要求写一个函数判断txt文件中保存的矩阵是不是符合幻方要求的矩阵。输入参数为文件路径。返回布尔值。

首先使用传入的文件路径新建一个file对象，再使用FileInputStream流读取文件的内容成一个长string，使用split("\n")方法将长文本按矩阵划分成行保存进一个字符串数组String line[]，再利用split("\t");方法划分成一个个数字（注意行列数不对报错：不是矩阵），随后定义如下变量保存之后的计算过程。

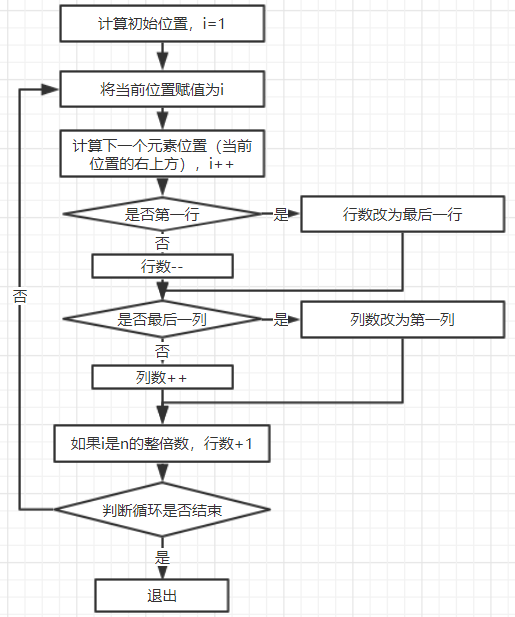


随后将字符换转换成int型整数（注意如果出现不是整数或者非数字报错），保存在二维数组int nums[][]。（出现负数报错）对每个元素遍历求各行和各列之和:

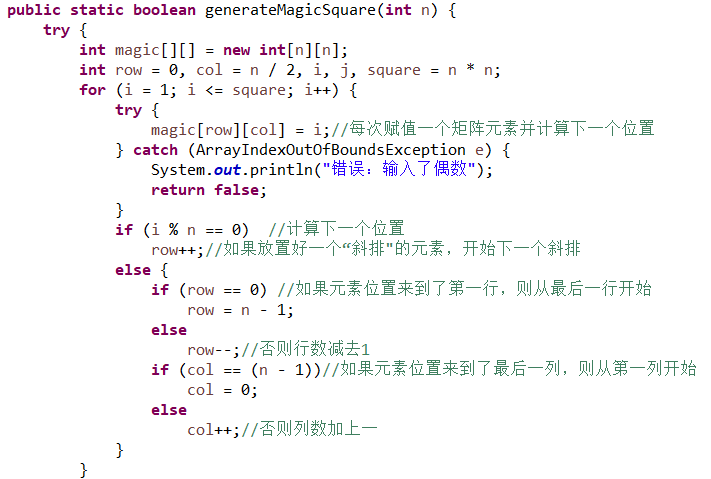


随后判断各行各列之和与对角线是不是都相等，如果是返回true，否则返回false。

### generateMagicSquare()



思路：首先计算初始位置（0，n/2），赋值为1，之后每次取当前位置的右上角的位置，设置的值每次加1，如果当前行是第一行，则下一行为最后一行，如果当前列是最右边的列，则下一次取左边第一列，如此重复平方次，就对整个矩阵赋值了，而且满足了每行每列以及对角线之和都是相同数字的效果。



完成矩阵的设置后输出矩阵成一个txt文件，代码如下：



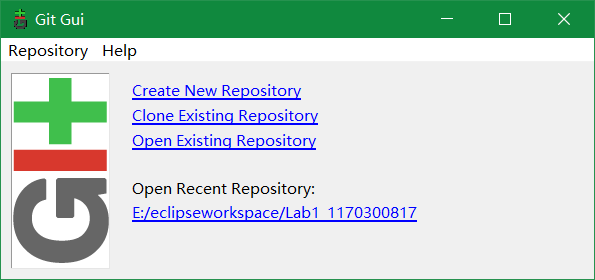
## Turtle Graphics

熟悉turtle Graphics的各种函数接口，调用设置好的函数，实现作图，使用.Math库的几个函数实现一些计算。

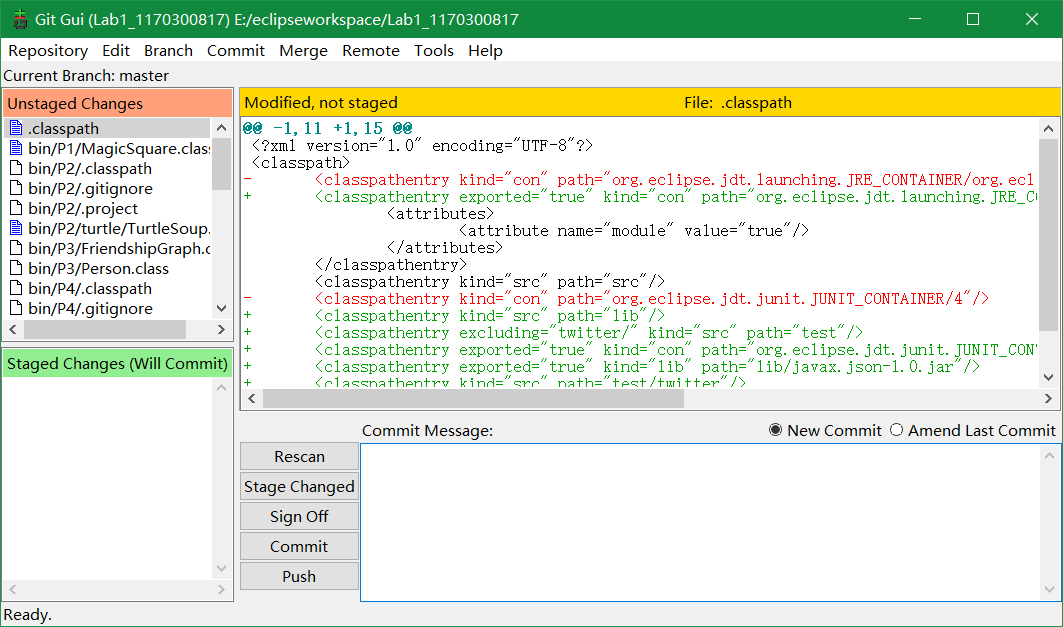
### Problem 1: Clone and import

从教学QQ群下载了原文件，使用eclipse建立相关文件夹后，把文件放到项目中的合适的文件夹中，刷新eclipse，修改包名后使用。

进入project文件夹，右键GIT GUI HERE，



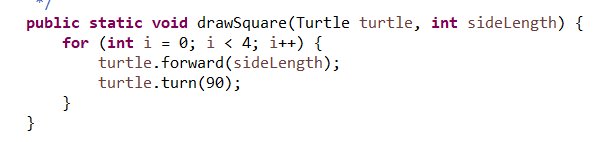
点击create new repository ，选择好目录后点击创建



就完成了。

### Problem 3: Turtle graphics and drawSquare

很简单，重复四次前进转弯90就行了。

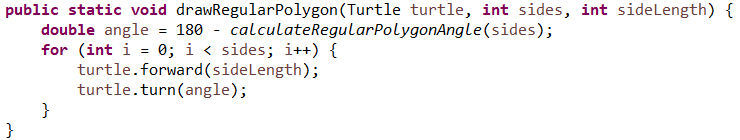


### Problem 5: Drawing polygons

利用以下代码编写calculateRegularPolygonAngle (int sides)计算n边形的内角：

(sides - 2) \* (180.0/ sides);

随后重复sides次前进转向即可



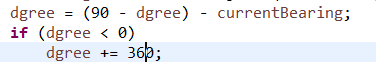
### Problem 6: Calculating Bearings

利用atan2函数计算前进方向与x轴正向夹角，



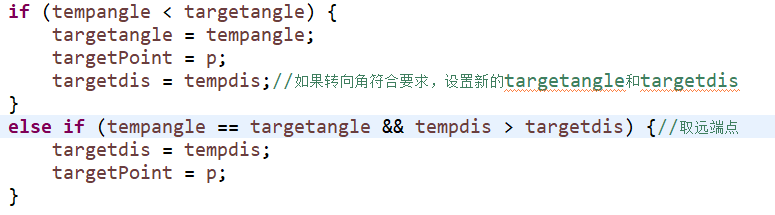
之后再把该角度转换成和y正向的夹角，

注意如果出现负数要加上360度



### Problem 7: Convex Hulls

运用Gift wrapping algorithm算法，每次选择转向角最小的点（同时距离最长的点），加入集合，求转向角函数直接使用的之前写好的calculateBearingToPoint函数，

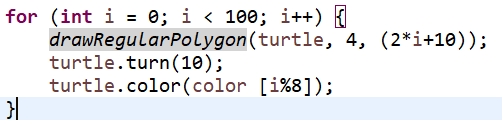


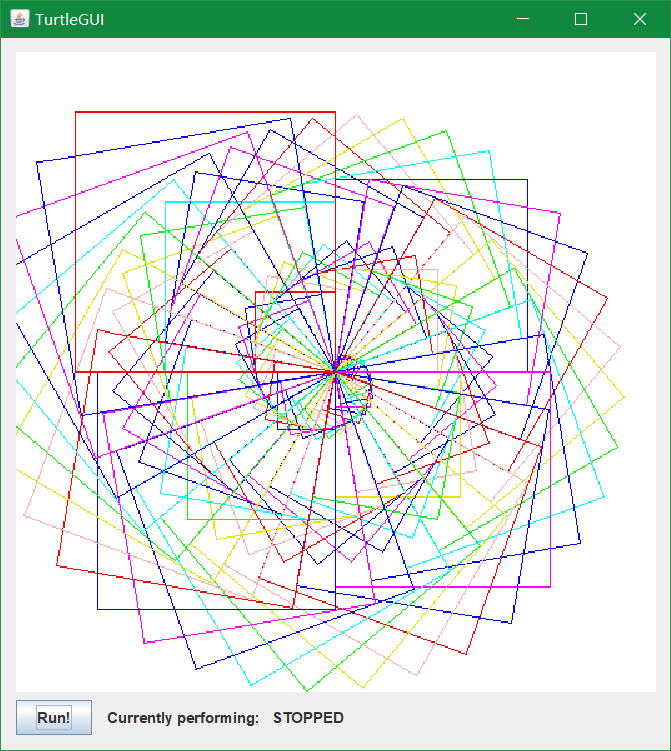
其中注意点是相同转向角额点的取舍，在循环中设置一个targetdis变量记录当前目标点的距离，如果之后出现了新的目标点，就用targetdis和计算得到的当前点的距离tempdis来比较进行取舍。

### Problem 8: Personal art

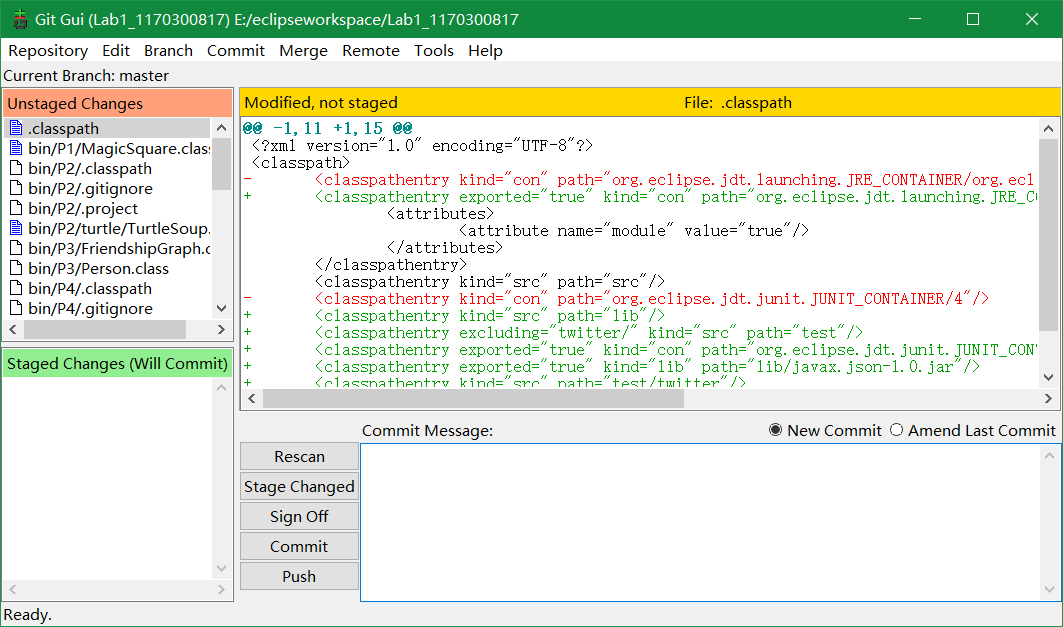
主要想法是把之前编写的drawRegularPolygon（画多边形）函数，和改变颜色的功能利用起来，所以循环：画四边形，转换颜色，边长增加，转一个小角度。

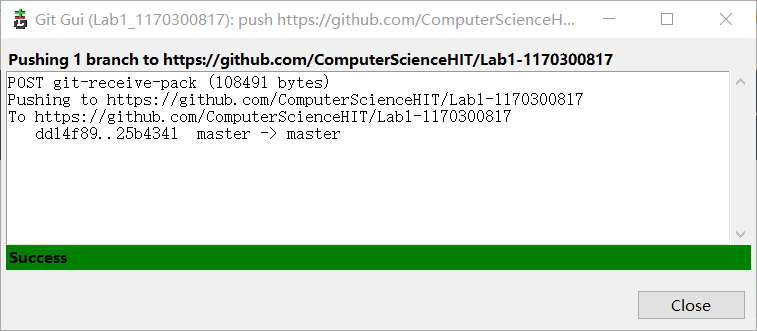
其中color数组保存了已经定义的各个颜色。





### Submitting

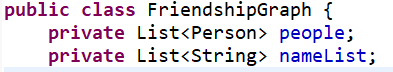
依次点击点击rescan ，stage changed 输入commit message在右侧方框之后点击commit，之后点击push，输入仓库地址，就成功了



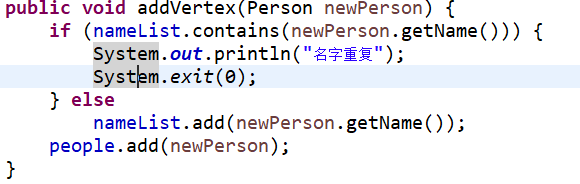
## Social Network

利用各种数据结构，实现一张人际关系网络。写一个计算人际关系“距离”的函数。

### 设计/实现FriendshipGraph类

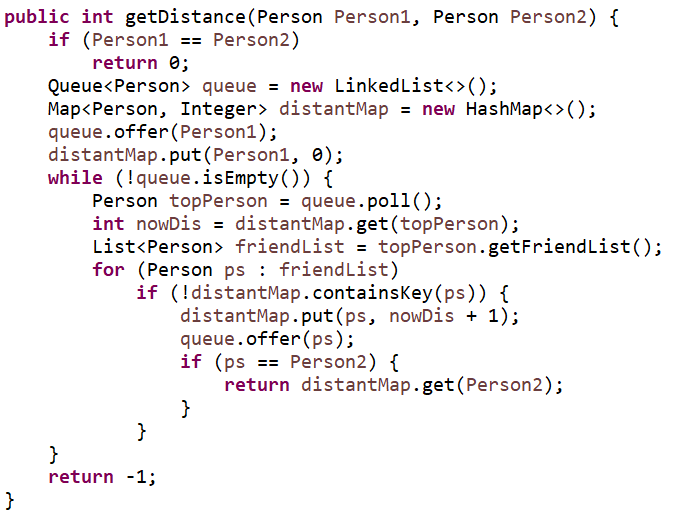


在FriendshipGraph类中声明两个list：people为person类型，用来保存人的实例，另写一个namelist为string类型，我用来检测是否有重复名字的，这样避免使用people列表来查找是否有重复的人，较为麻烦。



addVertex(Person newPerson)函数用来加入新的“人”到people列表，再次之前要在namelist中检查是否有重复的名字，有则输出提示。

addEdge(Person pa, Person pb)函数调用people类的addFriend方法，在每个人的朋友列表中增加新的朋友。



getDistance(Person Person1, Person Person2)函数为难点，声明一个Map<Person, Integer> distantMap数据结构，来保存起点到当前点的键值对（名字，距离），使用邻接表广度搜索的方法，借助队列，先将起点入队，然后执行循环，直到队列为空前：弹出队列头元素，计算当前距离，把弹出点的所有“朋友”入队，并加入map中，设置距离为先前计算的距离+1。直到找到结束的节点。返回这个点在map中键值对的“值”。

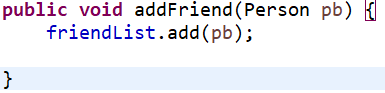
### 设计/实现Person类

Person类：

包含保存名字的字符串private String name

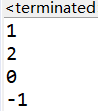
保存朋友的字符串private List<Person> friendList

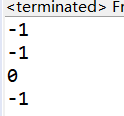
和他们相应的get方法getName()和getFriendList()



addFriend方法为对象的朋友列表新增一个朋友项。

### 设计/实现客户端代码main()

如果不修改提供的main代码，结果为。

如果把graph.addEdge(rachel, ross)，注释掉，结果为。原因是：rachel，到ross的边不存在了，所以rachel,和ross距离-1，其他点也无法达到。只有getDistance(rachel, rachel)返回0。

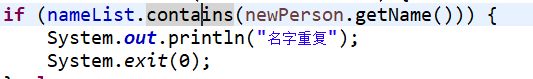
再次修改main（）试图输入重复姓名



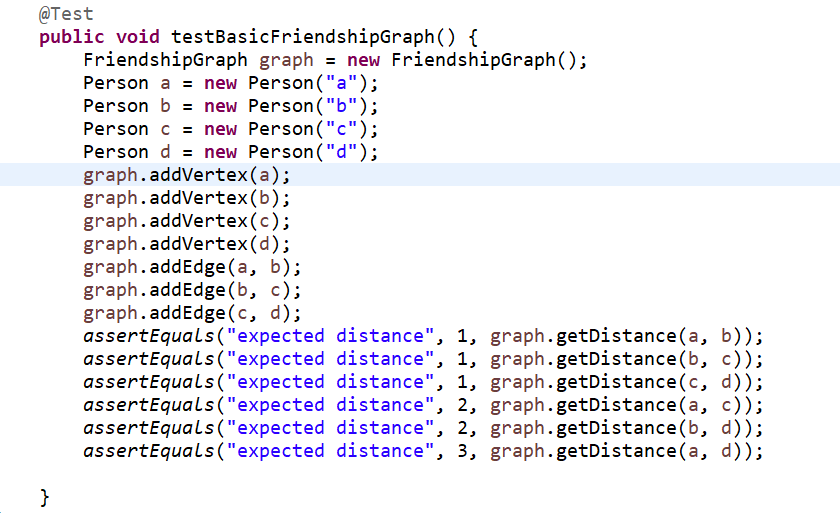
则输出



原因是之前就设置了查找到重复名字就退出的代码：



### 设计/实现测试用例



简单设计一个test检测各种距离和不可到达的情况，同名的情况根据要求会直接退出，就没写对应test。

。

## Tweet Tweet

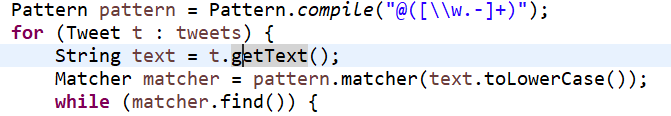
### Problem 1: Extracting data from tweets

getTimespan(List<Tweet> tweets)返回list中的推特覆盖的时间段，先设置好初始起止点，

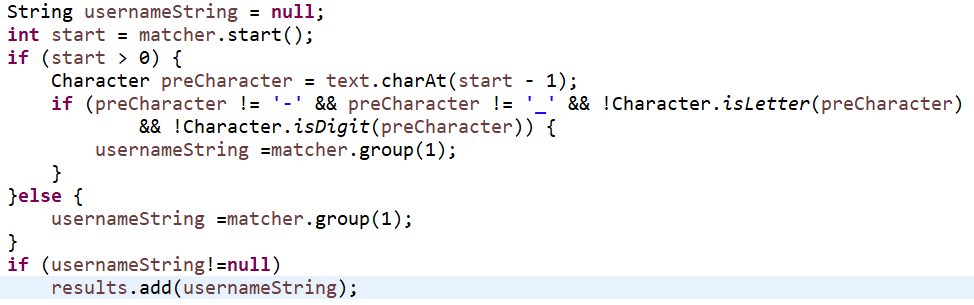


然后对每条推特，如果时间点在当前时间区间之外，就修改时间区间的上或者下界。

Set<String> getMentionedUsers(List<Tweet> tweets)函数根据tweet列表返回tweet中提到的用户名，具体做法是先用Pattern.compile("@([\\w.-]+)");方法匹配满足条件的字符串，



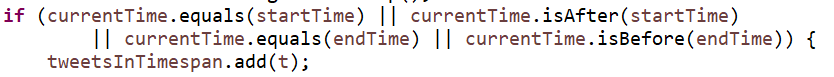
然后对找到的位置进行检查，如果前面不包含合法字符，则满足要求，或者前面没有字符（即开头）也满足要求，把符合条件的加入返回



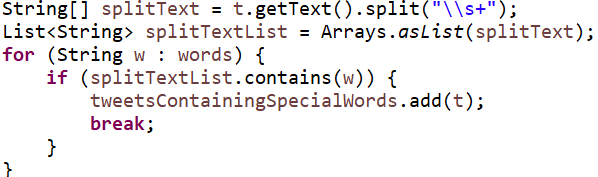
### Problem 2: Filtering lists of tweets

writtenBy(List<Tweet> tweets, String username)函数返回作者是输入的username的tweet的列表，只要对传入的list中的每个tweet调用getAuthor方法获得作者之后进行比对，如果（无视大小写）是一样的，就把当前的tweet加入返回的list。

inTimespan(List<Tweet> tweets, Timespan timespan)函数返回list中时间在传入的区间内的所有tweet，只需对所有tweet获取时间后比对，如果符合就加入返回的list。判断条件如下：



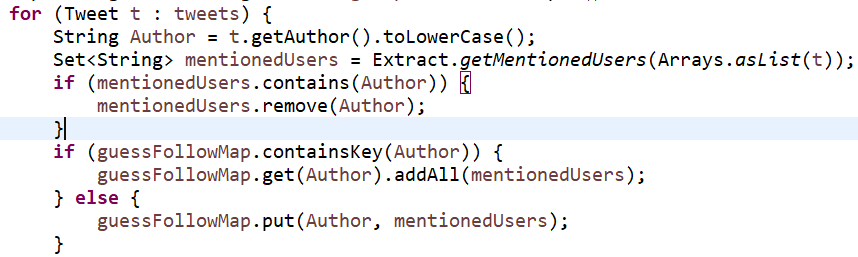
containing(List<Tweet> tweets, List<String> words)该函数输入多个关键词和一个tweet list，返回包含关键词的所有tweet列表组成的list。我的做法是先把每一个tweet分割成单词，因为数组没有contains方法，所以把数组转成list后，调用contains方法，检查是否包含关键词,有的话把当前检查的tweet加入返回的列表。



### Problem 3: Inferring a social network

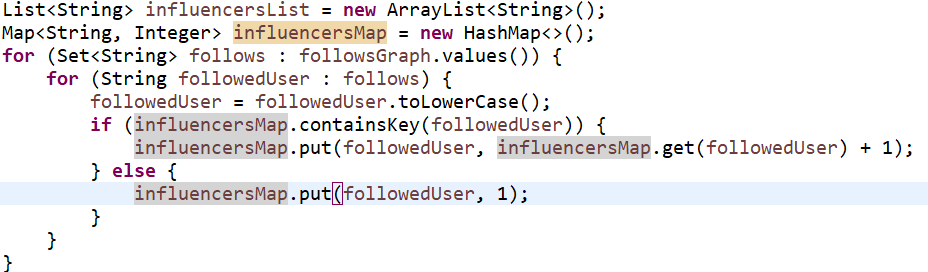
Map<String, Set<String>> guessFollowsGraph(List<Tweet> tweets)

思路：对每个tweet取作者Author和提到的人（调用getMentionedUsers）mentionedUsers，然后在map中查找是否有Author这个键，如果已经存在，把mentionedUsers列表添加进去，如果不存在Author这个键，就新建此键值对（Author，mentionedUsers）



List<String> influencers(Map<String, Set<String>> followsGraph)函数根据传入的Map对每个人按照影响力大小排序，

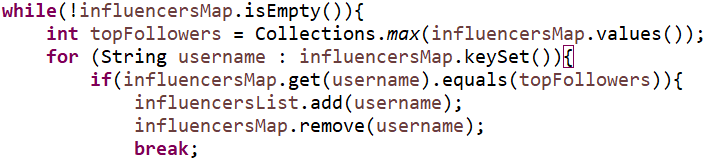
首先声明Map<String, Integer> influencersMap数据结构保存（人名，影响力），



对参数map中的每个string数组中的每个人，检查：如果此人已经在influencersMap中，则将对应的值+1，如果不在，则放置新的键值对（人名，1）。

因为会出现有的人没有被别人follow的情况，所以要把剩余的人加入map设置值为0。

随后按value的大小顺序取出map的元素依次加入list



### Problem 4: Get smarter

采用建议的第一种策略：修改guessFollowsGraph函数，提取twet中的hashtag，策略是用一个map数据类型保存每个人的名字，和他发过的所有hashtag，



之后再进行判断，对于每两个用户进行比对：如果此二人发过的hashtag集合的交集的数目大于3个，则认为此二人互相影响：



# 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。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 2019-02-25 | 15:45-17:30 | 学习git | 按计划完成 |
| 2019-02-25 | 18:50-22:00 | 编写isLegalMagicSquare和测试 | 按计划完成 |
| 2019-02-26 | 19:00-22:00 | 完成P2的1-6 | 按计划完成 |
| 2019-02-27 | 19:00-22:00 | 完成P2的convexHull和个人艺术 | 按计划完成 |
| 2019-02-28 | 15:00-22:00 | 完成P3 | 按计划完成 |
| 2019-03-3 | 19:00-22:00 | 完成P4的任务1.2 | 按计划完成 |
| 2019-03-6 | 19:00-22:00 | 完成P4的任务3 | 按计划完成 |
| 2019-03-7 | 19:00-22:00 | 完成P4的任务4 | 写完了测试始终不通过 |
| 2019-03-8 | 19:00-22:00 | 修改了P4实现，补充测试 | 按计划完成 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的难点 | 解决途径 |
| Git不会使用， | 上网查找Git教程  访问教程“廖雪峰的官方网站” |
| 实验导入问题，目录结构不知道怎么建立 | 咨询了有丰富经验的同学，理清了什么时候要建立folder什么时候要建立package。 |
|  |  |

# 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

对java语言的掌握十分不足，下一次实验前需要加强。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

对，具体就是很多数据结构直接可以用，上学期c语言数据结构手写很酸爽。

1. 关于Eclipse IDE

目录有时不灵敏。。

1. 关于Git和GitHub

需要继续学习才能熟练使用

1. 关于CMU和MIT的作业

能够向世界一流计算机专业学习是迈向世界一流必经之路

1. 关于本实验的工作量、难度、deadline

工作量大，难度不小，deadline要看同时期其他课程的安排。挤在一起就很累。

1. 关于初接触“软件构造”课程

代码量大。