

**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 | 夏铭远 |
| 学号 | 1173710209 |
| 班号 | 1737102 |
| 电子邮件 | 462272701@qq.com |
| 手机号码 | 13029878656 |

**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

通过求解4个问题，训练基本Java编程技能，能利用JavaOO开发基本的功能模块，能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的基本用法。

# 实验环境配置

Java和eclipse之前已经安装过了，直接使用。

从实验包中获取一些代码和jar包。

安装并初步学习了Git。

绑定了远程仓库。

<https://github.com/ComputerScienceHIT/Lab1-1173710209>

本来想通过eclipse中的Git进行版本管理操作，但出了一些问题并没有好的解决。

# 实验过程

Magic Squares

该任务第一部分要求从文件中读取矩阵，判断是否满足所给的条件：行，列，对角线之和等于同一个数，每个数必须是整数，且都用\t分隔。主要训练Java的文件操作，字符串处理，相应的计算方法等编程技能。

第二部分给出了一个产生MagicSquare的方法，让我们对其进行研究并完善。主要训练了代码阅读能力和加强健壮性的技能。

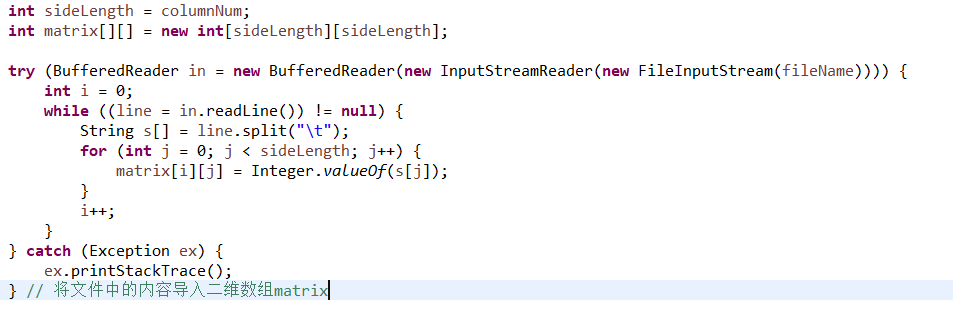
### isLegalMagicSquare()

1. 检测读入的文件中是否是一个格式正确的方阵。
   1. 首先从文件中按行读取，用split将整行字符串按 **’\t’** 分隔，得到字符串数组s，将这个s的长度的值赋给列数变量columnNum，再通过ascii码判别s中每个字符串是否是纯数字（这样可以直接屏蔽掉空格），如果不是直接返回false。
   2. 确认都为整数后，判断行列数是否相等
   3. 判断是否为空



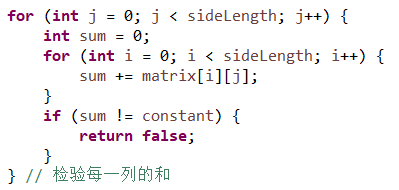
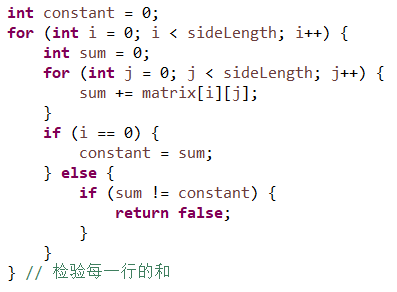
1. 确认是方阵后，将内容导入一个二维数组matrix[][]。

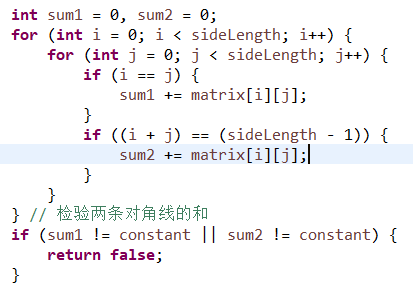
直接按行分隔然后遍历。



1. 检验行，列，对角线的和。

（定义constant为那个定值，后面都与其比较。）





最后全部筛选后返回true。

### generateMagicSquare()

通过罗伯法生成幻方。

数字1放在首行最中间的格子中

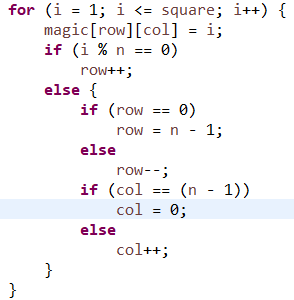
向右上角斜行，依次填入数字

如果右上方向出了上边界，就以出框后的虚拟方格位置为基准，将数字竖直降落至底行对应的格子中

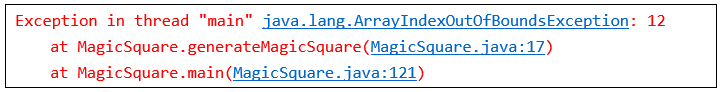
同上，向右出了边界，就以出框后的虚拟方格位置为基准，将数字平移至最左列对应的格子中

如果数字n右上的格子已被其它数字占领，就将n+1 填写在n下面的格子中

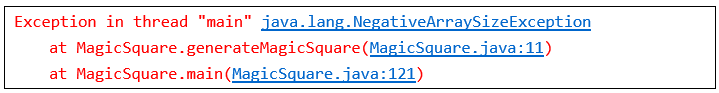
如果朝右上角出界，和上面重复的情况做同样处理



解释两个异常

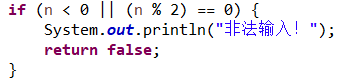


该异常为数组越界，若输入的是偶数，则当遍历到最下一行时，i%n=0，此时row++，到下一次循环时就会越界。

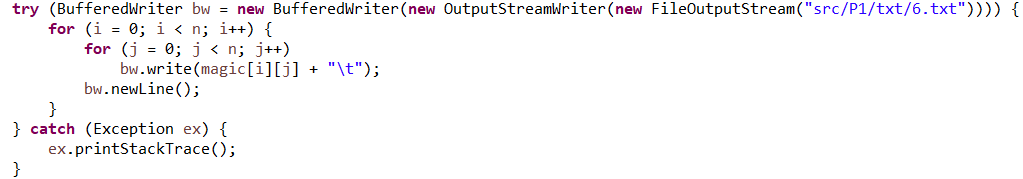


该异常为数组长度为负，数组的长度必须是正整数，传入负值显然会有异常。

按照题目要求，再遇到这些情况时优雅退出程序。



将生成的幻方写入文件



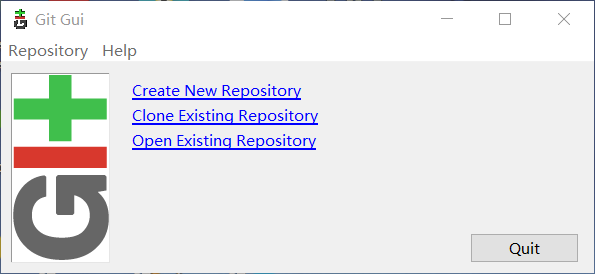
## Turtle Graphics

该任务提供了turtle这个API，用它的方法实现一系列函数。包括绘图，计算角度，边数以及凸包求解算法。同时这个任务让我们第一次接触JUnit，入门软件测试。

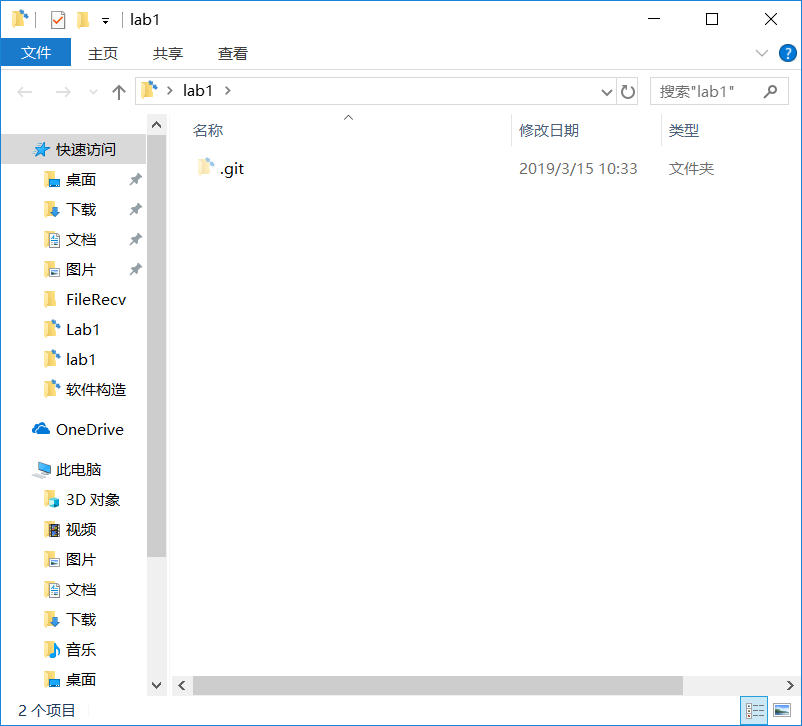
### Problem 1: Clone and import

打开老师给的网址，下载代码，拷贝到eclipse项目中

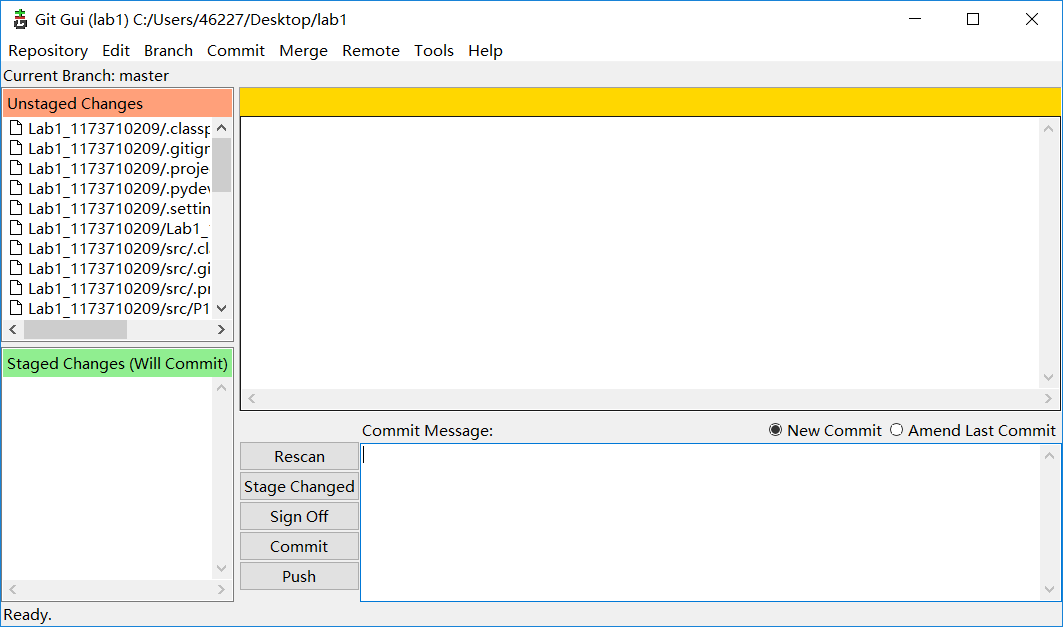
桌面右击选择Git GUI here



Create New Repository --> 选择目录 --> 创建成功

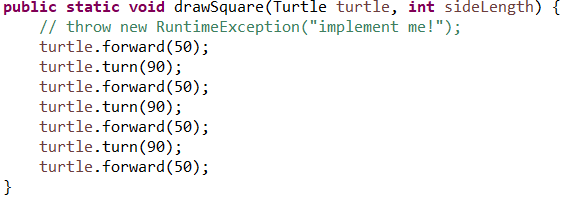


将eclipse项目拷贝到当前目录，就可以用Git管理开发了



### Problem 3: Turtle graphics and drawSquare

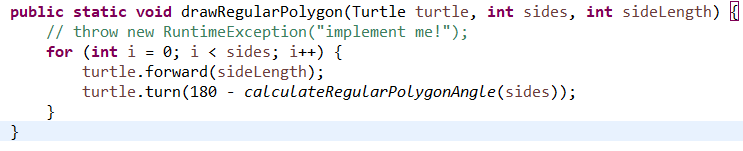
该问题使用forward和turn绘制正方形，很简单。



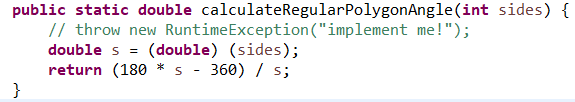
### Problem 5: Drawing polygons

该问题给定边数和边长绘制多边形。

实现此功能需调用calculateRegularPolygonAngle计算出每次需转过的角度，之后循环前进边长长度即可。

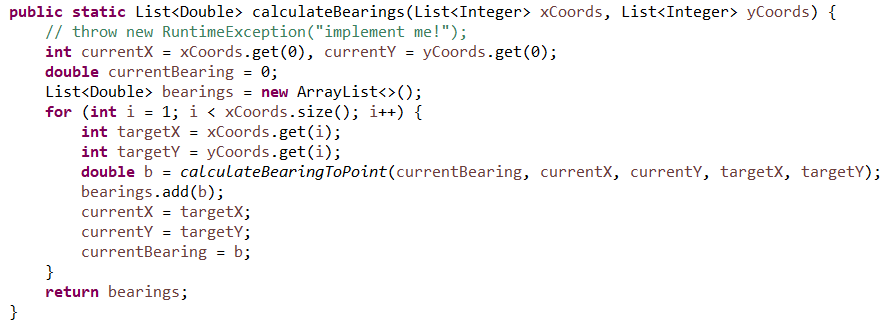


其中calculateRegularPolygonAngle计算角度方法：

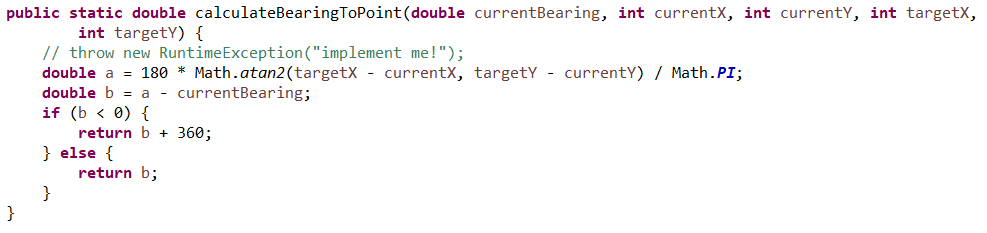


### Problem 6: Calculating Bearings

该问题计算每两点间的偏转角，直接循环调用calculateBearingToPoint即可，每次调用后的值加入返回结果列表中，再对下次循环的参数重新赋值。循环结束后返回列表bearings。



其中calculateBearingToPoint，使用atan2计算偏转角，并且需要分类讨论（当前的左还是右，这里代码用b的正负表示）



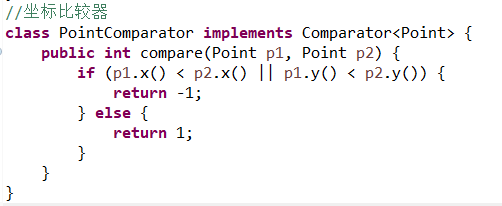
### Problem 7: Convex Hulls

该问题就是凸包求解算法gift-wrapping。

按照turtle的顺时针转向，设起点为点集中最左最下的点，然后开始遍历其它点并选出偏转角最小的作为下一个目标并加进结果集合，为了防止在同一条线上，当偏转角为0时，就要从结果集合中删除当前点。

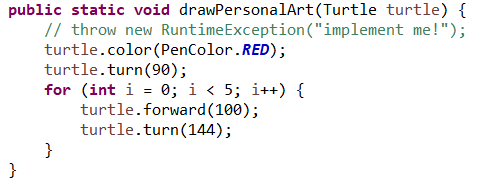


筛选最左最下点的比较器：



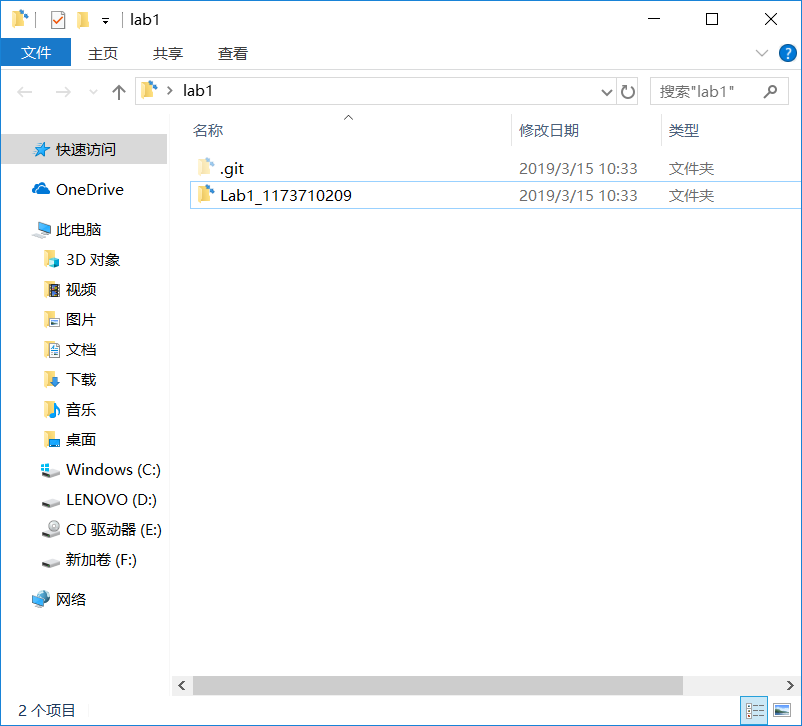
### Problem 8: Personal art

该问题随意作图，由于脑洞不大，只画了个红五角星。

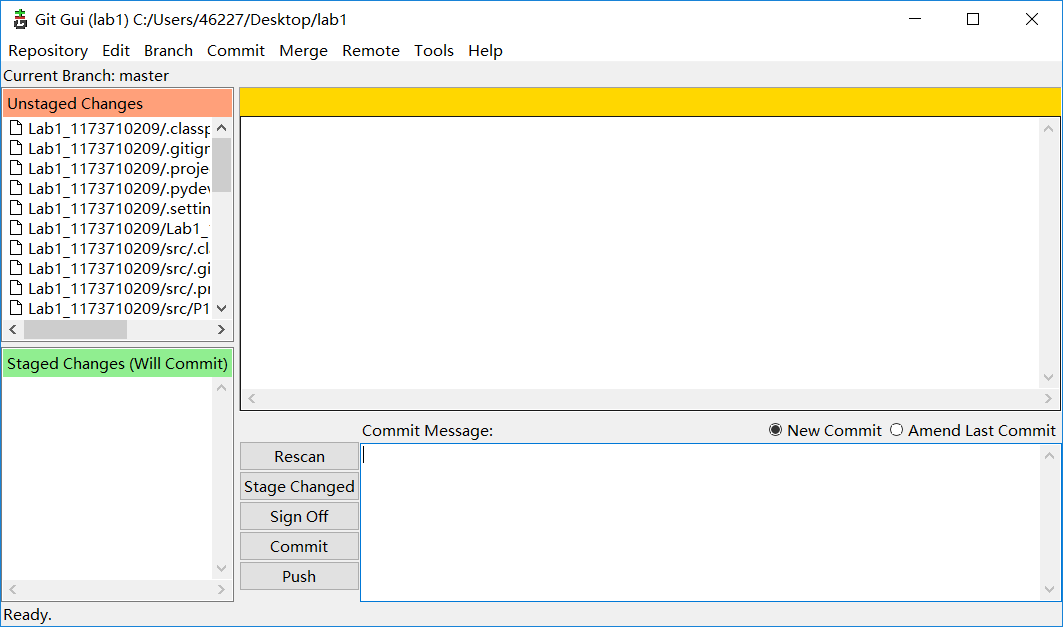


### Submitting

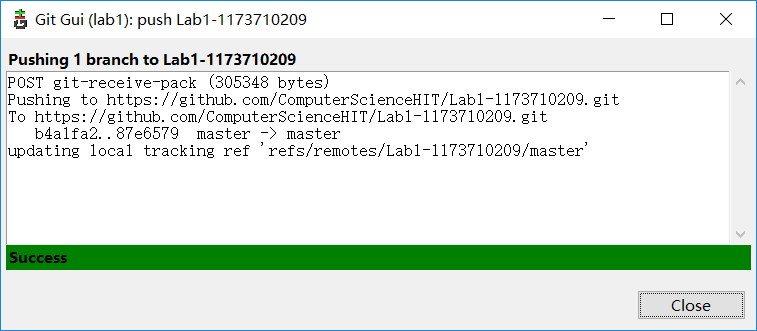
把项目文件夹拷贝到本地仓库

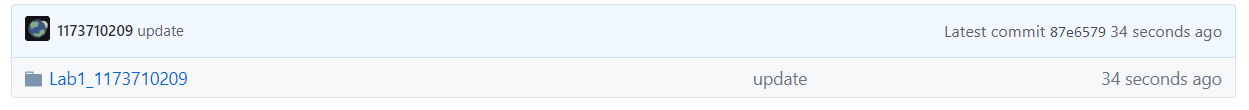


打开GIT GUI



点击Stage Changed 🡪 输入Message 🡪 Commit 🡪 Push 就完成了





## Social Network

该问题主要考查图模型相关算法，使用BFS计算无权图单源最短路径，设计Person类，并自己编写测试用例进行测试。

### 设计/实现FriendshipGraph类

一个属性，5个方法。

ArrayList<Person> vertex = **new** ArrayList<>();点的列表

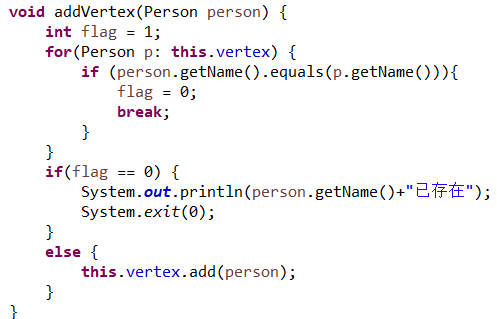
Main函数

**void** addVertex(Person person)用于添加点

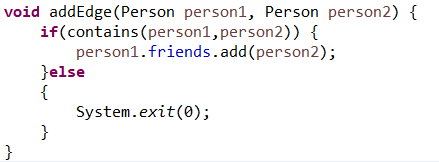
**void** addEdge(Person person1, Person person2)用于添加边

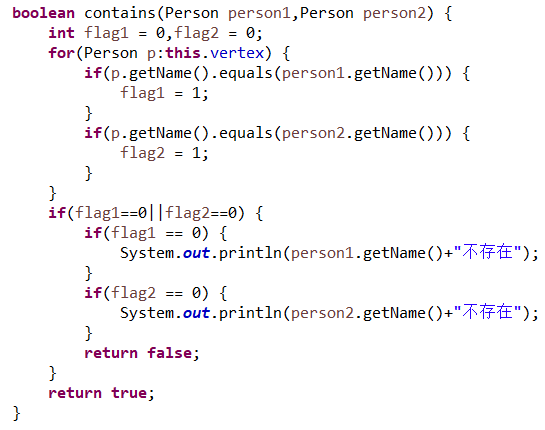
**int** getDistance(Person person1, Person person2)用于计算距离

**boolean** contains(Person person1,Person person2)用于判断两个参数对象是否存在于vertex

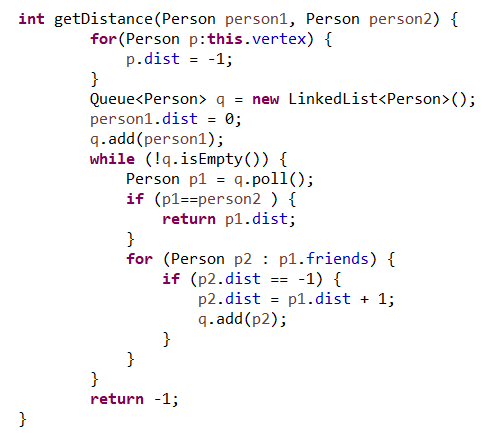


设标志变量判断是否已存在，之后用add添加





之后是求最短路径算法，这里采用BFS。每次调用时先将所有的dist初始化为-1，创建队列q，把起点person1的dist赋值为0（自己到自己的距离），person1进队。随后当队列q非空时，弹出队首传递给p1，如果p1正好是person2则返回p1.dist，其余情况则遍历p1的friends，并在p1的基础上加1点距离，同时进队。这样循环下去直到q是空集或者p1=person2。



### 设计/实现Person类

Person类包含三个属性，一个构造方法和一个getName()方法。

属性：String name;姓名

ArrayList<Person> friends;和该对象邻接的对象

int dist;该对象到原点的距离（计算最短路径时用）

构造方法：

**public** Person(String name) {

**this**.name = name;

**this**.friends = **new** ArrayList<>();

**this**.dist = -1;

}

### 设计/实现客户端代码main()

用户在控制台输入姓名列表，手动添加朋友关系，查看朋友关系。



### 设计/实现测试用例

给出你的设计和实现思路/过程/结果。

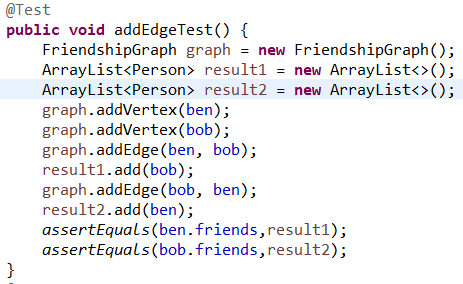
1. 测试addVertex

主要测试1个，多个，和重复的情况



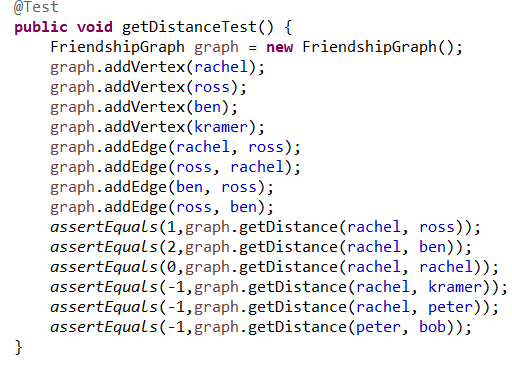
1. 测试getEdge

主要测试1个，多个，双向



1. 测试getDistance

主要测试不邻接，距离为1，距离大于1，孤立点等情况



## Tweet Tweet

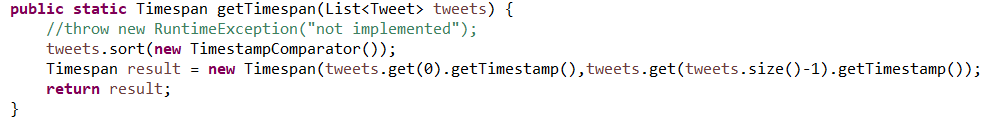
该任务从推特入手，完成各种词语处理，人际关系网络等模块。重在体会社交系统的各项功能的实现。

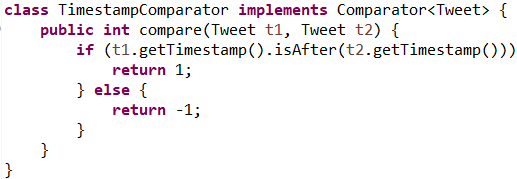
### Problem 1: Extracting data from tweets

该问题实现两个方法，分别为获取时间段，获取被@者。

1. 获取时间段

首先按照时间顺序将tweets列表排序（通过比较器实现），再获取首位两个Timestamp加入结果即可。





1. 获取被@者

遍历tweets，将每个Tweet的text用split方法按空格分隔，随后筛选出以@开头的字符串，由于不区分大小写，统一切换成小写，再判断@后的字符串是否符合username的规则（这里采用限定ascii码范围法），符合规则且还未在结果中的就可以添加到结果了。

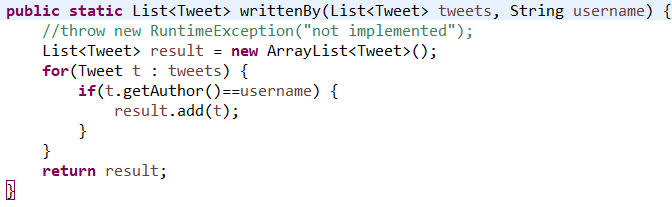


### Problem 2: Filtering lists of tweets

该问题实现三个筛选方法。

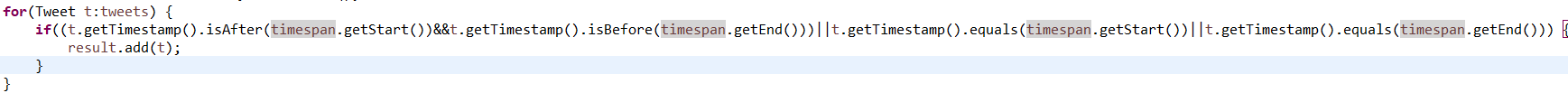
1. 按作者筛选

直接遍历就OK了



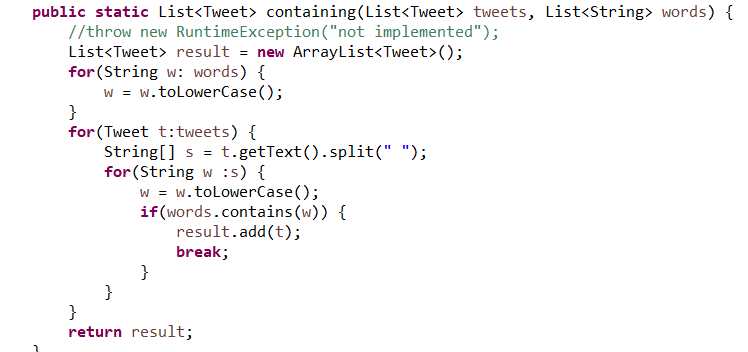
1. 按时间段筛选

同样是遍历，调用instant的isAfter和isBefore方法进行比较。



1. 按单词表筛选

还是用split将text分隔，然后遍历并调用contains方法判断。注意不区分大小写，同意转化成小写。



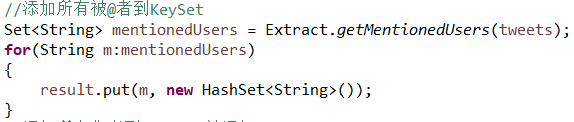
### Problem 3: Inferring a social network

要解决两个问题。

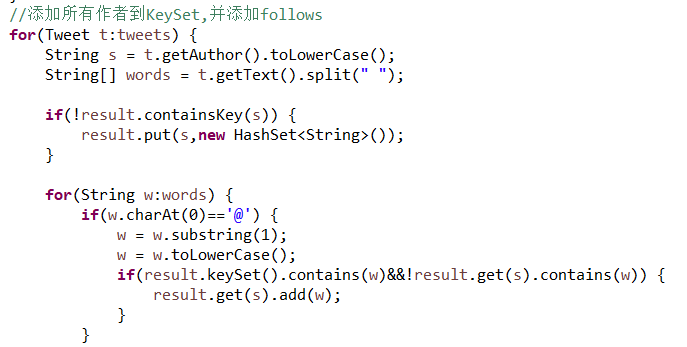
1. 关注列表

首先将所有在text中被@的用户添加到KeySet

（调用Extract中的getMentionedUsers）



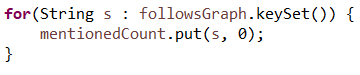
再将所有作者添加进去，同时添加他们关注的人，注意要避免重复和不区分大小写



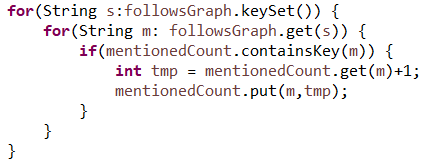
1. 影响力排行榜

建立一个新的Map，Key为用户名，Value为粉丝的数量

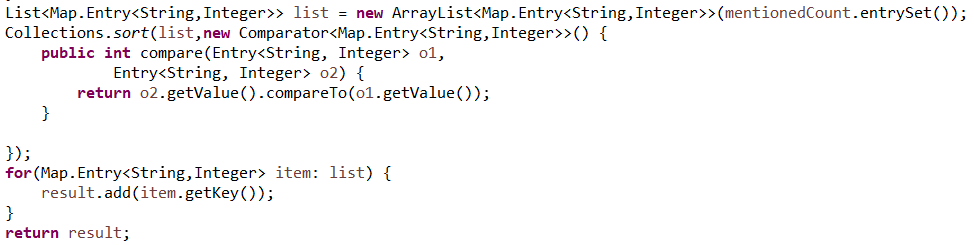
首先初始化



遍历关注列表，统计粉丝数量



从大到小排序（比较器实现）后返回结果

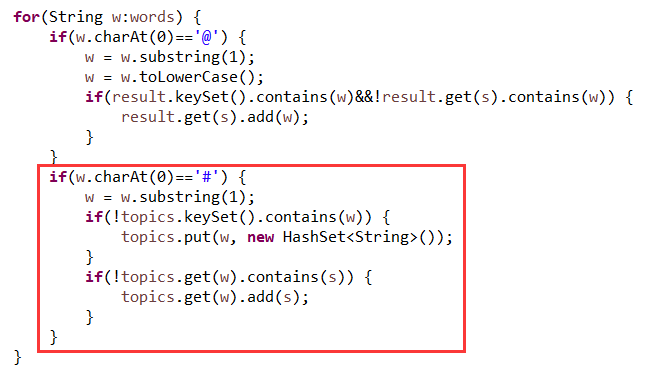


### Problem 4: Get smarter

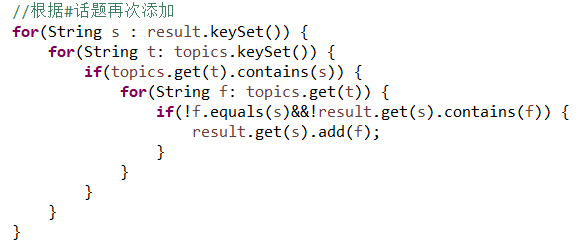
此处我选的常用主题标签。

建立新的Map，Key为主题名（#后面的）,Value为引用标签的用户列表。

在Problem3中构建关注列表的同时，按照#字符构建标签列表。



得到主题标签列表后，按照其内部相关用户，对关注列表进行再处理



# 实验进度记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 2019-02-25 | 14:00-16:30 | P1 | 按计划完成 |
| 2019-03-04 | 18:00-21:00 | P2 | 按计划完成 |
| 2019-03-11 | 18:00-20:00 | P3 | 按计划完成 |
| 2019-03-12 | 18:00-22:00 | P4 | 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的使用 | 查阅资料 |
| 项目目录结构设置 | 请室友帮忙解决了 |
|  |  |

# 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

以后编程先写SPEC，再写测试用例，最后写代码。

要注意能用private尽量用。

一定要边实验边写报告。

以后要实时用Git管理项目，强化Git使用技能。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

很对胃口，至少没有指针。

1. 关于Eclipse IDE

功能很强大，但是想全都挖掘出来并熟练掌握很难。

1. 关于Git和GitHub

目前算是体会到了它的用处，以后工作还会一直用，所以要更熟练运用。

1. 关于CMU和MIT的作业

设计的比HIT其他课程的实验好，就是英文看不懂。

1. 关于本实验的工作量、难度、deadline

不难，慢慢写就好了。

1. 关于初接触“软件构造”课程

大一下学期学过JavaEE，编程上没什么困难，但写代码效率还是低，以后要加强。