

**2020年春季学期  
计算机学院《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 尤国淳 |
| 学号 | 1180300819 |
| 班号 | 1803008 |
| 电子邮件 | 664429961@qq.com |
| 手机号码 | 18845636596 |

**目录**

[1 实验目标概述 1](#_Toc29325521)

[2 实验环境配置 1](#_Toc29325522)

[3 实验过程 1](#_Toc29325523)

[3.1 Magic Squares 1](#_Toc29325524)

[3.1.1 isLegalMagicSquare() 1](#_Toc29325525)

[3.1.2 generateMagicSquare() 2](#_Toc29325526)

[3.2 Turtle Graphics 3](#_Toc29325527)

[3.2.1 Problem 1: Clone and import 3](#_Toc29325528)

[3.2.2 Problem 3: Turtle graphics and drawSquare 3](#_Toc29325529)

[3.2.3 Problem 5: Drawing polygons 4](#_Toc29325530)

[3.2.4 Problem 6: Calculating Bearings 4](#_Toc29325531)

[3.2.5 Problem 7: Convex Hulls 4](#_Toc29325532)

[3.2.6 Problem 8: Personal art 5](#_Toc29325533)

[3.2.7 Submitting 6](#_Toc29325534)

[3.3 Social Network 6](#_Toc29325535)

[3.3.1 设计/实现FriendshipGraph类 6](#_Toc29325536)

[3.3.2 设计/实现Person类 6](#_Toc29325537)

[3.3.3 设计/实现客户端代码main() 7](#_Toc29325538)

[3.3.4 设计/实现测试用例 8](#_Toc29325539)

[4 实验进度记录 8](#_Toc29325540)

[5 实验过程中遇到的困难与解决途径 8](#_Toc29325541)

[6 实验过程中收获的经验、教训、感想 9](#_Toc29325542)

[6.1 实验过程中收获的经验和教训 9](#_Toc29325543)

[6.2 针对以下方面的感受 9](#_Toc29325544)

# 实验目标概述

本次实验通过求解三个问题，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

下载jre和jdk进行安装，配置环境变量。

下载安装Idea，使用学生证号激活，建立project并关联到git的仓库。

写好代码后，配置调试环境，使用jdk10.0.2，其他ide都能自动完成，比较方便。

仓库地址：<https://github.com/Stauch99/Lab1-1180300819>

# 实验过程

## Magic Squares

对幻方的判断函数进行编程，对幻方生成函数的代码理解。

### isLegalMagicSquare()

观察了一下几个txt文件，里面幻方的阶数都不同，所以应该先判断幻方的阶数，然后通过阶数计算出幻和，用于后面判定幻方是否符合行列对角线之和等于幻和。

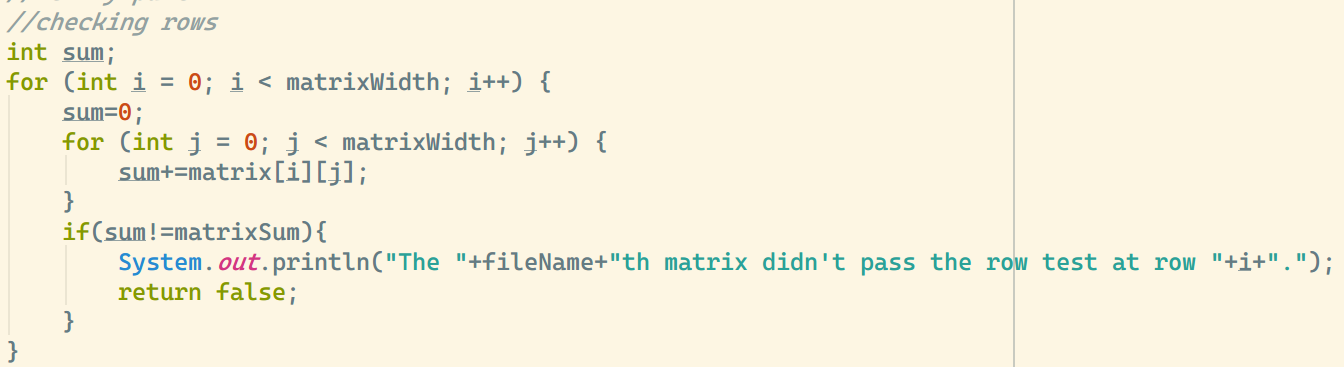
*//matrix exponent needs to check the \t symbol*int matrixWidthTabSplit = line.split("\t").length;  
int matrixWidth = line.split("\\s+").length;

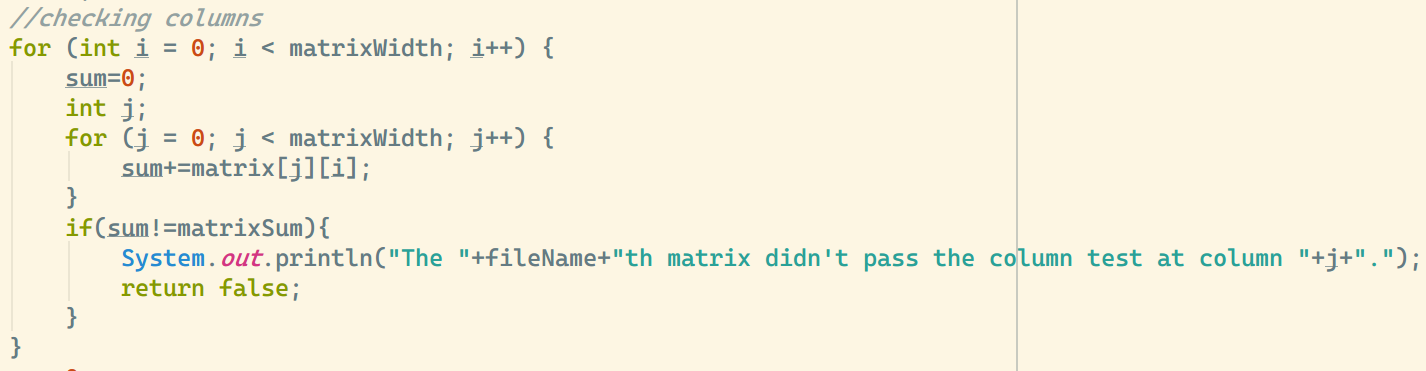
if (matrixWidth!=matrixWidthTabSplit){  
 System.*out*.println("The "+fileName+"th matrix has used none table blank.");  
 return false;  
}

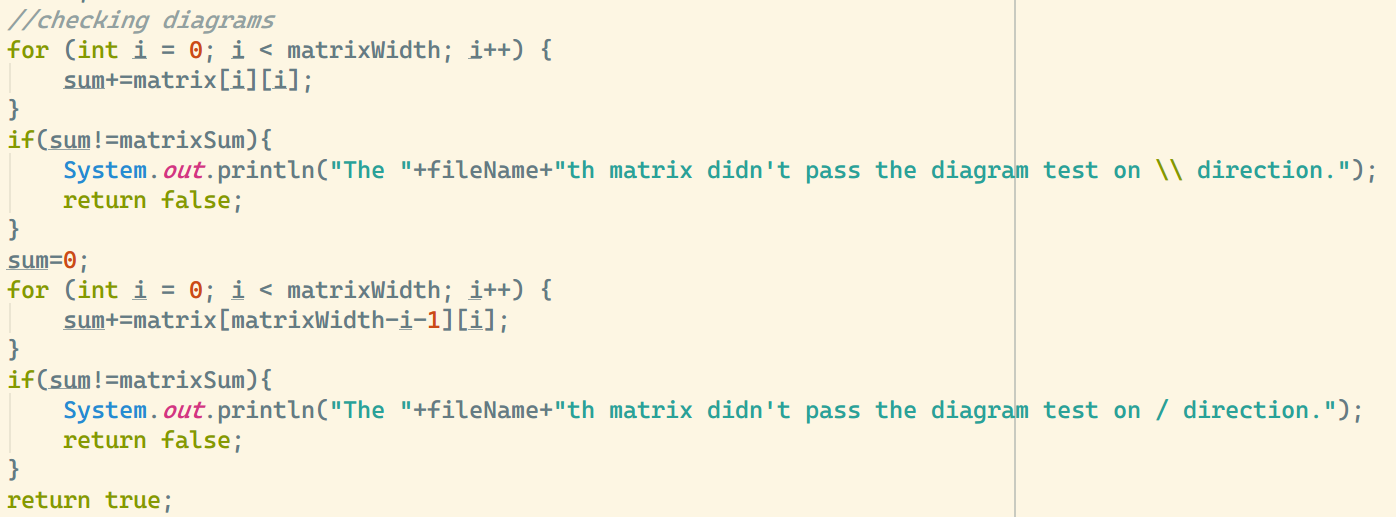
int matrixSize = matrixWidth\*matrixWidth;

int matrixSum = matrixWidth\*(matrixSize+1)/2;

随后对行列对角线进行判定，如果不满足条件，输出提示，告知错误位置。

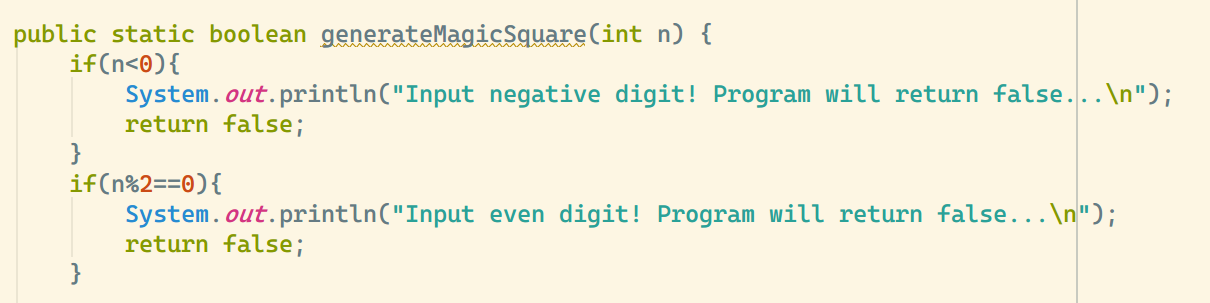






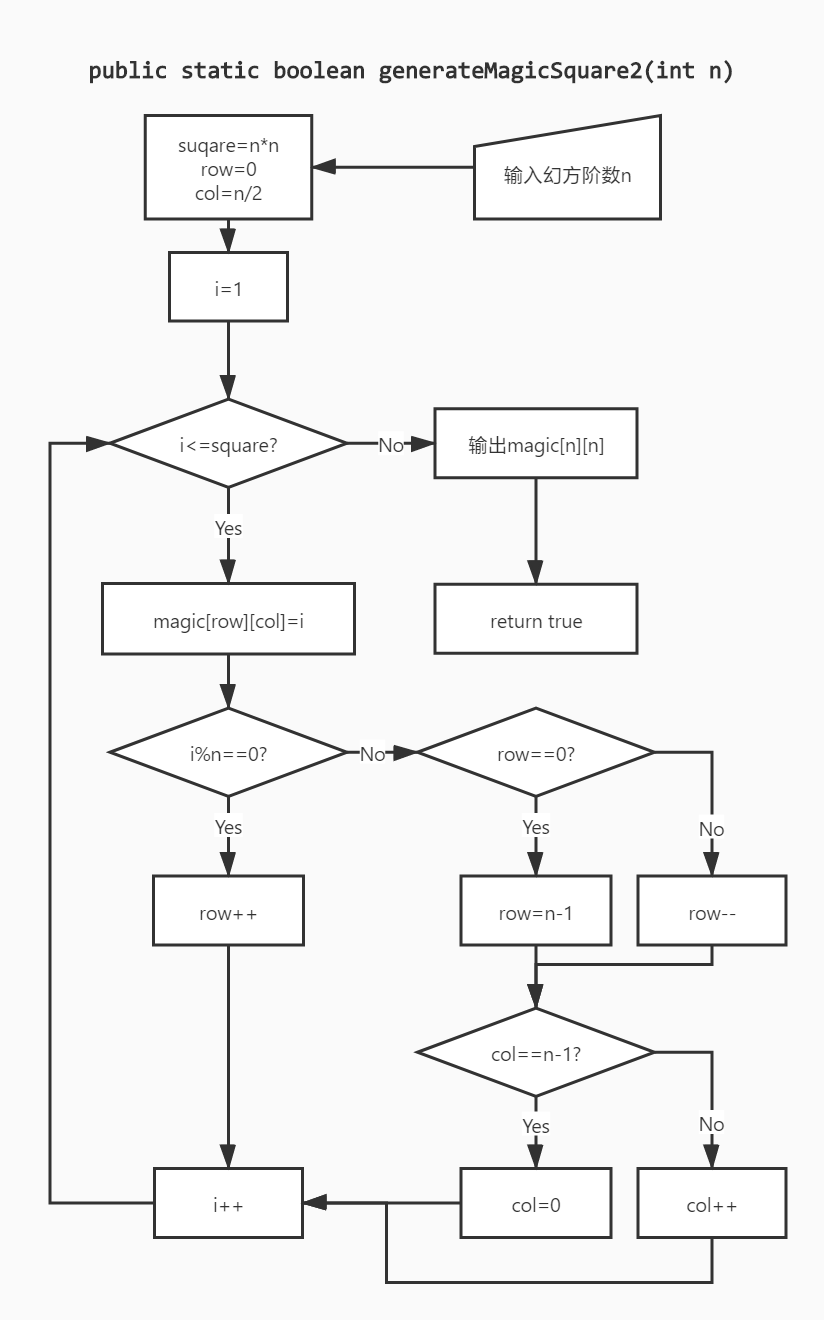
如果全部核验都能通过，就return true

### generateMagicSquare()

这个是给的现成代码，让改写一小部分，然后画一个流程图。

前面这里加入对负数和偶数的判定，输出提示语并返回false。

流程图：



## Turtle Graphics

小乌龟画图，小时候微机课有学习过，前几个问题都比较基础，后面的问题和小乌龟没啥关系了……主要是一个gift-wrapping算法。

### Problem 1: Clone and import

下载的时候Github非常的卡，于是从QQ群下载的其他同学分享的文档。

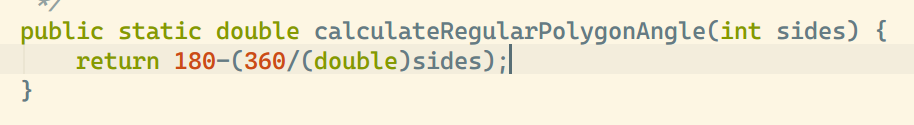
### Problem 3: Turtle graphics and drawSquare

这个没啥说的，画条长为a的边，转90度，重复四次，正方形就画好啦！

主要是熟悉一下基本操作，为后面打基础。

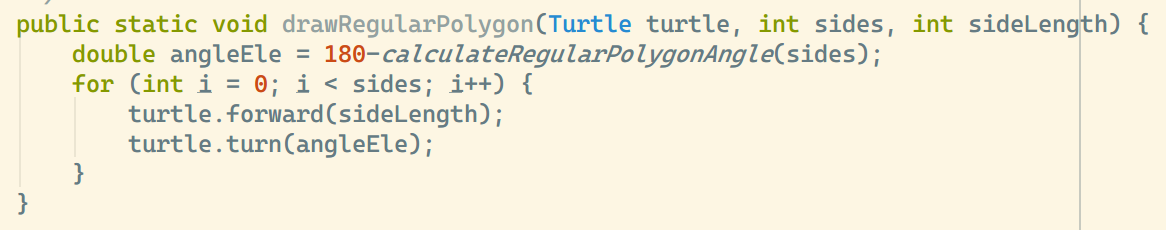
### Problem 5: Drawing polygons

正多边形内角公式推导了一下，写一个根据边数计算内角大小的函数就ok了。



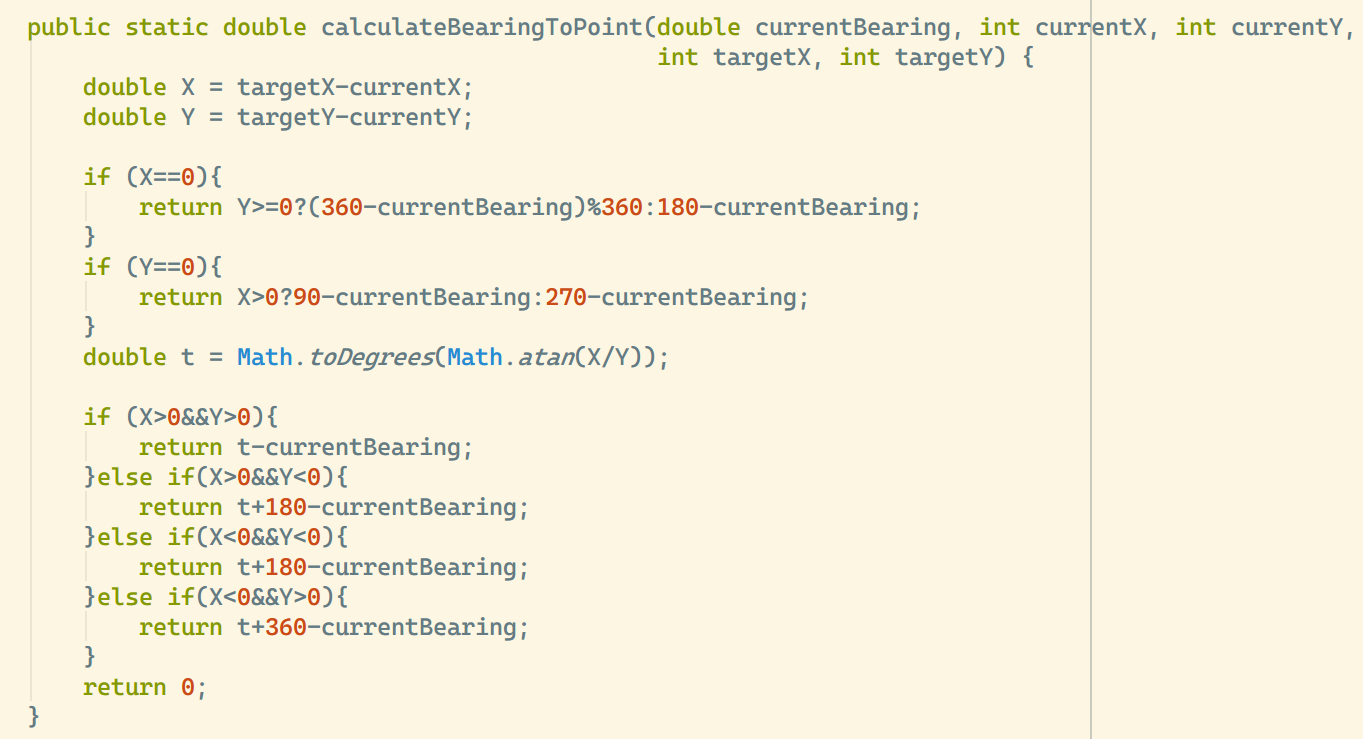
还有一个从角反推边数的公式，不赘述。

绘制多边形时，小乌龟转的是外角，所以需要求内角补角作为转向角。



### Problem 6: Calculating Bearings

第一个函数本质是计算在有一个偏移角的前提下，计算两个向量的夹角，这里要用Math里面的反三角函数，用哪个都差不多，得分情况判断对结果进行符号和范围处理，最后用%360规范结果在0-360之间。

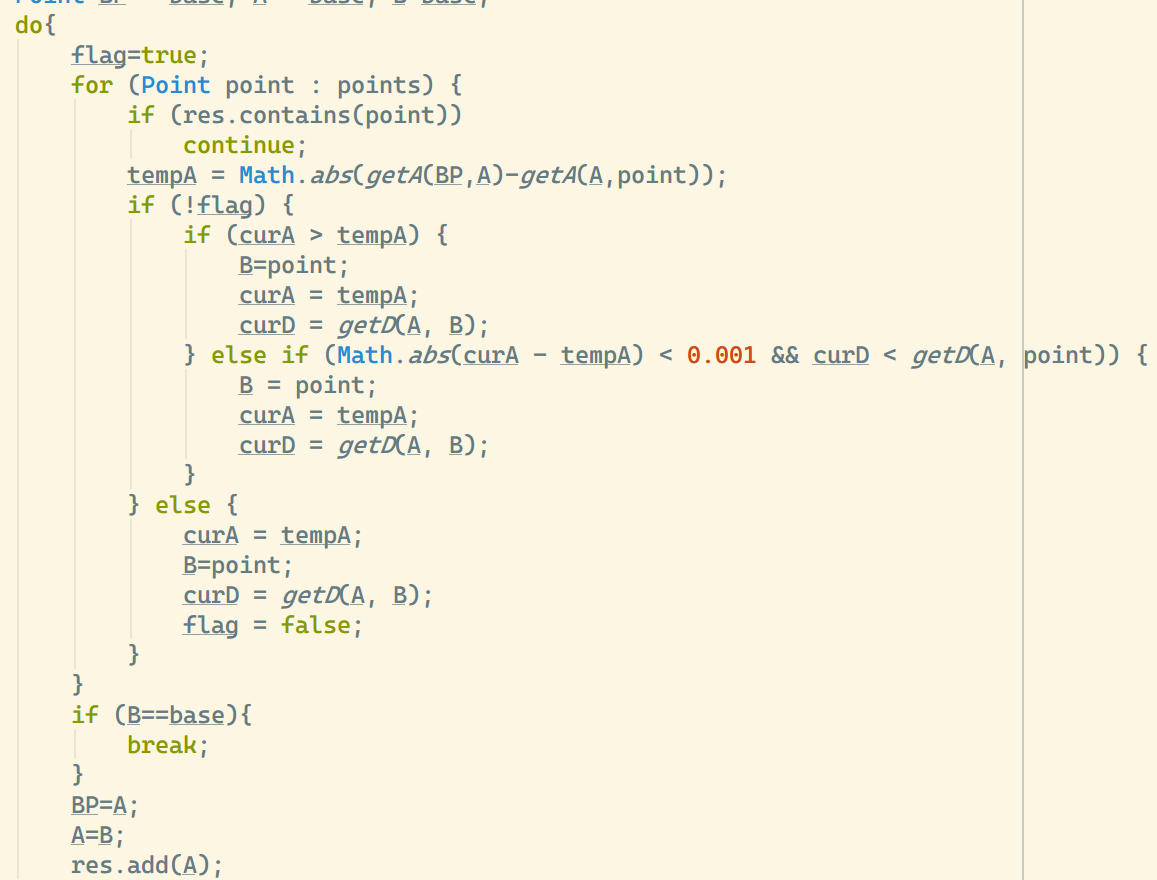


第二个函数是计算一组点进行跳转时的偏移角，第一个函数会用到。把list里面的点组遍历一遍就行，问题不大。

### Problem 7: Convex Hulls

凸包问题，一开始感觉很难，琢磨琢磨发现这个gift-wrapping算法实际上就是两步：

1. 循环找相对当前偏移方向最小的后继点
2. 循环到起始点结束寻找，凸包围出来了



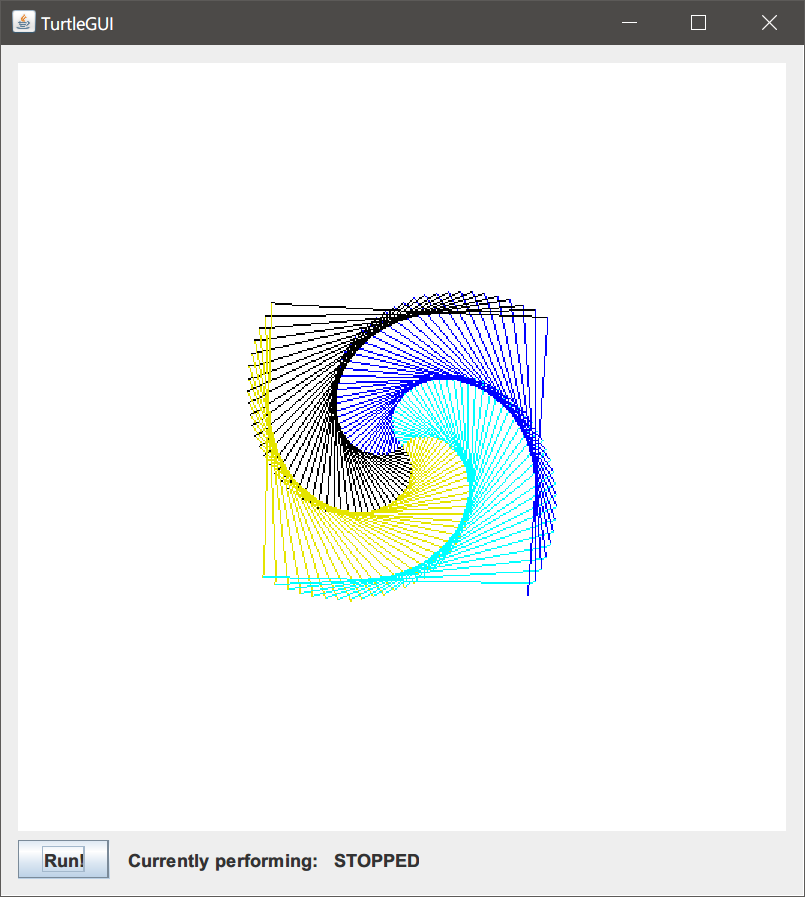
核心在循环的处理，各种临时变量容易混，比较考验算法能力。

有个小点，角度是double，不能用==直接比较，要这样：

### Problem 8: Personal art

画幅画，写一个循环函数，每次让小乌龟运动改变一点，就可以画出很神奇的形状。

浏览了一下其他class里面的方法，发现还能换颜色，于是就画了个花里胡哨的东东：



### Submitting

使用IDEA的VCS功能，commit然后push，现代IDE就是如此贴心（笑

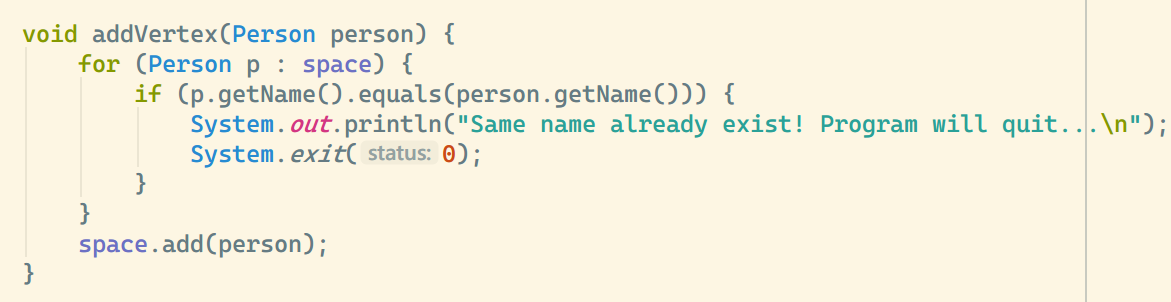
## Social Network

用到了上学期数据结构与算法里面图结构和最短路径的知识。

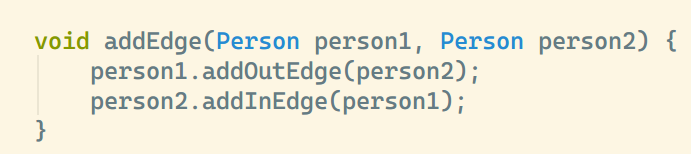
### 设计/实现FriendshipGraph类

设置了一个池，新的Person对象都丢进去，便于重名判断和后面计算距离的遍历。

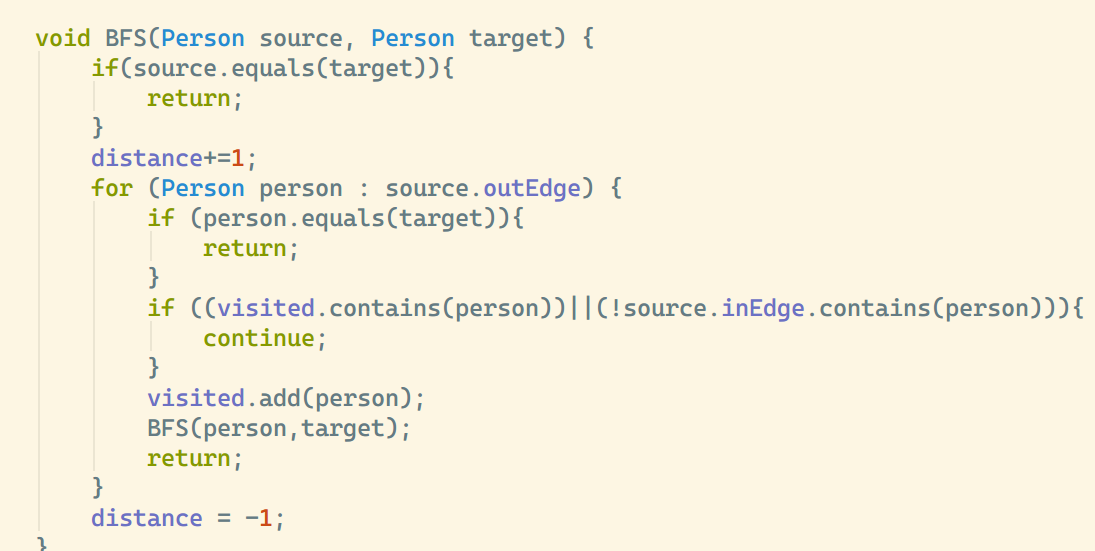
AddVertex要处理一下重名的情况：



AddEdge是按照有向图写的，所以要注意下：



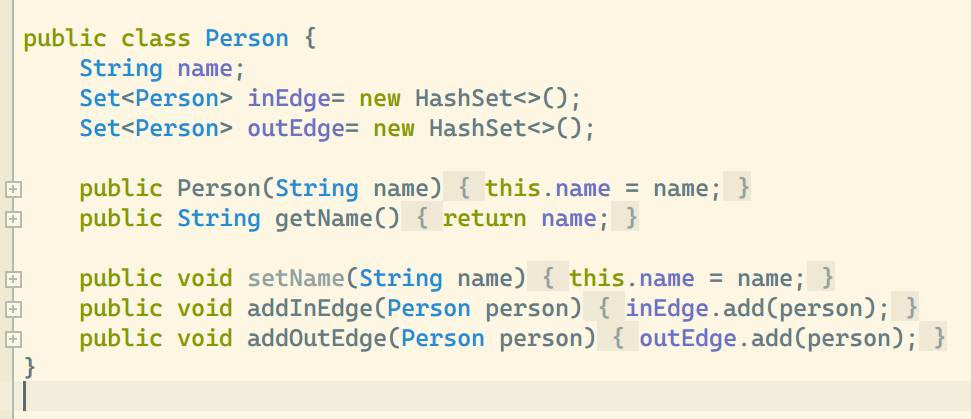
Lab里推荐用BFS，这里BFS是自己琢磨写的，java写起来比C简单，高级语言真香



通过BFS递归，逐层寻找目标Person，经历过的加入visited集合，每递归一层，在函数内distance+1记录路长。

### 设计/实现Person类

因为要预留有向图的能力，所以每个Person都设置InEdge和OutEdge



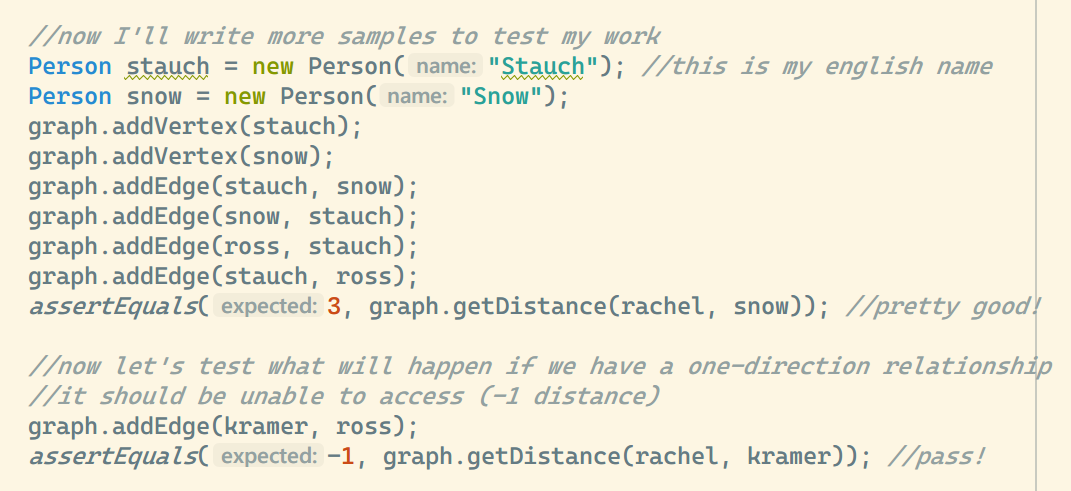
在判断相连的时候，就要保证源Person的两个Set里面都有目标Person。

### 设计/实现客户端代码main()

### 

从pdf里面复制的，测试了自己的代码，结果和预期一致！

### 设计/实现测试用例



这里测试了更长的路径计算，测试了存在单向桥求最短路径的情况，结果都很好，-1就是无穷远。

# 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 |
| 2020-03-7 | 14:00-17:30 | 完成问题1 | 按计划完成 |
| 2020-03-11 | 9:00-15:00 | 完成问题2 | 按计划完成，中间吃了个饭 |
| 2020-03-15 | 12:30-14:30 | 完成问题3 | 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 |
| Jar打包，maven和gradle都没太弄明白，浪费了很多时间 | 用IDEA自带的build artifact勉强生成了能运行的jar文件。 |
|  |  |
|  |  |

# 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

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

比C好写多了，很喜欢

1. 关于Eclipse IDE

安装配置时链接服务器很慢很慢，多次因为速度过慢安装失败，最后换用了IDEA。

1. 关于Git和GitHub

Git之前用过，所以还是很好用的，不过有些新知识还是通过课堂了解到。

1. 关于CMU和MIT的作业

很好的锻炼了java基础，让我热身，但是由于来源不同，实验要求和目录就很混乱，希望还是**做好本地化**之后再作为实验发布。

1. 关于本实验的工作量、难度、deadline

难度都在打包jar上。最后真的做的很心烦，本来留一天写报告，研究jar打包弄了好久好久，最后还是碰巧弄出来的……

1. 关于初接触“软件构造”课程

建议老师用中文课件或者直接英文授课，课件和讲述语言不一样，脑子得来回切，看的十分疲惫……另外老师很多时候表述过于复杂冗余，导致一个事翻来覆去的说，失去焦点了。这个问题是很多同学都反馈的，但是没有在群里说，希望老师能够理解。