

**2023年春季学期**

**计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 胡宏彬 |
| 学号 | 2021111204 |
| 班号 | 2137101 |
| 电子邮件 | 1953802638@qq.com |
| 手机号码 | 17816078651 |

**目录**

[1 实验目标概述 1](#_Toc18284)

[2 实验环境配置 1](#_Toc10966)

[2.1 配置Java8 1](#_Toc17446)

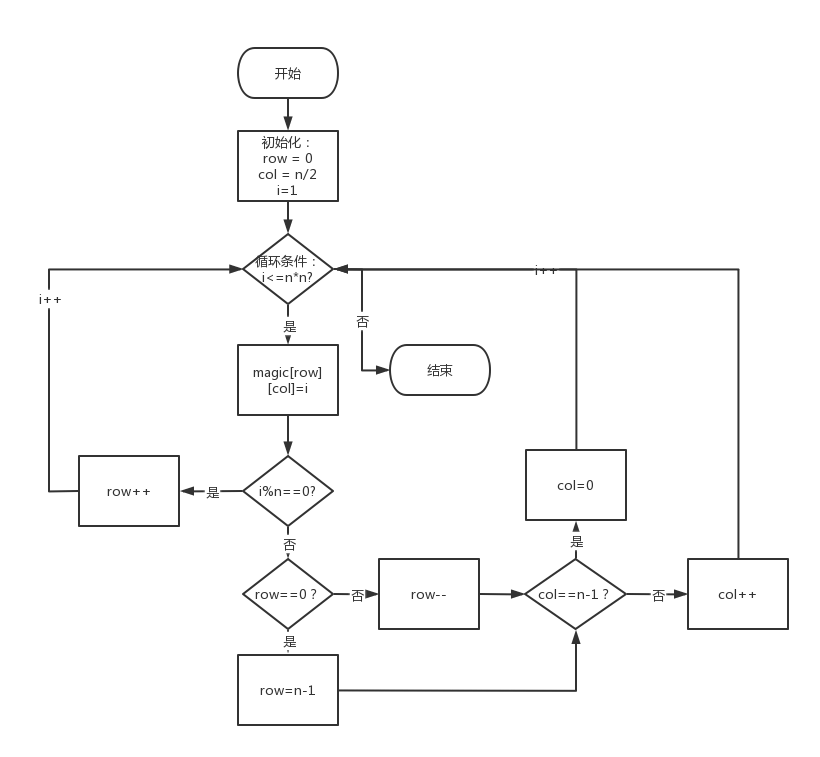
[2.2 下载IDEA、Junit 2](#_Toc13307)

[3 实验过程 4](#_Toc24033)

[3.1 Magic Squares 5](#_Toc11758)

[3.1.1 isLegalMagicSquare() 5](#_Toc26298)

[3.1.2 generateMagicSquare() 6](#_Toc19178)

[函数流程图 ：  7](#_Toc5707)

[3.2 Turtle Graphics 9](#_Toc8519)

[3.2.1 Problem 1: Clone and import 9](#_Toc31271)

[3.2.2 Problem 3: Turtle graphics and drawSquare 9](#_Toc32380)

[3.2.3 Problem 5: Drawing polygons 10](#_Toc3925)

[3.2.4 Problem 6: Calculating Bearings 12](#_Toc25396)

[3.2.5 Problem 7: Convex Hulls 13](#_Toc29832)

[3.2.6 Problem 8: Personal art 14](#_Toc31016)

[3.2.7 Submitting 15](#_Toc25378)

[3.3 Social Network 15](#_Toc6293)

[3.3.1 设计/实现FriendshipGraph类 15](#_Toc19783)

[3.3.2 设计/实现Person类 16](#_Toc25219)

[3.3.3 设计/实现客户端代码main() 16](#_Toc12679)

[3.3.4 设计/实现测试用例 17](#_Toc24131)

[4 实验进度记录 18](#_Toc30957)

[5 实验过程中遇到的困难与解决途径 18](#_Toc15298)

[6 实验过程中收获的经验、教训、感想 19](#_Toc6775)

[6.1 实验过程中收获的经验和教训（必答） 19](#_Toc10670)

[6.2 针对以下方面的感受（必答） 19](#_Toc22328)

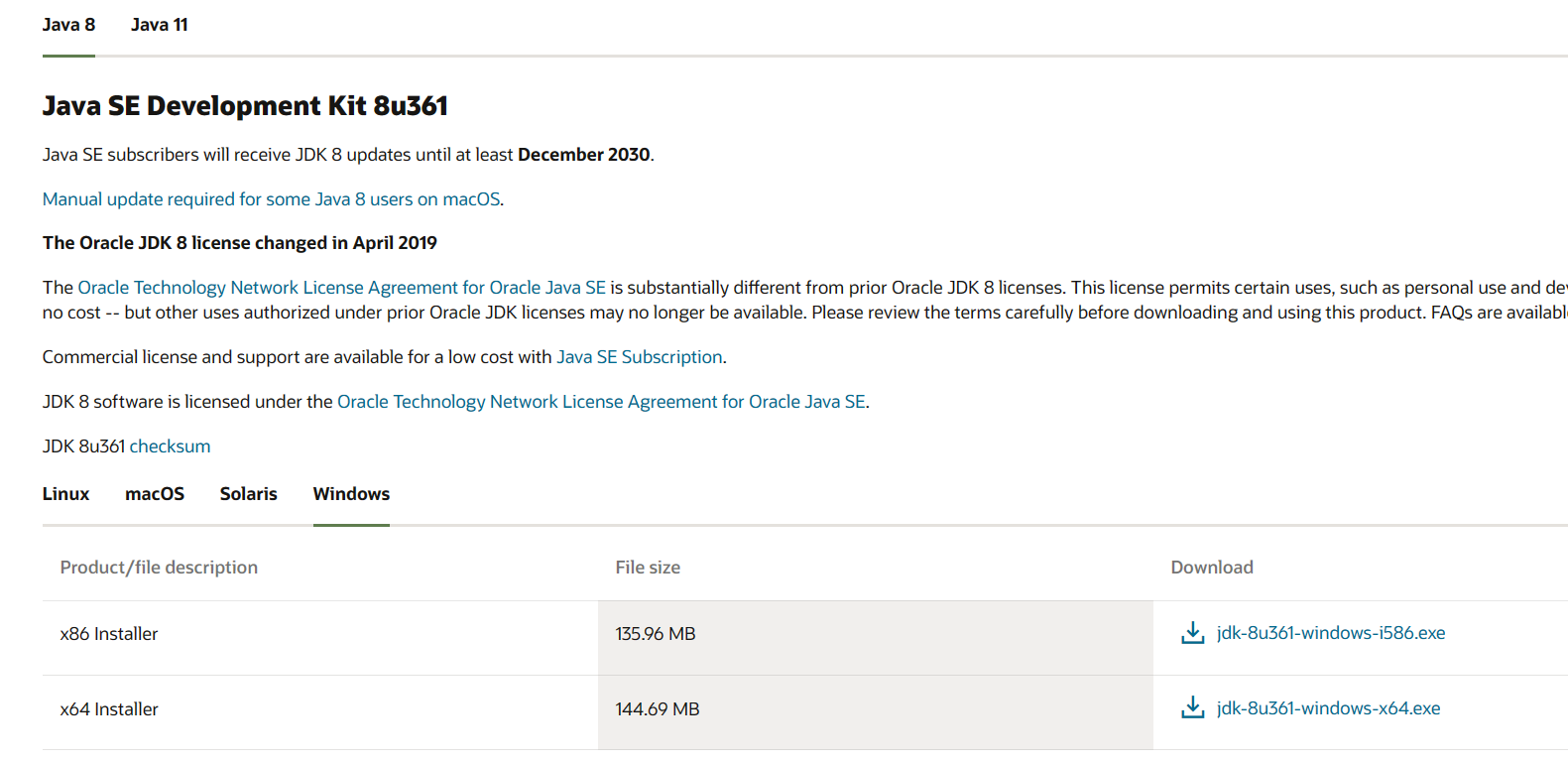
# 实验目标概述

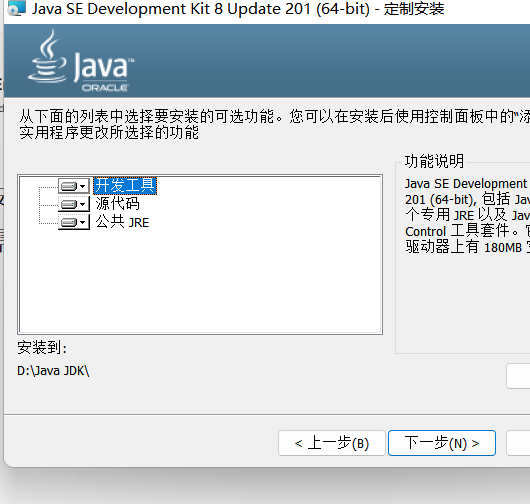
本次实验通过求解三个问题，训练基本 Java编程技能，能够利用Java OO开发基本的功能模块，能够阅读理解已有代码框架并根据功能需求补全代码，能够为所开发的代码编写基本的测试程序并完成测试，初步保证所开发代码的正确性。另一方面，利用Git作为代码配置管理的工具，学会Git的基本使用方法。

# 实验环境配置

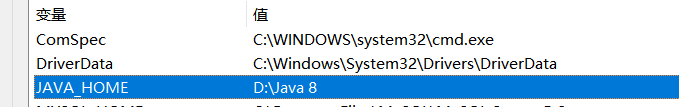
## 配置Java8

**2.1.1下载java8：从Oracle官网找到JAVA模块，选择JAVA8的Windows版本，根据电脑配置选择x64安装包安装，最后打开安装包按提示下载即可。**





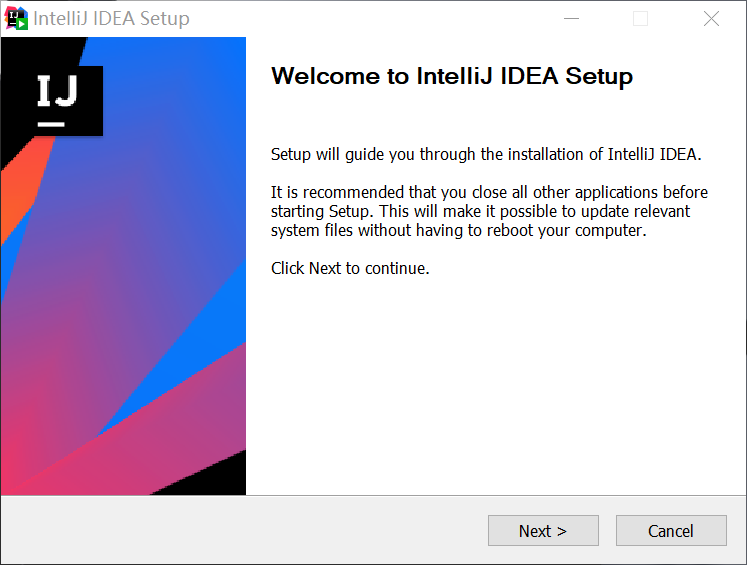
**2.1.2配置java8环境变量：**





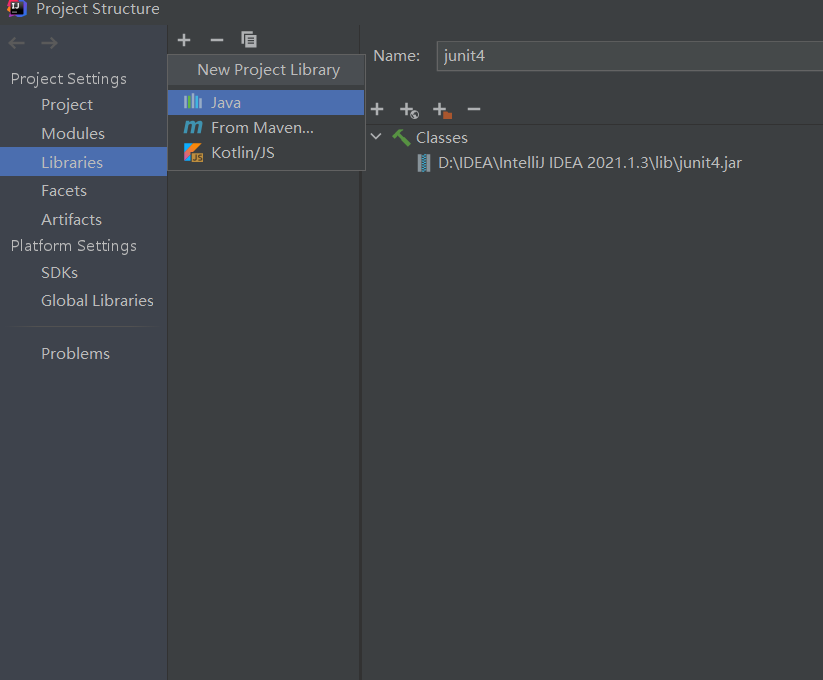
## **下载IDEA、Junit**

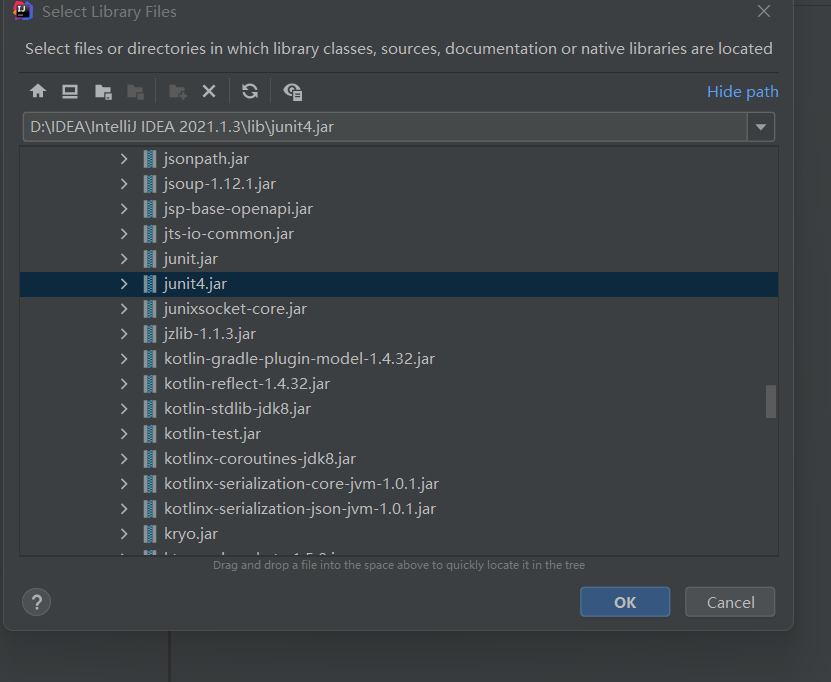
**IDEA下载界面：从官网下载IDEA安装包，按步骤next下载即可。**



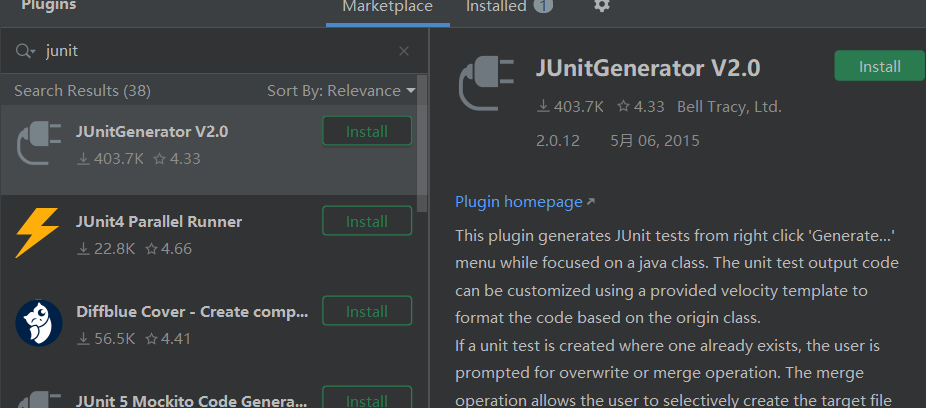
Junit 配置：

从File 选择Project structure，在Libraries点击+号，选择java，找到IDEA的lib目录下的Junit4.jar，添加并应用即可。





选择JUnit4相关插件下载，下载成功后重启IDEA即可



GitHub Lab仓库的URL地址：https://github.com/ComputerScienceHIT/HIT-Lab1-2021111204.git。

# 实验过程

请仔细对照实验手册，针对三个问题中的每一项任务，在下面各节中记录你的实验过程、阐述你的设计思路和问题求解思路，可辅之以示意图或关键源代码加以说明（但无需把你的源代码全部粘贴过来！）。

为了条理清晰，可根据需要在各节增加三级标题。

## Magic Squares

理解:1、幻方是一个有n\*n个不同数字、且每行、每列和斜线上都有相同的和的方形结构。要求写出程序判断输入一个矩阵是否是幻方，并且构造幻方。

2、主函数首先要判断5个txt文件中的矩阵是否为幻方（需要读取各个文件中的数据，然后进行识别），然后要运行函数构造一个幻方存入6.txt文件之中。

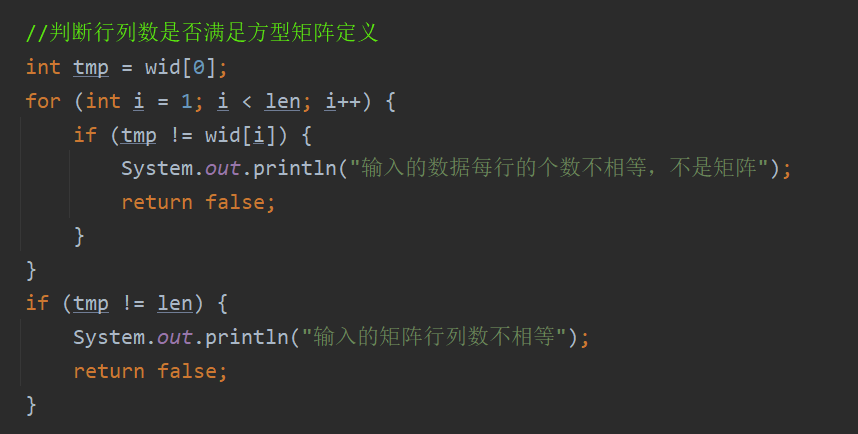
### isLegalMagicSquare()

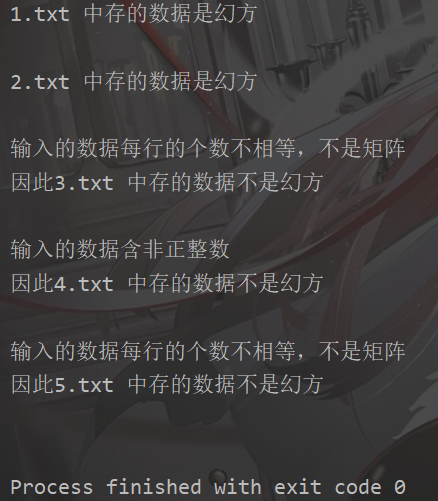
过程：设计思路：

1. 首先要读取矩阵，要考虑如何设计矩阵的变量类型，以及读取矩阵中可能发生的一些错误，需要识别并抛出。
2. 判断读取的矩阵是否符合方形矩阵的定义，即行数和列数是否相等、每一行的数据数是否相等，若不满足，返回false。
3. 判断读取的数是否为正数字，通过和0比较判断正负，通过正则表达式判断是否为数字（不是则由题意说明分隔符不为\t），通过浮点型和整型之间的类型转换判断是否有小数，以上有一个不满足，返回false
4. 判断各行、各列、主次对角线的数字和是否相同，若不同，返回false。
5. 若以上判断都通过，判定矩阵为幻方，返回true。

过程：按思路构造对应函数。

关键代码：



运行结果如下：

### generateMagicSquare()

函数创建幻方原理：

1、首先创建一个n\*n的方阵空间。

2、把“1”放在中间一列最上边的方格中，从它开始，按对角线方向（从左下往右上）顺次把由小到大的各数放入方格中。

3、如果碰到顶，则折向底，如果到达右侧，则折向左边。

4、如果进行中轮到方格中已经有数或者到达右上角则退至前一格的下方。

当n\*n个数均填完时，幻方创建完毕。

### 函数流程图 ： wps

**思路：**

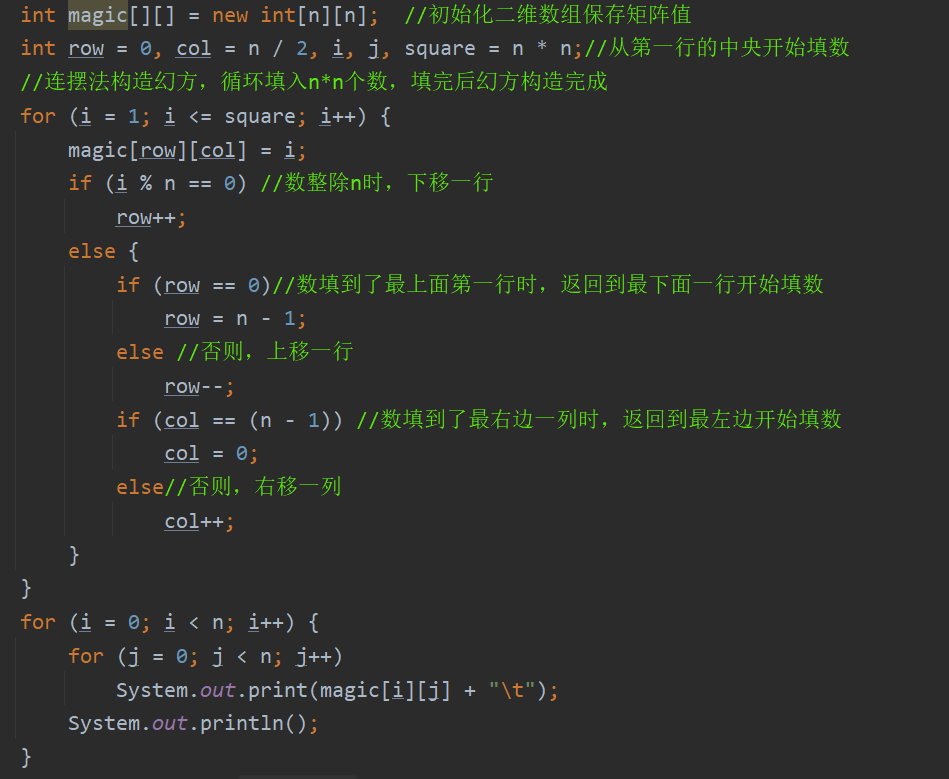
1. 将函数写入文件中，并在主函数中完成相关调用
2. 调用函数运行程序，并查看偶数、负数的情况，并分析原因

3、扩展函数：

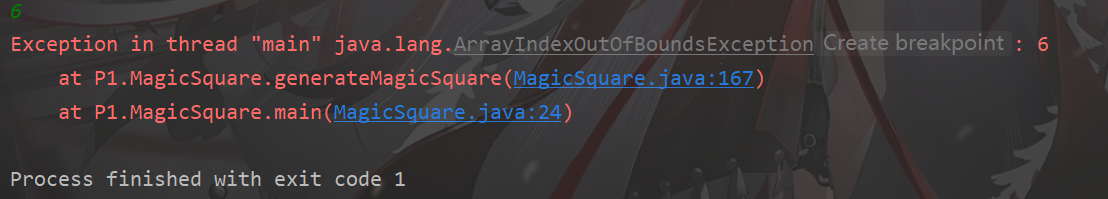
1）首先完成文件写入操作，通过字符缓冲输出流将矩阵写入文件

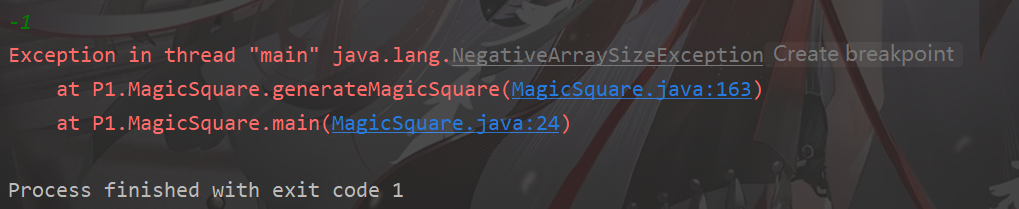
2）完成对输入偶数、负数情况的识别，使程序能够在该情况下优雅退出

代码：

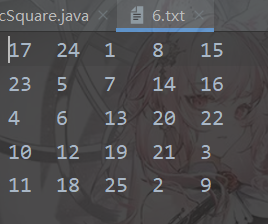
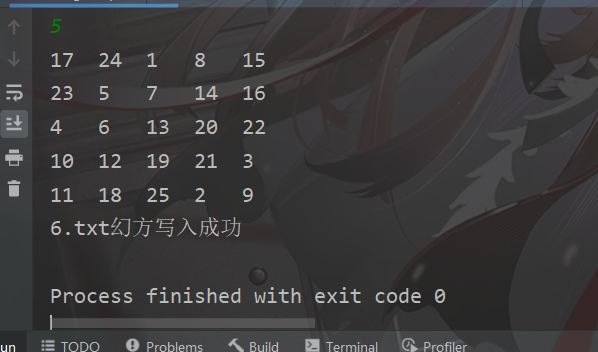


**过程及结果：**

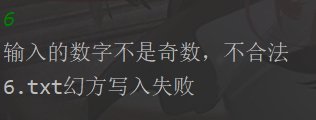
输入偶数结果：

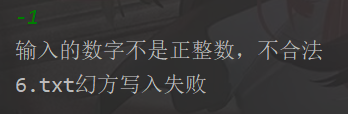
输入负数运行结果：

正常输入结果及文件情况：



扩展后：

输入偶数：

输入负数：

**分析：**

1. 输入偶数出现异常：

异常意义：数组下标越界异常

出现原因：输入的数是偶数时，由于下面一行必有一个数可以整除n，根据代码会使坐标下移一行，导致数组越界。

1. 输入负数出现异常：

异常意义：数组长度异常

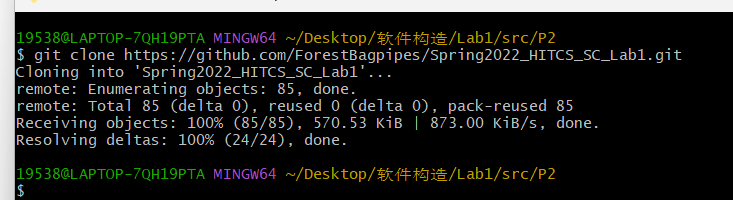
出现原因：输入的n是作为定义数组时数组的行数和列数的，由于n为负数，而数组的长度不能为负数，因此出现异常。

## Turtle Graphics

该实验要求我们先从对应的github上把代码clone下来，然后完成TurtleSoup中的各个方法，并运用该方法完成图形的绘制，并且最后完成个人艺术绘图。

### Problem 1: Clone and import

1、在对应的文件夹内右键打开 Git Bash

2、通过输入指令 git clone https://github.com/ForestBagpipes/Spring2022\_HITCS\_SC\_Lab1.git 从而将代码克隆到当前的文件夹内

然后将需要的文件移动到P2即可

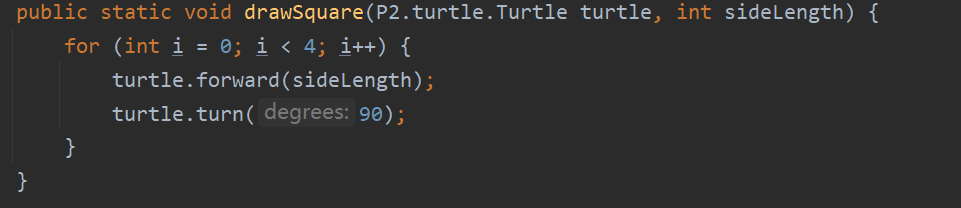
### Problem 3: Turtle graphics and drawSquare

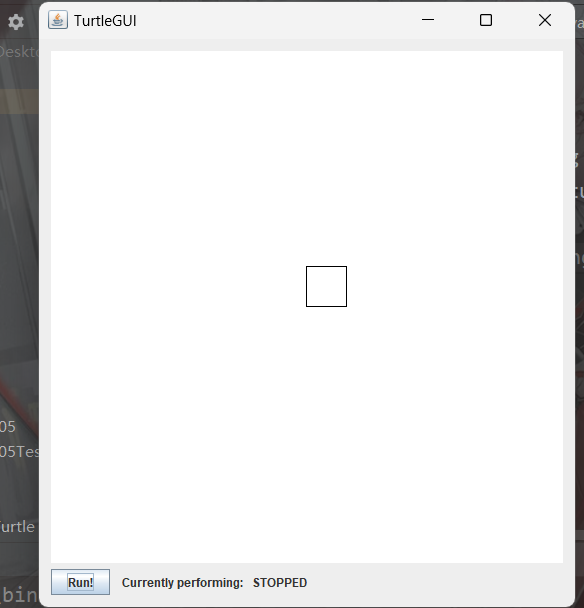
首先：由于Turtle被放入了P2包内部，所以在文件开头：package turtle 要改为：

package P2.turtle。

然后是画一个正方形：实现只需要每次走sidelength长，右旋90°，完成4次即可。

代码：





### Problem 5: Drawing polygons

画正多边形需要完成3个函数内容,并通过JUnit测试。

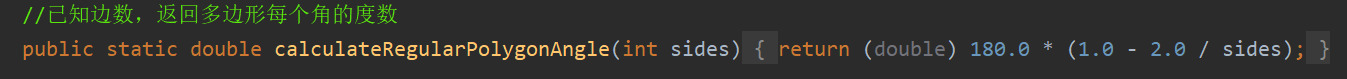
分别为：

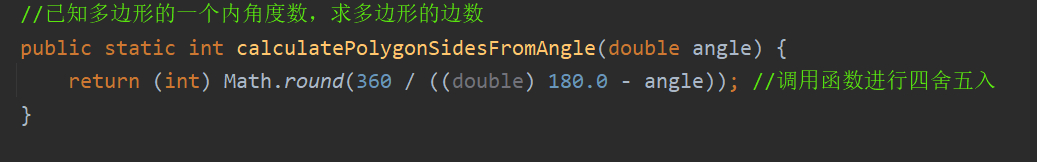
calculateRegularPolygonAngle：根据多边形边数计算内角，实现只需运用几何知识简单计算即可。

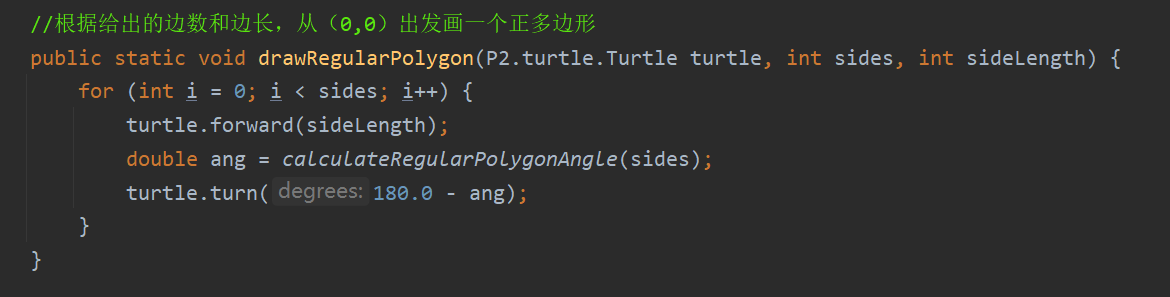
calculatePolygonSidesFromAngle：根据多边形内角计算多边形边数，实现需要运用几何知识，由于涉及除法，需要进行四舍五入，并且需要转换结果为整型输出。

drawRegularPolygon：画正多边形，原理同画正方形，需要调用calculateRegularPolygonAngle，实现简单。

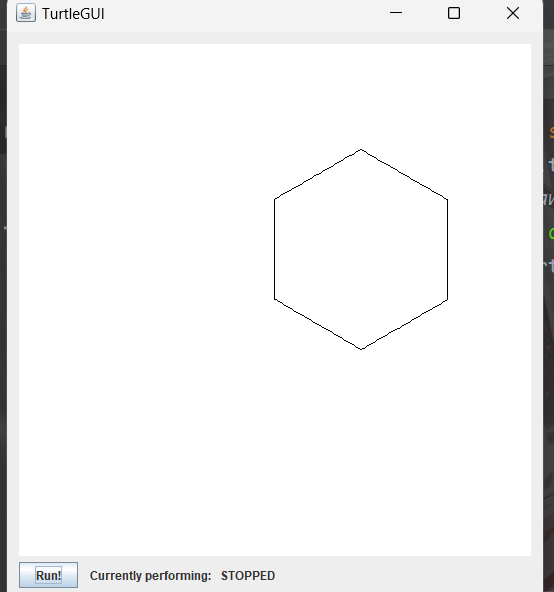
代码：





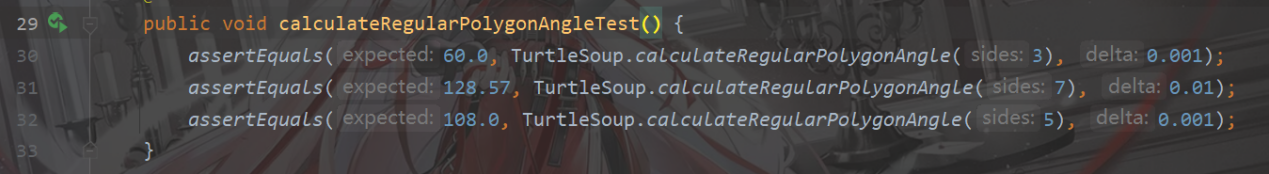


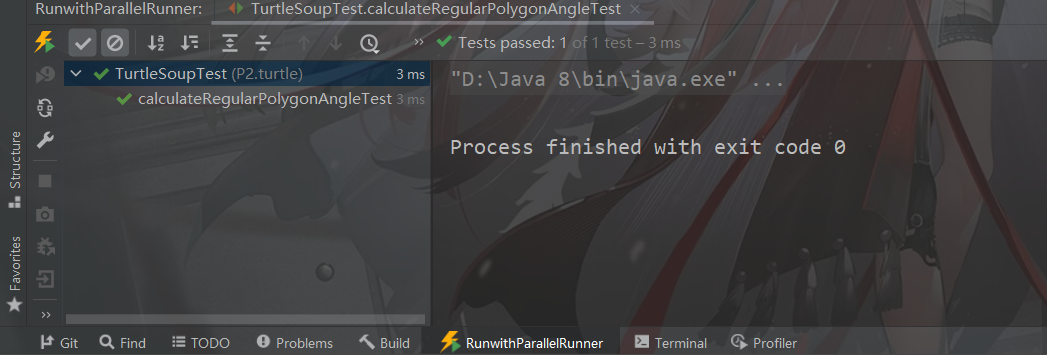
画六边形：



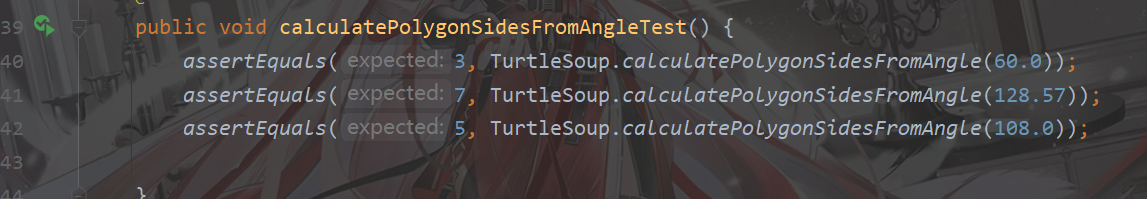
Junit测试：

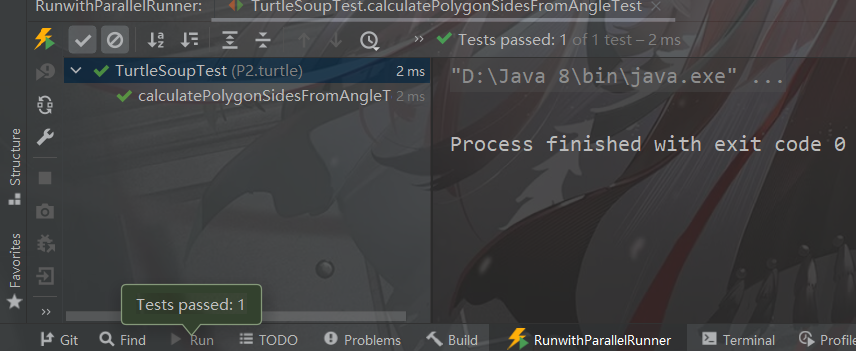
calculateRegularPolygonAngle：





calculatePolygonSidesFromAngle：





### Problem 6: Calculating Bearings

实现方法calculateBearingToPoint和calculateBearings，并通过JUnit测试。

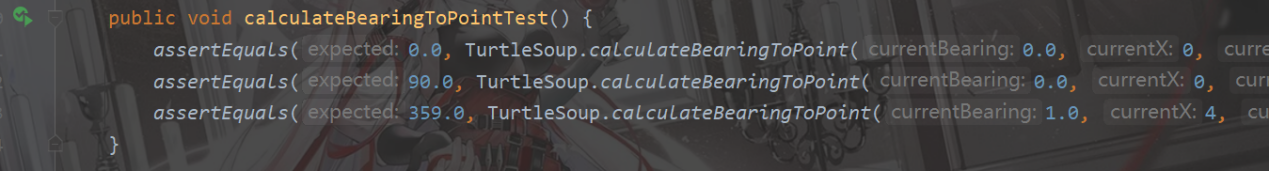
思路：

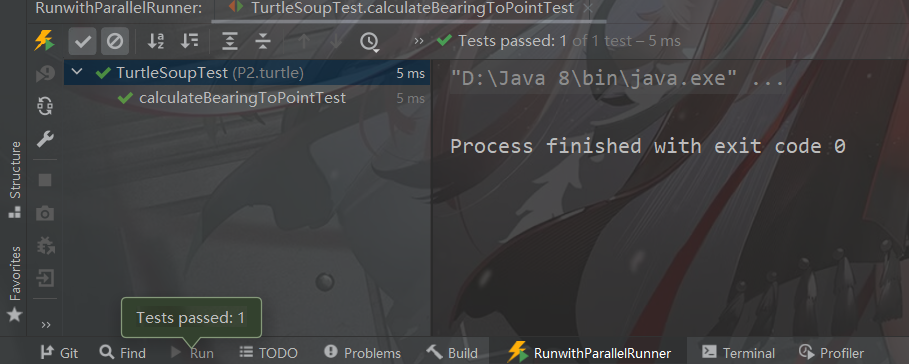
calculateBearingToPoint：首先根据点间距离是否为0判断两点是不是同一个，然后通过几何知识，调用Math.atan2()方法求出两点形成的直线与水平正半轴之间的夹角，从而得到其和竖直正半轴直接的夹角。以此角减去当前的夹角，若为负数加上360度，最终得到结果。

calculateBearings：根据得到的一系列x，y坐标，算出每次的偏转角并保存到集合中返回。

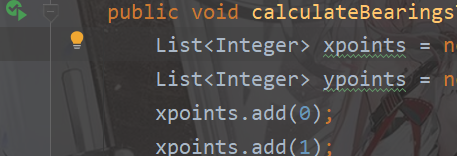
Junit测试：

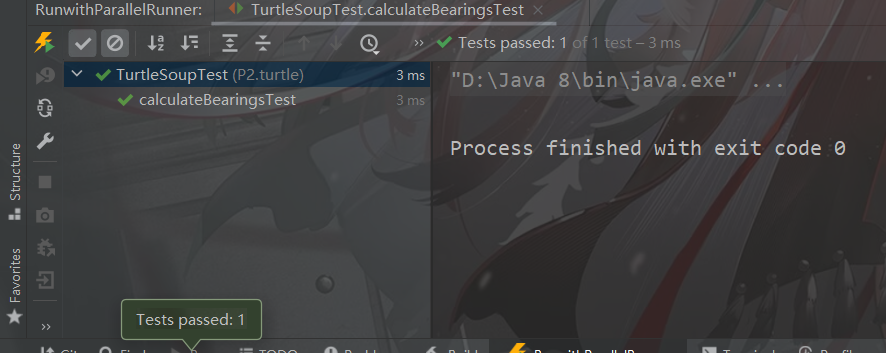
calculateBearingToPoint：





calculateBearings：





### Problem 7: Convex Hulls

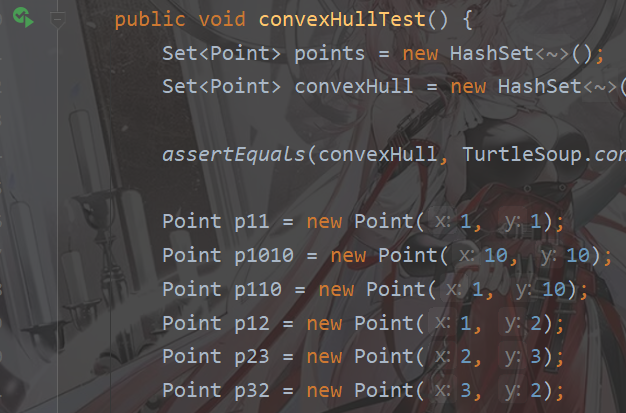
求解凸包问题，并通过Junit测试

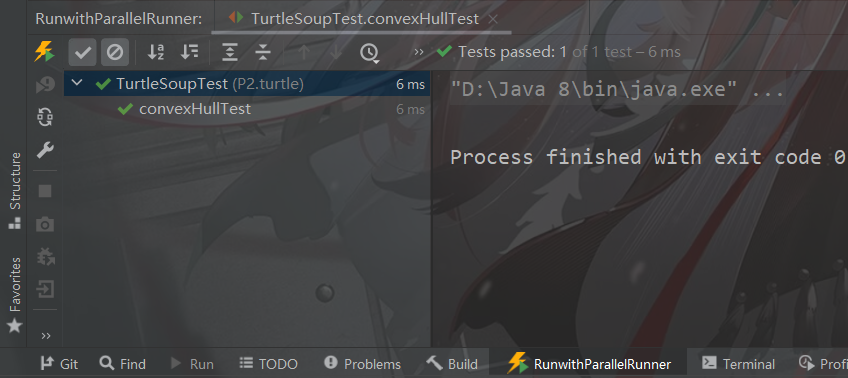
思路：首先找到整个点集中位于最左下角的点，即x最小的点中，y也最小的点，作为P1,然后遍历点击，找到右旋角度最小的点P2,P2显然在凸包上，然后从P2出发再依次找下去，直到最后，就能得到整个凸包。

关键代码：

double x3 = i.x() - curPoint.x(), y3 = i.y() - curPoint.y(); //当前凸包点到点i的向量坐标  
double angle = Math.*acos*((x1 \* x3 + y1 \* y3) / (Math.*sqrt*(x1 \* x1 + y1 \* y1) \* Math.*sqrt*(x3 \* x3 + y3 \* y3)));//用向量积的方式求夹角  
//当右旋角度相同时，选择距离较远的点  
if (angle < minAngle || (angle == minAngle && x3 \* x3 + y3 \* y3 > Math.*pow*(i.x() - minPoint.x(), 2) + Math.*pow*(i.y() - minPoint.y(), 2))) {  
 minPoint = i;  
 minAngle = angle;  
 x2 = x3;  
 y2 = y3;

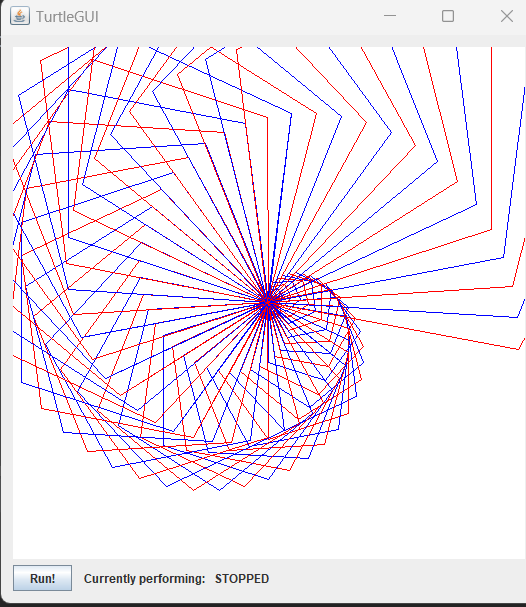
Junit测试:





### Problem 8: Personal art

方法：旋转50次，每次画一个正五边形，边长每次加5，每次旋转角度为7.2°，奇数次颜色为红，偶数次颜色为蓝色，即可。



### Submitting

在该文件夹下打开GitBash,使用git add指令，git commit指令添加到本地，最后使用git push<仓库地址> master 将当前的文件上传到远程仓库里去。

## Social Network

要求我们设计一个社交网络，网络为无向图，可以连接人与人，能够完成添加人结点，添加连接，计算两个人之间的最短社交距离，判断人是否重名等功能。

### 设计/实现FriendshipGraph类

思路：用HashMap的形式保存图，键为Person ，值为HashSet（Person）。

private HashMap<Person, HashSet<Person>> friendship;

addVertex: 向HashMap中添加键为Person，值为空HashSet的元素

关键代码：

friendship.put(p, new HashSet<Person>());

addEdge: 两个人向对方键的集合中添加自己。

关键代码：

friendship.get(A).add(B);

getDistance:运用广度优先搜索，遍历点集即可。

关键代码：

int distance = 0; //距离  
HashSet<Person> visited = new HashSet<>(); //记录访问过的顶点  
HashMap<Integer, HashSet<Person>> map = new HashMap<>(); //记录距离p1一定距离的顶点集  
map.put(0, new HashSet<Person>());  
map.get(0).add(p1);  
do {  
 if (map.get(distance).contains(p2)) {  
 return distance;  
 } else {  
 visited.addAll(map.get(distance));  
 map.put(distance + 1, new HashSet<Person>());  
 for (Person i : map.get(distance)) {  
 if (friendship.get(i).size() != 0) {  
 for (Person p : friendship.get(i)) {  
 if (!visited.contains(p)) {  
 map.get(distance + 1).add(p);  
 }  
 }  
 }  
 }  
 }  
 distance++;  
} while (map.get(distance).size() != 0);

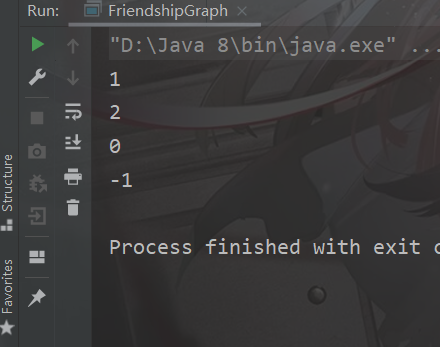
### 设计/实现Person类

思路：Person类私有属性只有Name。其他只要添加类中的构造方法即可。

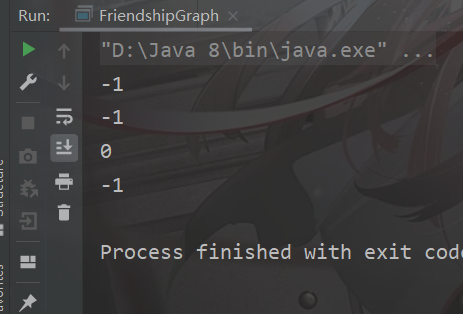
### 设计/实现客户端代码main()

思路：添加人名、人与人之间的关系，并调用方法计算距离，并与预期进行对比，看结果是否相同，主函数代码已经提供。

运行结果：

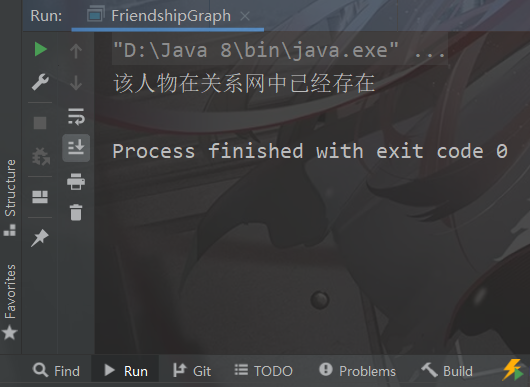


如果将主函数代码的第 10 行注释掉，我的主程序第14-17行应该分别返回：-1，-1,0，-1。如图所示：



如果将主程序第 3 行引号中的“Ross”替换为“Rachel”，我的程序会在终端中打印“”该人物在关系网中已经存在”，并结束程序。

如图：



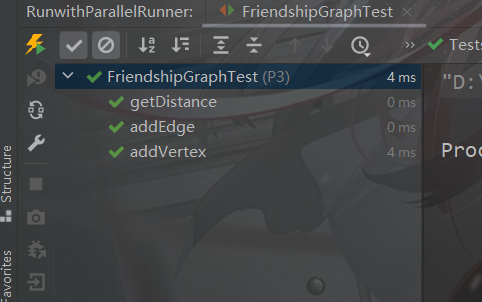
### 设计/实现测试用例

实现思路：

addVertex：向关系网中添加节点，检测是否添加成功

addEdge：给其中两人添加关系，然后判断其中一个的朋友列表中的最后一人是否是另外一个人。

getDistance：设计测试用例检测人与人之间的距离与期望是否相等。



# 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 |
| 2023-03-07 | 20:00-22:00 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按计划完成 |
| 2023-03-08 | 19:00-22:00 | 编写问题1的generateMagicSquare()并完成问题1的相关报告 | 按计划完成 |
| 2023-03-09 | 20:00-22:00 | 完成问题2的克隆代码环节，补全TurtleSoup.java中从drawSquare到calculateBearings的函数的代码 | 按计划完成 |
| 2023-03-13 | 20:00-22:00 | 完成问题2到calculateBearings的相关报告和代码的junit测试，对凸包问题代码进行编写 | 按计划完成 |
| 2023-3-14 | 20:20-22:13 | 完成问题3的Friendshipgragh的编写以及class的编写 | 按计划完成 |
| 2023-3-15 | 19:30-21:30 | 完成问题3的test文件并完成报告 | 按计划完成 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的困难 | 解决途径 |
| 计算凸包问题，没有接触过 | 上网查找资料，借鉴思路，看视频，看各种解决方法并总结 |
| 社交关系网络不知道用什么数据结构保存 | 上网查找资料，询问同学，看教学视频 |
| Java的一些语法操作、git操作不会 | 上网观看教学视频，去csdn等论坛看博客 |

# 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（必答）

经验教训：通过这次实验，我发现我在真正动手写代码的时候，还是十分生疏，对于编译环境的使用也很不熟悉，在进行种种操作时往往要从网上进行搜索，严重降低了自己的任务完成的速度。还有在拿到代码题目的时候，往往没有什么思路，手足无措。

## 针对以下方面的感受（必答）

1. Java编程语言是否对你的口味？与你熟悉的其他编程语言相比，Java有何优势和不足？

是。优势：相比我熟悉的C语言，java的语言十分的简洁优美，能更加直观地让人看懂。不足：Java程序的运行依赖于 Java虚拟机,所以相对于其他语言(比如C)编写的程序慢。

1. 关于Eclipse或IntelliJ IDEA，它们作为IDE的优势和不足；

优势：IDEA的功能十分强大，操作方便，有许许多多的快捷键和扩展功能能让你轻松实现代码。不足：强大的功能性容易造成巨大的依赖性，让人脱离它之后写Java代码变得困难。

1. 关于Git和GitHub，是否感受到了它在版本控制方面的价值；

是。Git和GitHub 在版本控制和开发方面十分方便，能更好地让我们控制、保存电脑上的文件和完成文件的开发

1. 关于CMU和MIT的作业，你有何感受；

他们的作业考虑到了程序开发的方方面面，注重对一个人的全方面培养。需要的知识面十分广，对人的锻炼作用巨大。

1. 关于本实验的工作量、难度、deadline；

本实验的工作量较大，难度适中，deadline较紧

1. 关于初接触“软件构造”课程；

软件构造相对于其他课对于实践更加重视，在理论之外注重我们的编程能力的培养，能够潜移默化地培养一个人的思维方式和习惯，影响深远，作用巨大。