

**2022年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 郑旭然 |
| 学号 | 120L020719 |
| 班号 | 2003005 |
| 电子邮件 | zxrshawn@icloud.com |
| 手机号码 | 16645013218 |

**目录**

[1 实验目标概述 1](#_Toc97648154)

[2 实验环境配置 1](#_Toc97648155)

[3 实验过程 1](#_Toc97648156)

[3.1 Magic Squares 1](#_Toc97648157)

[3.1.1 isLegalMagicSquare() 1](#_Toc97648158)

[3.1.2 generateMagicSquare() 1](#_Toc97648159)

[3.2 Turtle Graphics 1](#_Toc97648160)

[3.2.1 Problem 1: Clone and import 2](#_Toc97648161)

[3.2.2 Problem 3: Turtle graphics and drawSquare 2](#_Toc97648162)

[3.2.3 Problem 5: Drawing polygons 2](#_Toc97648163)

[3.2.4 Problem 6: Calculating Bearings 2](#_Toc97648164)

[3.2.5 Problem 7: Convex Hulls 2](#_Toc97648165)

[3.2.6 Problem 8: Personal art 2](#_Toc97648166)

[3.2.7 Submitting 2](#_Toc97648167)

[3.3 Social Network 2](#_Toc97648168)

[3.3.1 设计/实现FriendshipGraph类 2](#_Toc97648169)

[3.3.2 设计/实现Person类 2](#_Toc97648170)

[3.3.3 设计/实现客户端代码main() 2](#_Toc97648171)

[3.3.4 设计/实现测试用例 3](#_Toc97648172)

[4 实验进度记录 3](#_Toc97648173)

[5 实验过程中遇到的困难与解决途径 3](#_Toc97648174)

[6 实验过程中收获的经验、教训、感想 3](#_Toc97648175)

[6.1 实验过程中收获的经验和教训（必答） 3](#_Toc97648176)

[6.2 针对以下方面的感受（必答） 3](#_Toc97648177)

# 实验目标概述

本次实验一方面训练基本 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。涉及以下几个方面：基本的 Java OO 编程、基于 Eclipse/IDEA 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

JetBrains IntelliJ IDEA：Ultimate 2021.1.3 x64

Git：2.36.0

JDK：11.0.13

在配置过程中由于IDEA和Git已经在电脑上安装并配置和使用过，所以这一部分省略。之前使用高版本JDK，这一次由于实验以及一些其他原因正好将高版本JDK下载，重新安装我们所要求的JDK11。在JDK以及JUnit配置中遇到了如下问题：

JDK在Oracle官网上下载速度非常慢，因此在国内论坛上找了另外的一个JDK 11.0.13源下载。下载后直接配置环境变量：

图形用户界面, 文本, 应用程序, 聊天或短信, 电子邮件

描述已自动生成

测试一下是否安装成功：

文本

描述已自动生成

等等。到此为止JDK安装成功，后续在IDEA选定此JDK即可。

JUnit：之前对Java及相关的工程开发接触不多，经过一番探究我在IDEA里安装了JUnitGenerator V2.0插件：

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

之后如果需要进行单元测试直接使用快捷键，然后选择JUnit4即可。

我的GitHub Lab1仓库的URL地址：

<https://github.com/ComputerScienceHIT/HIT-Lab1-120L020719>

# 实验过程

## Magic Squares

题目首先介绍了幻方的定义，要求我们实现一个Java程序，读取给定的文本文件，检定该方阵每行/每列/每条对角线上数字之和是否为一常数，判断其是否为幻方。

### isLegalMagicSquare()

读取文本文件后检查格式，然后检定矩阵是否为我们想要的MagicSquare。

首先读入文件，创建FileReader、StringBuilder、BufferReader对象，初始化line。

将字符串转换为整型的矩阵进行存储。将readline的字符串分隔开，头尾空格删去，依照要求存储至一个二维数组中，并且时刻要判断当前的数字在之前是否出现过，若出现过则报错，否则正常将其存储至Boolean中去。最后看一下行列是否相等。

接下来是计算任务，计算一下两条对角线的和，接下来每行和每列的和也都要计算。这里总而言之是要看一下所有的结果是否相等，否则有一个不相等的话就要报错。

最后确定是否为幻方即可。

异常处理：我们当然需要FileNotFoundException、IOException、NumberFormatException这些。除此之外我们还自己定义了：

NonSquareExcption

RepeatedNumberException

DifferentSumException

NegativeNumberException

NotAvailableException

这部分在代码中可以展示出来。

流程图：

图示

描述已自动生成

### generateMagicSquare()

这个方法需要生成一个奇数阶幻方。

首先是初始化工作。生成一个空的方阵，初始row=0，col=n/2。

然后开始填充矩阵。将坐标[row,col]位置填充i，之后保证坐标是合法的时候然后进行操作：row--以及col++。‘每次从数组第一行的中间位置开始构造。每当循环次数达到n，列数+1，而如果不满足且row为0时，跳到第n-1行，否则row-1；当列数达到边界后，自动跳到第1列，否则每次循环列数+1。由此

实现目标数组中每个元素的赋值。

最终将结果写在文件中。

流程图：

图示

中度可信度描述已自动生成

异常处理：

ArrayIndexOutOfBoundsException

NegativeArraySizeException

## Turtle Graphics

任务Turtle Graphics需要我们将给定的程序clone后，调用Java标准库使用Turtle Graphics按要求画出图形。

### Problem 1: Clone and import

如何从GitHub获取该任务的代码、在本地创建git仓库、使用git管理本地开发。

在src目录下git bash后git init.

文本

描述已自动生成

git remote add -f origin https://github.com/rainywang/Spring2022\_HITCS\_SC\_Lab1.git

文本

描述已自动生成

git pull origin master

文本

描述已自动生成

成功。

### Problem 3: Turtle graphics and drawSquare

此处需要我们实现画出一定边长的正方形，参数turtle与sidelength。

首先PenColor.*BLACK.*然后我们需要前进sidelength的一个长度，转90度，并且执行4次以上操作，那么就画出了边长为sidelength的正方形。示意图如下：

图形用户界面, 应用程序, Word

描述已自动生成

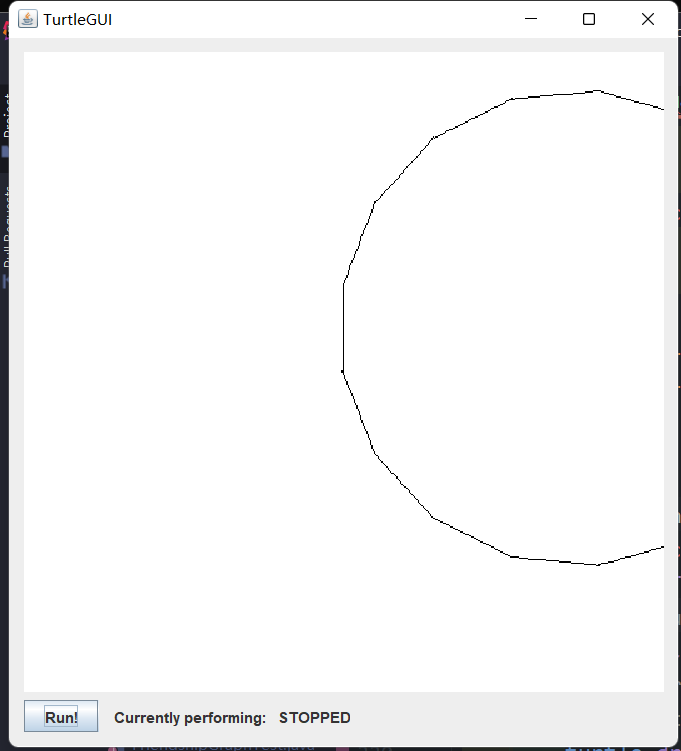
### Problem 5: Drawing polygons

此处需要我们计算正多边形的单角内角度。这其实是：(double) 180.0 - (double) 360.0 / sides。我们调用*calculateRegularPolygonAngle()，然后运行*TurtleSoupTest中的JUnit测试，有以下的结果：

屏幕的截图

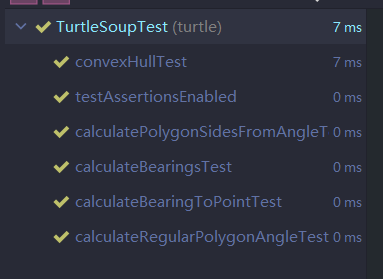
描述已自动生成

我们还需要利用已知的正多边形边数以及边长画一个正多边形。与刚才大致相同地，我们前进sidelength的一个长度，转一个角度（计算出的内角的补角，依然用*calculateRegularPolygonAngle()*）；以上的过程一直重复，次数为边的数量。示意图如下：（这里是正17边形）

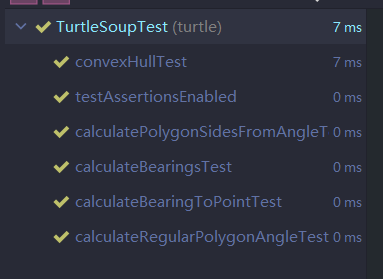


### Problem 6: Calculating Bearings

此处首先是要利用已知的起点和当前的角度，要求求出到达终点所要旋转的角度。首先计算两点之间所连线段在坐标系中的角度，与当前朝向角度作差；这时由于坐标轴上的角度计算都是逆时针的，而turtle旋转是顺时针的，我们给结果取相反数；接下来考虑到turtle计算角度所用坐标系极轴与原坐标系x轴之间相差90度，所以应将结果再与90度作差；最终将结果进行mod 360°的运算。测试如下：



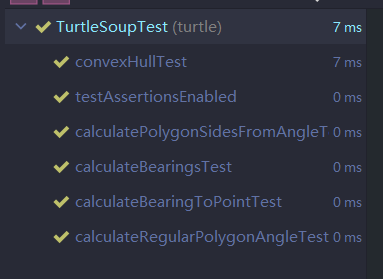
接下来给定了很多个点，需要类似地知道每两个点之间旋转的角度。我们将起点选为(xCoords.get(0), yCoords.get(0))，然后一直循环直到遍历所有的点；每次都将下一个编号的点当作终点并用函数计算旋转角度，存储至List中；将本次的终点当作下一次计算的起点，再次重复以上过程；退出循环后，最终返回List。测试如下：



### Problem 7: Convex Hulls

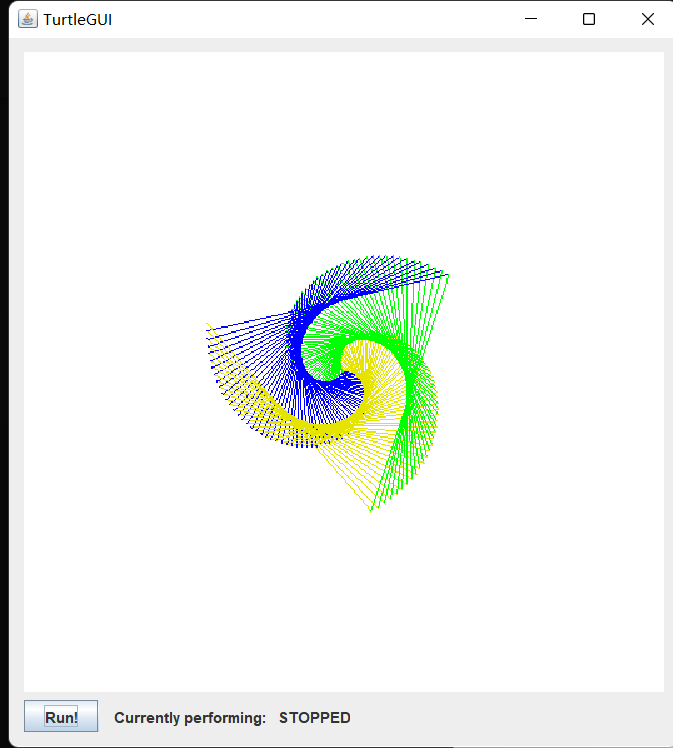
此处是给定了一个非空点集，需要输出凸包上的点，也即凸包上的点必须在拓扑意义上包含其他点。

我们使用推荐的Gift-Wrapping算法。找出一个笛卡尔坐标系下纵坐标最小的点定义为出发点，随后以该点为极点，与x轴平行同方向作极轴，寻找其他所有点中与该出发点连接所成极角最小点作为凸包上的下一个点；按照这样的步骤循环操作，最终一定能够回到最初的起点。这样就形成了该点集的凸包。测试如下：



### Problem 8: Personal art

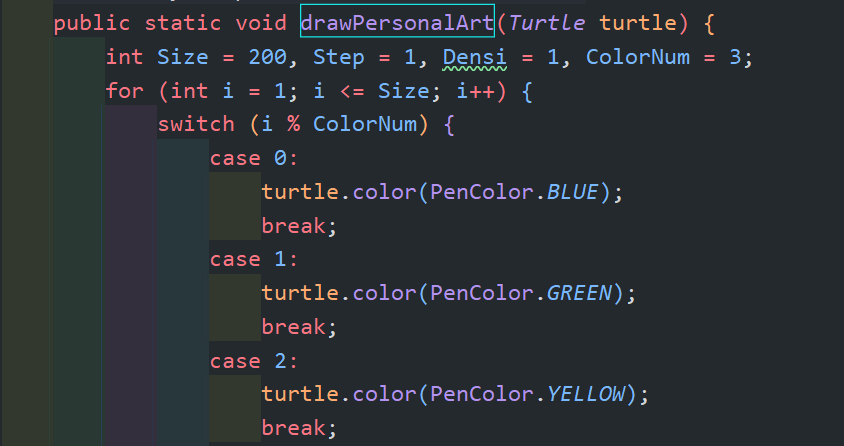
这里实在没想到什么好的图形，我参考了一下网络上的创意，感觉具体的图案其实实现效果都不很好，所以我考虑了一种抽象一点的。



应用程序

低可信度描述已自动生成

具体的实现的话要看代码：



可以调整各种参数。

### Submitting

只需git bash下键入以下命令即可。

文本

描述已自动生成

文本

中度可信度描述已自动生成

注：这里是之后的演示。之前一次提交时忘记截图了。

## Social Network

实现FriendshipGraph.java和Person.java，其中FriendshipGraph有addVertex，addEdge，getDistance；Person类有一个String类型的name成员，之后进行异常处理和测试。

### 设计/实现FriendshipGraph类

首先是addVertex()，接收一个person增加节点。我们这里是list存储，因此只需要将节点加入到存储结点的List中即可。错误处理时要注意，不能出现重复的名字，因此加入节点前应该判断名字是否已出现，并进行相应错误提示。

之后是addEdges()。接受两个Person person，为person\_fir与person\_sec间添加关系。这里注意两种异常：1.不能对自己发起关系；2。是否添加了重复关系。我们调用judgeExist()来判断是否已存在一个关系。如果通过检查则将关系新增至connections中。

GetDistance()中我们采用了BFS算法，在数据结构课程中学习过。使用help queue，当main queue为空时遍历结束。每次检查队列中最前方元素时，应该先将同一深度的所有元素压入help queue，检索完毕后再将他们放回main queue中，这样的话就可以计算distance。我们搜索前还需要先进行参数的判定，参数相等返回0，如果搜索到则返回距离，没有关系的话直接返回-1.

展示：

电脑屏幕截图

描述已自动生成

### 设计/实现Person类

这里很简单，每个Person类中有private final String name,后续可以返回Person的名字。

设计如下：

文本

描述已自动生成

### 设计/实现客户端代码main()

首先完成规定任务，创建节点后使用showMembers()输出所有的成员。创建边后使用showConnections()输出关系，同时支持了有向图的构建。

输出结果：

电脑屏幕截图

描述已自动生成

还要求将Rachel和Ross的关系注释掉：

文本

描述已自动生成

### 设计/实现测试用例

为addVertex()，addEdge()，getDistance()均编写了测试。

如刚才思路所说，addVertex()中测试了是否能成功添加新的成员，能否将重复的关系抛弃掉。

addEdge()中测试是否能成功添加边，是否能抛弃掉指向自身的关系以及重复关系的抛弃。

getDistance()中需要检测距离输出是否合法。无关系返回1，指向自身返回0.

# 实验进度记录

都是用整块时间进行开发的，感觉这样效果会更好。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 04-30 | 早 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按计划完成 |
| 05-01 | 午 | 编写问题 1 的 isLegalMagicSquare 所有需要调用的函数并测试 | 按计划完成 |
| 05-01 | 午晚 | 完成任务2（除了PersonalArt） | 按计划完成 |
| 05-02 | 早午 | 完成任务3 | 按计划完成 |
| 05-03 | 午晚 | 完成问题2的PersonalArt | 按计划完成 |
| 05-06/08 | 晚 | 最终修改完善 | 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 |
| 文件读写的路径问题不是很熟悉 | 查阅相关博客与CSDN |
| Java 队列的使用 | 了解 Java Queue 的用法 |
| 凸包算法 |  |
| Exception in thread "main" java.lang.Error: Unresolved compilation problem，编译器/package不一致 | 检查编译器  打开cmd检查版本  项目名（右键）-> propriety -> Java Compiler  Enable…打勾，Compiler…level 与java和javac一致  可以把Use…勾去掉进行修改  若无误，则非编译器问题  发现package名字有红线\*\*（重要特征）\*\*  发现在build path之后不一致，统一package完成build path请见  解决  修改代码中package名字\*\*（每一个包中的代码）\*\*  再次运行check  注意！！还需要更改同一个package下所有代码的部分import（旧package名） |

# 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）

第一次接触软件构造，github和git也是第一次接触，这样的版本管理让整个开发过程都非常高效，而且之前对Java接触也不多，所以其实这一次解除了很多新东西，给自己的精神上的正向刺激非常的大，感觉非常有趣。

另外的一些感想请见博客：（这部分也是作为课程要求中的要发在自己博客上的内容，所以可能之后更新）

<https://pandalandala.github.io/>

后续将有更多补充。

## 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？

符合，Java的OOP非常顺手，又相近于之前熟悉的C++

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

调试、Junit都很好用，但eclipse感觉界面并不那么美观，IDEA可能更好一点。

1. 关于Git和GitHub，是否感受到了它在版本控制方面的价值；

能掌握Git基本功能，并且确实非常好用。

1. 关于CMU和MIT的作业，你有何感受；

语言问题，有些地方理解比较困难

1. 关于本实验的工作量、难度、deadline；

工作量非常大，难度很大，Deadline适中，不过最近课多

1. 关于初接触“软件构造”课程；

非常奇妙。这是我第一次接触系统的软件开发，感觉很有意思。

注：关于课程的感受感想以及理解时刻增加着，因此报告会有补充。

博客：<https://pandalandala.github.io/>