

**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 | 罗瑞欣 |
| 学号 | 1170300821 |
| 班号 | 1703008 |
| 电子邮件 | 310295709@qq.com |
| 手机号码 | 15048260039 |

**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

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。

⚫ 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

Windos10专业版，eclipse，jdk-9.01，Junit5.

环境是好久好久以前配置的。。。困难主要是在环境变量的path设置上，一开始看了几个网上的教程配好了，后来再配的时候看懂了原理。

仓库地址：git@github.com:ComputerScienceHIT/Lab1-1170300821.git

# 实验过程

## Magic Squares

问题理解：本问题要求实现判断一个nxn矩阵是否是一个幻方，并且理解一个针对n为奇数的幻方的生成算法。主要考察点：Java基本语法，文件读写，判断输入的合法性。

### isLegalMagicSquare()

分三次读取文件，用

BufferedReader bReader2 **=** **new** BufferedReader**(new** FileReader**(**file**));**

分别计算行数，检查行数=列数，将string转化为int。

对于每行使用.split("\t");进行分割，对于分割之后的每一项，如果不能使用Integer.valueOf成功转化，这有两种可能原因，一种是该矩阵不满足使用\t分隔，一种是该项包含非数字。

对于非法数据类型，用.contains(".")和.contains("-")检查。

使用三次for循环分别检查行、列、对角线数字的和是否相等。

在 try-catch 语句内完成上述设计，发生Exception e错误时打印System.out.println("file error!");

### generateMagicSquare()

思路：使用罗伯法构造一个n为奇数的nxn幻方。罗伯法的算法为

把1（或最小的数）放在第一行正中；按以下规律排列剩下的(n×n－1)个数：

（1）每一个数放在前一个数的右上一格；

（2）如果这个数所要放的格已经超出了顶行那么就把它放在底行，仍然要放在右一列；

（3）如果这个数所要放的格已经超出了最右列那么就把它放在最左列，仍然要放在上一行；

（4）如果这个数所要放的格已经超出了顶行且超出了最右列，那么就把它放在底行且最左列；

（5）如果这个数所要放的格已经有数填入，那么就把它放在前一个数的下一行同一列的格内。

## Turtle Graphics

任务理解：实现一个完整的绘制工具Trurtle Graphics。主要考点：计算几何基础知识，正多边形的内角外角，java函数调用绘制多边形，计算向量之间的夹角，计算凸包，使用Junit进行单元测试。

### Problem 1: Clone and import

从本地创建git仓库：

1.git init初始化本地仓库

2.git add remote origin 添加远程库源

3.在远程仓库创建master分支

4.git pull origin master将远程仓库同步到本地

5.git add->git commit->git push 将本地文件加入本地仓库，将本地仓库同步到远程仓库

### Problem 3: Turtle graphics and drawSquare

根据turtle.java中提供的方法，实现在GUI中画出一个正方形。根DrawableTurtle.java中的定义，forward (steps)根据当前坐标和朝向计算step步之后的坐标，生成线段类lineSeg添加到ActionList。turn(degrees)更改当前朝向，向ActionList中添加turn的Action。

### Problem 5: Drawing polygons

调用calculateRegularPolygonAngle计算正多边形内角。根据正多边形内角=（边数-2）/sides。

### Problem 6: Calculating Bearings

计算从一个点到下一个点，依次需要转过的角度。先计算第一个到第二个点和y轴之间的夹角，然后通过for循环，计算以result中上一个角为准，到下一个点转动的角度。

### Problem 7: Convex Hulls

采用gift-wrapping算法实现convexHull

（1）先找出最左下的点A，加入返回的结果result中

（2）利用方法calculateBearingToPoint，依次计算从上一个加入result的点和所有未加入点（除A外）的直线和y轴之间的夹角，将夹角最小的点标记为可加入。

（3）如果标记的点的夹角，与上一个加入的点的夹角相同，则去除上一个点。

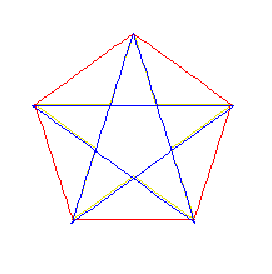
（4）将标记的点加入result

（5）直到A点被第二次标记，算法结束。

### Problem 8: Personal art



运行结果



### Submitting

Git push提交代码。

## Social Network

实现Person和FriendshipGraph两个类，模拟社交网络，提供添加节点（人），节点之间添加边（社交关系），通过BFS计算两节点之间最短路的功能。主要考点：java类的实现，java基本数据结构，BFS求最短路。

### 设计/实现FriendshipGraph类

使用List<Person> name存储所有节点。int[][]graph以邻接矩阵记录边。

addVertex：添加节点，向persons List中添加一个person对象。

addEdge： 添加边，计算两人下标，graph数组对应位置改为1 。

getDistance：利用BFS计算两人间最短路径。

### 设计/实现Person类

String name 储存人名。

SetName(String name) 设置名字

String GetName() 返回储存的名字

Person(String string) 传参并调用SetName设置名字

### 设计/实现客户端代码main()

无。

### 设计/实现测试用例

测试单边联通的特殊情况，测试重名的情况

## Tweet Tweet

### Problem 1: Extracting data from tweets

getTimespan：循环比较时间，调用Time.isBefore和Time.isAfter函数得到最早最晚时间。

getMentionedUsers：使用正则表达式"@([\\w,-]+)"匹配字符串，为了保证前面满足为非法字符，还需要对于匹配项的前一个字符进行特殊判断。

getMentionedHashTages：使用正则表达式"#([\\w,-]+)"匹配字符串，为了保证前面满足为非法字符，还需要对于匹配项的前一个字符进行特殊判断。

### Problem 2: Filtering lists of tweets

writtenBy：for循环检查所有tweet，比较author和username在lowercase下是否相同。

inTimespan：利用Time.isBefore和Time.isAfter进行比较。

containing：循环调用每一条tweet的text，调用String.split(“ ”)进行分割，然后检查words的关键字是否存在于分割后的单词中。

### Problem 3: Inferring a social network

guessFollowGraph：扫描每条tweet，调用Extract.getMentionedUsers提取出该条tweet中所有被@到的人，将这些人加入到Map[author]->Set<Stirng>中。

influencers：按照社交网络的影响力（follow数目）对所有user进行排名。

### Problem 4: Get smarter

getMentionedHashTages：使用正则表达式"#([\\w,-]+)"匹配字符串，为了保证前面满足为非法字符，还需要对于匹配项的前一个字符进行特殊判断。

# 实验进度记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 2019.2.25 | 19:00-22:00 | P1 MagicSquare | 完成 |
| 2019.3.2 | 9:00-15:00 | P2 | 完成 |
| 2019.3.3 | 9:00-17:00 | P3+P4 | 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 |
| 凸包问题难以实现，无法用比较普通的方法实现某些环节的功能 | 舍友指导 |
|  |  |
|  |  |

# 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

很多很多Java的标准库的基本用法，尤其是set、map、list等等

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

好是挺好，就是口味比较重

1. 关于Eclipse IDE

界面比IDEA丑，其他都挺好，尤其内存占用不是那么过分

1. 关于Git和GitHub

吼啊

1. 关于CMU和MIT的作业

功夫到家

1. 关于本实验的工作量、难度、deadline

难度适中

1. 关于初接触“软件构造”课程

吼啊