

**2022年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 马翊轩 |
| 学号 | 120L021002 |
| 班号 | 2003001 |
| 电子邮件 | [1041547307@qq.com](mailto:1041547307@qq.com) |
| 手机号码 | 13931181219 |

**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

首先在lab0的实验报告中根据给出的网址下载安装了ecilpse（后来改用IDEA），通过 ToolBox 安装IntellJ IDEA Ultimate 2021.1

在git官网下载安装了git-bash。由于git的操作并不熟悉，所以在csdn上查找了关于用git的操作与github进行联系的一些介绍，详见参考文章：<https://blog.csdn.net/qq_37512323/article/details/80693445>

然后下载了jdk18，由于jdk18支持jdk11所有功能，因此在IDE里设置以jdk11编译运行。

下载了junit4.13和hamcrest-core-1.3的jar包。

在这里给出你的GitHub Lab1仓库的URL地址。

# 实验过程

请仔细对照实验手册，针对四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

任务一：能够从五个已经提供的txt文档中读入数据，来判断输入的数据能否构成一个MagicSquare，若是则返回true，若不是则返回false且说明原因。MagicSquare即幻方，是一种将数字安排在正方形格子中，使每行、列和对角线上的数字和都相等的方法。

任务二：对给出的generateMagicSquare函数进行扩充，使之能够产生一个MagicSquare并且将其输入到文本6.txt中，然后对其判断是否为MagicSquare。并且满足若输入的为奇数能产生MagicSquare，而输入偶数或者负数会返回false。

### isLegalMagicSquare()

本题要求写一个函数判断txt文件中保存的矩阵是不是符合幻方要求的矩阵。输入参数为文件路径。返回布尔值，检验是否为符合条件的 Magic Square。

首先在于文件的读取，利用Bufferedreader，读入一行后进行分割。

图形用户界面, 文本

描述已自动生成

每一行读取后每行按照 \t 进行分割。

文本

描述已自动生成

随后尝试转化成数字。如果转化过程抛出异常，则可以认为数字不合法。

文本

描述已自动生成

如果分割符问题，那么会造成数目不同，也比较好判断。因此分割完成后转换过程进行判断是否成功转换，转换失败可以认为数字非法，转换完成后以第一行作为标准，判断后面每行是否长度相等。之后比较列数等等如果没有问题，再判断横竖和对角线即可。

### generateMagicSquare()

此部分报告中已经给了一个例子代码。以那个为例，所需要进行的改动主要是异常情况的判断以及文件的写出。

思路：首先计算初始位置（0，n/2），赋值为1，之后每次取当前位置的右上角的位置，设置的值每次加1，如果当前行是第一行，则下一行为最后一行，如果当前列是最右边的列，则下一次取左边第一列，如此重复平方次，就对整个矩阵赋值了，而且满足了每行每列以及对角线之和都是相同数字的效果。

原理：关于这种幻方，曾见到过一个生成的口诀：一居上行正中央，依次斜填切莫忘，上出框界往下写，右出框时左边放，重复便在下格填，出角重复一个样。

异常情况判断：

文本

描述已自动生成

写入6.txt：

文本

描述已自动生成

## Turtle Graphics

包括绘图（按照要求绘图以及自行绘制喜好图案）、一些基本的计算（比如角度等等），和一个凸包覆盖的问题。熟悉turtle Graphics的各种函数接口，调用设置好的函数，实现作图，使用.Math库的几个函数实现一些计算。

### Problem 1: Clone and import

熟悉从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。

git init 初始化仓库

touch .gitignore 创建 ignore 文件，并且后面编辑该文件

git add 将当前目录暂存

git commit -m “idea init”

git remote add origin git@github.com:ComputerScienceHIT/Lab1-120L021002.git

git push origin master --force 强制覆盖远程分支

### Problem 3: Turtle graphics and drawSquare

绘制正方形，按照规则每次走一定长度，然后向右旋转90度即可

### Problem 5: Drawing polygons

利用数学公式编写calculateRegularPolygonAngle (int sides)计算n边形的内角：

180．0-360.0/sides;

随后重复sides次前进转向即可

文本

描述已自动生成

### Problem 6: Calculating Bearings

利用atan2函数计算前进方向与x轴正向夹角，之后再把该角度转换成和y正向的夹角，

注意如果出现负数要加上360度，如果超过360度要减去360度。

文本

描述已自动生成

### Problem 7: Convex Hulls

计算给定集合中的凸包需要利用Gift wrapping algorithm算法，每次选择转向角最小的点（同时距离最长的点），加入集合，求转向角函数直接使用的之前写好的calculateBearingToPoint函数，其中注意点是相同转向角额点的取舍，在循环中设置一个targetdis变量记录当前目标点的距离，如果之后出现了新的目标点，就用targetdis和计算得到的当前点的距离tempdis来比较进行取舍。

### Problem 8: Personal art

利用for循环逐层持续画出，利用整除关系获得层次和渐变色彩效果。代码如下：

文本

描述已自动生成

结果如下：

图片包含 直方图

描述已自动生成

### Submitting

git add src/P2

git push origin master

## Social Network

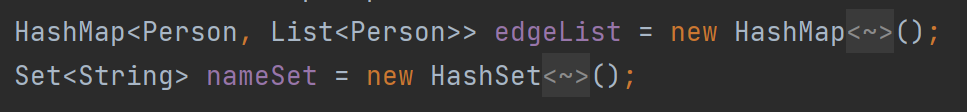
建立一张人际关系网络，并写一个计算人际关系“距离”的函数。抽象来看的话，就是一个有向图中（无向图按照两个方向都有的有向图）求解任意指定的两个点的距离，其中每条边的距离均视为1，之后通过 BFS 求解最段距离。

。

### 设计/实现FriendshipGraph类

**思路**：在FriendshipGraph类中我们要完成的功能为：增加一个人的对象addVertex（若是增加的人为已存在的则输出“此名已存在，重复”并结束程序），增加一条有向边addEdge，得到两个人之间的最短距离getDistance。对此三个功能的实现，主要是功能三的实现相对复杂，我们决定采取广度优先搜索（BFS）的方式求最短路径。

**过程：**根据题目要求，有 addVertix, addEdge, getDistance 三个函数。对于存储数据，通过 HashMap 记录边， HashMap<Person, List<Preson>> 的方式存储某个人及其认识的人。而为了实现不重复，采用了 Set<String> 的方式，每次插入人员的时候。判断要插入的是否在这个集合内部。后面每次插入边的时候，首先判断两个人是否都已经记录了，如果都已经记录了，则再判断是否已经存在这条边。



文本

描述已自动生成

对于获取距离，则是使用标准的 BFS 的方法，通过队列实现。

文本

描述已自动生成

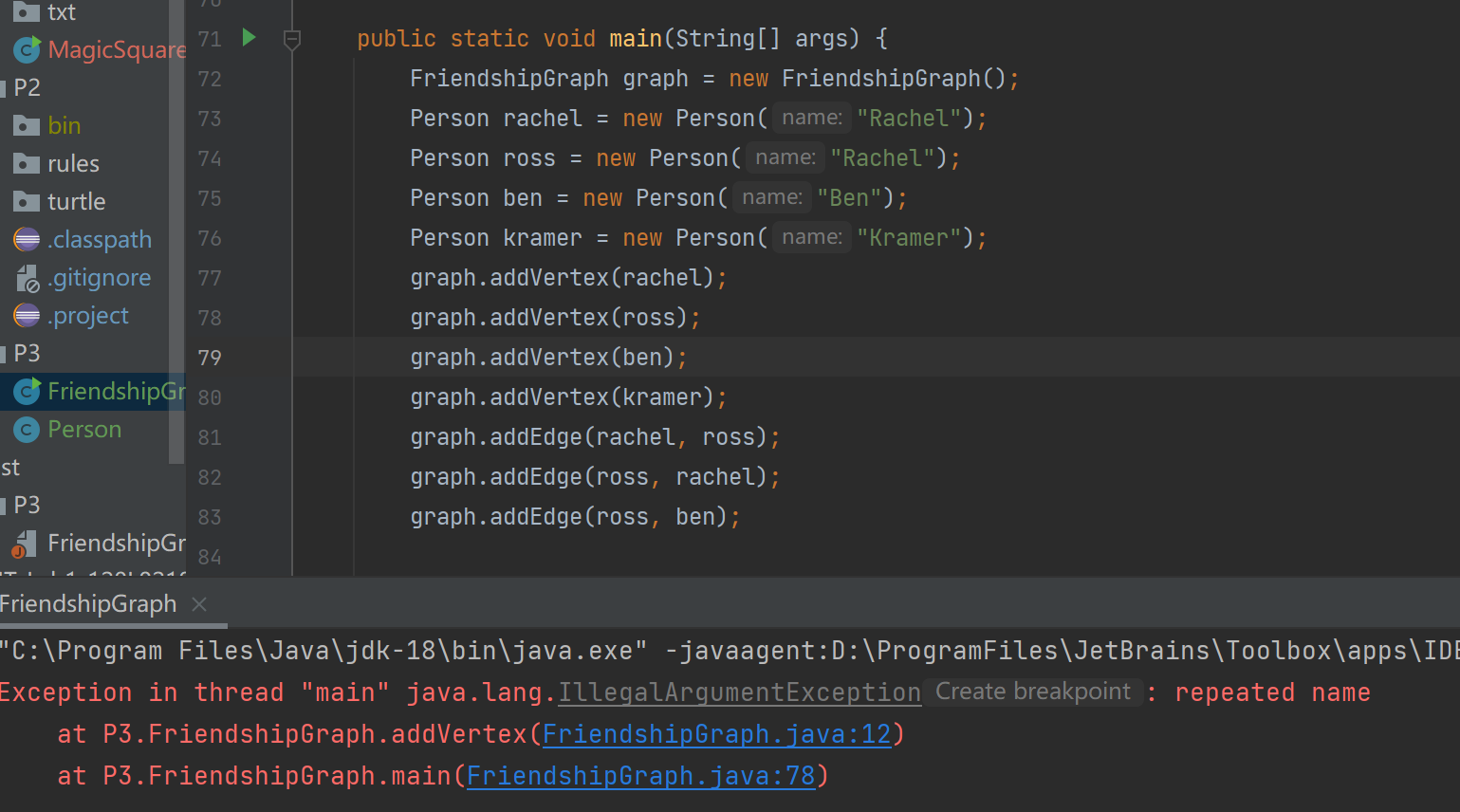
**结果：**

如果将上述代码的第10 行注释掉（意即rachel 和ross 之间只存在单向的

社交关系 ross->rachel）：

文本

描述已自动生成如果将第 3 行引号中的“Ross”替换为“Rachel”：抛出异常



### 设计/实现Person类

对于一个人，其标识为姓名，通过构造函数的方式进行写入，单独提供了一个函数 getNanme() 进行读取。

文本

中度可信度描述已自动生成

### 设计/实现客户端代码main()

按题目要求，稍加修改复制粘贴即可。

文本

描述已自动生成

### 设计/实现测试用例

**思路**：测试部分，参考 P2 测试的方式。

第一个测试点，判断测试是否可用

第二个测试点，testAll 是无向图（与 main() 实现相同）

第三个测试点，testAll2 为注释某条添加边的代码后的测试，即有向图

第四个测试点，插入两个名字相同的人，判断是否失败

第五个测试点，插入同一个人两次，判断是否失败

第六个测试点，插入一个未提含有未未AddVertix 的人的边，判断是否失败

第七个测试点，插入两次同一条边，判断是否失败

文本

描述已自动生成 文本

描述已自动生成 文本

描述已自动生成

# 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 |
| 2021-04-29 | 20:00~12:00 | 安装调整环境 | 按计划完成 |
| 2021-05-01 | 15:00~21:00 | 编写测试P1 | 正常完成，未commit |
| 2021-05-03 | 16:00~21:00 | P1 细节部分修改，P2 主要内容 | 正常完成，未commit |
| 2021-05-07 | 16：00-20：00 | P3 主要内容，以及 P1 P2 部分修改 | 正常完成，未commit |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的困难 | 解决途径 |
| Git不会使用， | 上网查找Git教程  访问教程“廖雪峰的官方网站” |
| Java 语言问题 | 查阅参考文档 |
|  |  |

# 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）

1.初步了解 ava的众多语法；

2.学到了git的使用、junit的编写；

3.对java面向对象的特质有了更深的理解。

## 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？

之前未接触过 Java，最初较为陌生。

1. 要求相对于其他语言来说更为严格，比如异常处理、数据类型匹配等

2. java.util.\* 内的数据结构面向对象特性很强，比如抽象类和具体实现的分离，这个在其他语言不常见（每次想用一个东西都是要先找到可以实现的数据结构）

3. 很多数据结构不采用通用的方法名称，比如 ArrayList.add()，Queue.add()/offer()，queue.peek()，queue.poll() 等（一般对于数组可能是 append, insert 或者 push，对于一个队列，更常见的是 push, front 和 pop）

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

使用的是 IntellJ IDEA有其他 Jetbrains 软件经验，差别不大。自动补全和纠错功能强大，对初学者较为友好。

1. 关于Git和GitHub，是否感受到了它在版本控制方面的价值；

进一步理解“版本控制”，但是由于是单人完成，有部分功能尚未感受到。

1. 关于CMU和MIT的作业，你有何感受；

对初学者入门java很有帮助

1. 关于本实验的工作量、难度、deadline；

工作量、难度适中，deadline 冲突较为严重，部分内容完成较为仓促。

作为有其他语言接触没有学过 Java 的人来说，难度和工作量的话，除了开始大量文件操作比较麻烦之外（java各种 File, buffered, stream），其他的比较适合 java 入门。基本上用到了不少相关的特性，三个题下来能够熟悉基本的 java 特性。

1. 关于初接触“软件构造”课程；
2. 关于 java 和 git.

之前没有接触过，短时间上手比较难，比如 java 的包管理， git 的一些概念等。

1. 时间记录的表格

初步体会到大型项目开发的流程。