



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

2018 年春季学期

计算机学院大二软件构造课程

Lab 1 实验报告

姓名	肖一诺
学号	1160300222
班号	1603002
电子邮件	m18846183092@163.com
手机号码	18846183092

目录

1 实验目标概述	1
2 实验环境配置	1
3 实验过程	1
3.1 Magic Squares	1
3.1.1 isLegalMagicSquare()	2
3.1.2 generateMagicSquare()	3
3.2 Turtle Graphics	3
3.2.1 Problem 1: Clone and import	4
3.2.2 Problem 3: Turtle graphics and drawSquare	4
3.2.3 Problem 5: Drawing polygons	4
3.2.4 Problem 6: Calculating headings	5
3.2.5 Problem 7: Personal art	5
3.2.6 Submitting	5
3.3 Social Network	5
3.3.1 设计/实现 FriendshipGraph 类	5
3.3.2 设计/实现 Person 类	5
3.3.3 设计/实现客户端代码 main()	5
3.3.4 设计/实现测试用例	6
3.4 Tweet Tweet (选作, 额外记分)	错误!未定义书签。
4 实验进度记录	6
5 实验过程中遇到的困难与解决途径	6
6 实验过程中收获的经验、教训、感想	6

实验目标概述

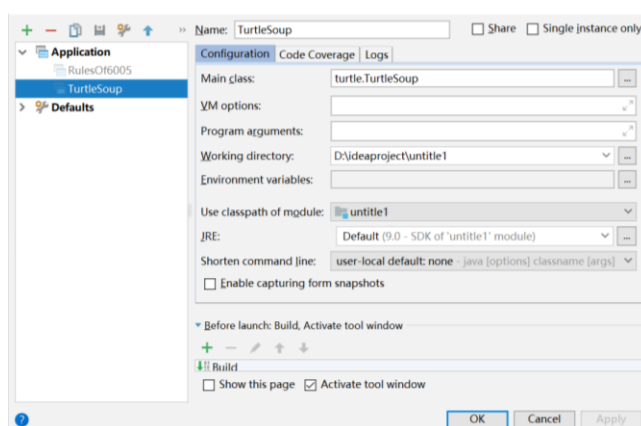
本次实验通过求解四个问题（其中一个可选），训练基本 Java 编程技能，能够利用 Java OO 开发基本的功能模块，能够阅读理解已有代码框架并根据功能需求补全代码，能够为所开发的代码编写基本的测试程序并完成测试，初步保证所开发代码的正确性。另一方面，利用 Git 作为代码配置管理的工具，学会 Git 的基本使用方法。

1 实验环境配置

1. 到官网上下载最新版本 java

网址：<https://www.java.com>

2. 在 IntelliJ 中配置 java



如图所示为实验一第二个项目的 configuration

- 在这里给出你的 GitHub Lab1 仓库的 URL 地址（Lab1-学号）。

<https://github.com/ComputerScienceHIT/Lab1-1160300222.git>

2 实验过程

2.1 Magic Squares

幻方问题。检测一个方形矩阵是否为幻方。幻方就是每一行，每一列，以及对角线

上元素的和值全部相等的方阵。其中, 如果方阵中存在浮点数, 或者幻方中数字数量不符合要求, 或者数字没有按照制表符分隔, 就会出现错误。

2.1.1 isLegalMagicSquare()

1. 首先判断矩阵中是否存在空格。如果存在空格直接报错返回。方法是从文件中读入一行, 之后调用字符串的 `indexOf` 方法判断有无空格。

2. 接下来从文件中读入数字, 通过如下代码:

```
FileInputStream fileInputStream=new FileInputStream(filename);
```

```
Scanner scanner=new Scanner(fileInputStream);
```

之后调用 `scanner.nextInt` 读入一个整数。注意, 这里如果发现读入了浮点数, 系统一定会抛出一个异常。利用 `try catch` 进行捕捉, 然后控制台打印输出。

3. 记入读入的第一行中元素的数量, 在计入总的整数数量, 如果二者不是平方的关系, 抛出一个异常。

4. 接下来判断每一行, 每一列以及对角线上元素的数值之和是否相等。

接下来显示函数调用结果:

输入 1:



```
13 @Test
14 public void testfile10 {
15     boolean result=magicSquares.isLegalMagicSquare(filename: "1.txt");
16     // System.out.println(magicSquares.isLegalMagicSquare("1.txt"));
17     Assert.assertEquals( expected: true, result);
18 }
```

testMagicSquares

1 test passed - 29ms

"C:\Program Files\Java\jdk-9.0.4\bin\java" -Didea.test.cyclic.buffer.size=1048576 "-javaagent:D:\IntelliJ IDEA Community Edition 2017.3.4\lib\idea_rt.jar=56211:"

Process finished with exit code 0

输入 2:



```
19 @Test
20 public void testfile20 {
21     // MagicSquares magicSquares=new MagicSquares();
22     boolean result=magicSquares.isLegalMagicSquare(filename: "2.txt");
23     Assert.assertEquals( expected: true, result);
24 }
```

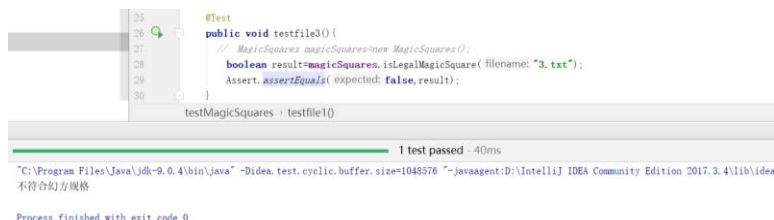
testMagicSquares > testfile10

1 test passed - 26ms

"C:\Program Files\Java\jdk-9.0.4\bin\java" -Didea.test.cyclic.buffer.size=1048576 "-javaagent:D:\IntelliJ IDEA Community Edition 2017.3.4\lib\idea_rt.jar=56211:"

Process finished with exit code 0

输入 3:



```
25 @Test
26 public void testfile30 {
27     MagicSquares magicSquares=new MagicSquares();
28     boolean result=magicSquares.isLegalMagicSquare(filename: "3.txt");
29     Assert.assertEquals( expected: false, result);
30 }
```

testMagicSquares > testfile10

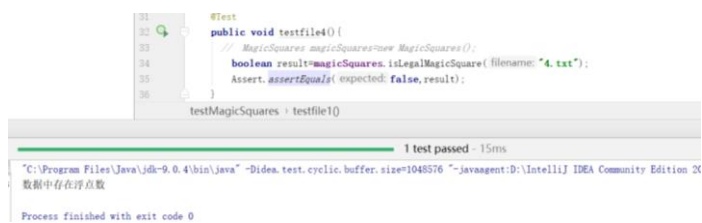
1 test passed - 40ms

"C:\Program Files\Java\jdk-9.0.4\bin\java" -Didea.test.cyclic.buffer.size=1048576 "-javaagent:D:\IntelliJ IDEA Community Edition 2017.3.4\lib\idea_rt.jar=56211:"

不符合幻方规格

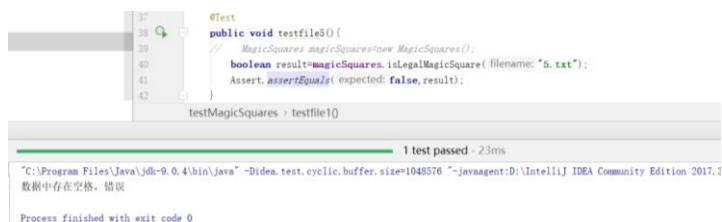
Process finished with exit code 0

输入 4:



```
31 @Test
32 public void testfile4() {
33     // MagicSquares magicSquares=new MagicSquares();
34     boolean result=magicSquares.isLegalMagicSquare( filename: "4.txt");
35     Assert.assertEquals( expected: false,result);
36 }
testMagicSquares : testfile4()
1 test passed - 15ms
"C:\Program Files\Java\jdk-9.0.4\bin\java" -Didea.test.cyclic.buffer.size=1048576 "-javaagent:D:\IntelliJ IDEA Community Edition 2017.1\lib\idea_rt.jar=1048576:D:\IntelliJ IDEA Community Edition 2017.1\bin" -jar D:\IntelliJ IDEA Community Edition 2017.1\bin\idea_rt.jar 1048576
数据中存在浮点数
Process finished with exit code 0
```

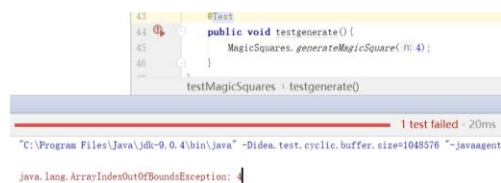
输入 5:



```
37 @Test
38 public void testfile5() {
39     // MagicSquares magicSquares=new MagicSquares();
40     boolean result=magicSquares.isLegalMagicSquare( filename: "5.txt");
41     Assert.assertEquals( expected: false,result);
42 }
testMagicSquares : testfile5()
1 test passed - 23ms
"C:\Program Files\Java\jdk-9.0.4\bin\java" -Didea.test.cyclic.buffer.size=1048576 "-javaagent:D:\IntelliJ IDEA Community Edition 2017.1\lib\idea_rt.jar=1048576:D:\IntelliJ IDEA Community Edition 2017.1\bin" -jar D:\IntelliJ IDEA Community Edition 2017.1\bin\idea_rt.jar 1048576
数据中存在空格, 错误
Process finished with exit code 0
```

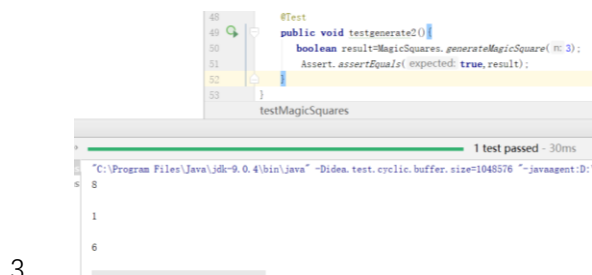
2.1.2 generateMagicSquare()

1. 如果传入参数为偶数, 那么结果会抛出异常, 如图所示:



```
43 @Test
44 public void testgenerate() {
45     MagicSquares.generateMagicSquare( 4);
46 }
testMagicSquares : testgenerate()
1 test failed - 20ms
"C:\Program Files\Java\jdk-9.0.4\bin\java" -Didea.test.cyclic.buffer.size=1048576 "-javaagent:D:\IntelliJ IDEA Community Edition 2017.1\lib\idea_rt.jar=1048576:D:\IntelliJ IDEA Community Edition 2017.1\bin" -jar D:\IntelliJ IDEA Community Edition 2017.1\bin\idea_rt.jar 1048576
java.lang.ArrayIndexOutOfBoundsException: 4
Process finished with exit code 1
```

2. 如果传入参数为奇数, 那么不会产生异常, 并产生一系列输出。如图所示:



```
48 @Test
49 public void testgenerate2() {
50     boolean result=MagicSquares.generateMagicSquare( 3);
51     Assert.assertEquals( expected: true,result);
52 }
testMagicSquares : testgenerate2()
1 test passed - 30ms
"C:\Program Files\Java\jdk-9.0.4\bin\java" -Didea.test.cyclic.buffer.size=1048576 "-javaagent:D:\IntelliJ IDEA Community Edition 2017.1\lib\idea_rt.jar=1048576:D:\IntelliJ IDEA Community Edition 2017.1\bin" -jar D:\IntelliJ IDEA Community Edition 2017.1\bin\idea_rt.jar 1048576
8
1
6
Process finished with exit code 0
```

- 3.

2.2 Turtle Graphics

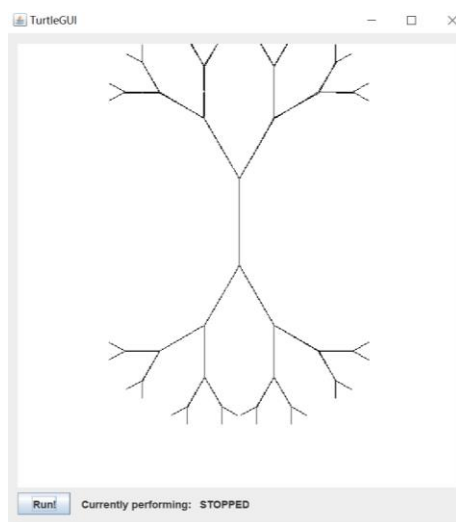
这是一个画图程序, 通过完成几个函数来用 turtle 画出自己的图形。

1 函数 calculateRegularPolygonAngle(int sides): 通过给出正多边形的边, 计算正多边形的各个内角的大小。

2 . drawSquare(Turtle turtle, int sideLength): 通过给出边的长度, 画出一个正方形。

3. `calculatePolygonSidesFromAngle(double angle)`: 通过给出内角的大小, 计算出是几边形。
4. `drawRegularPolygon(Turtle turtle, int sides, int sideLength)`: 画出给定边长的多边形。
5. `calculateHeadingToPoint`: 计算从当前位置向目标位置前进需要转过的角度。
6. `calculateHeadings`: 参数为两个列表, 通过列表给出的位置计算应当转过的角度。

我画的图是一个“二叉树”, 画了一个树形结构, 递归调用函数得出图形。



2.2.1 Problem 1: Clone and import

新建工程, 并将 `rules` 和 `turtle` 文件夹导入到工程中。

2.2.2 Problem 3: Turtle graphics and drawSquare

见前页

2.2.3 Problem 5: Drawing polygons

见前页

2.2.4 Problem 6: Calculating headings

见前页

2.2.5 Problem 7: Personal art

见前页

2.2.6 Submitting

1. `git init`
2. `git add .`
3. `git remote add origin https://github.com/ComputerScienceHIT/Lab1-1160300222`
4. `git push -u origin master`

通过以上步骤成功将项目上传到 github 上

2.3 Social Network

通过构件一个有向图的数据结构，实现社交网络。每一个人相当于一个节点，节点之间如果有一条有向边，例如 A 指向 B，则说明 A 对 B 满足关系。

2.3.1 设计/实现 FriendshipGraph 类

首先定义一个接口，接口为 IGraph，期中定义了几个应该被实现的函数，比如添加边，添加节点等等。之后定义 FriendshipGraph 实现这个接口。

2.3.2 设计/实现 Person 类

有一个 name 私有变量。

2.3.3 设计/实现客户端代码 main()

见代码注释。

2.3.4 设计/实现测试用例

在图形网络中添加几个人，名字记为“xiao”,“han”,“Tom”,“je”,“tim”。随机取两个人，找出两个人之间的最短路径。

3 实验进度记录

忘记了

4 实验过程中遇到的困难与解决途径

有向图的广度优先搜索重新学习了一遍。

5 实验过程中收获的经验、教训、感想

(1) Java 编程语言是否对你的口味？

不错

(2) 关于 Eclipse IDE

不喜欢，用 intelij

(3) 关于 Git

命令太多，记不过来

(4) 关于 CMU 和 MIT 的作业

不喜欢

(5) 关于本实验的工作量、难度、deadline

无所谓

(6) 关于初接触“软件构造”课程

不知道上课讲的东西有什么用。