

**2020年春季学期  
计算机学院《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | cycleke |
| 学号 | xxxxxxxxxx |
| 班号 | xxxxxxx |
| 电子邮件 | [cycleke@gmail.com](mailto:cycleke@gmail.com) |
| 手机号码 | xxxxxxxxxxx |

**目录**

[1 实验目标概述 1](#_Toc35039171)

[2 实验环境配置 1](#_Toc35039172)

[2.1 实验环境配置 1](#_Toc35039173)

[2.2 GitHub Lab1地址 3](#_Toc35039174)

[3 实验过程 3](#_Toc35039175)

[3.1 Magic Squares 3](#_Toc35039176)

[3.1.1 isLegalMagicSquare() 4](#_Toc35039177)

[3.1.2 generateMagicSquare() 4](#_Toc35039178)

[3.2 Turtle Graphics 6](#_Toc35039179)

[3.2.1 Problem 1: Clone and import 6](#_Toc35039180)

[3.2.2 Problem 3: Turtle graphics and drawSquare 7](#_Toc35039181)

[3.2.3 Problem 5: Drawing polygons 7](#_Toc35039182)

[3.2.4 Problem 6: Calculating Bearings 7](#_Toc35039183)

[3.2.5 Problem 7: Convex Hulls 8](#_Toc35039184)

[3.2.6 Problem 8: Personal art 8](#_Toc35039185)

[3.2.7 Submitting 9](#_Toc35039186)

[3.3 Social Network 9](#_Toc35039187)

[3.3.1 设计/实现FriendshipGraph类 10](#_Toc35039188)

[3.3.2 设计/实现Person类 11](#_Toc35039189)

[3.3.3 设计/实现客户端代码main() 11](#_Toc35039190)

[3.3.4 设计/实现测试用例 12](#_Toc35039191)

[4 实验进度记录 13](#_Toc35039192)

[5 实验过程中遇到的困难与解决途径 13](#_Toc35039193)

[6 实验过程中收获的经验、教训、感想 14](#_Toc35039194)

[6.1 实验过程中收获的经验和教训 14](#_Toc35039195)

[6.2 针对以下方面的感受 14](#_Toc35039196)

# 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

## 实验环境配置

1. 安装JDK、Eclipse、Git

在Arch Linux下，在终端中输入sudo pacman -S jdk8-openjdk eclipse-java git即可完成安装。



图 2‑1安装工具

1. 配置Eclipse

将项目导入Eclipse，进而完成实验

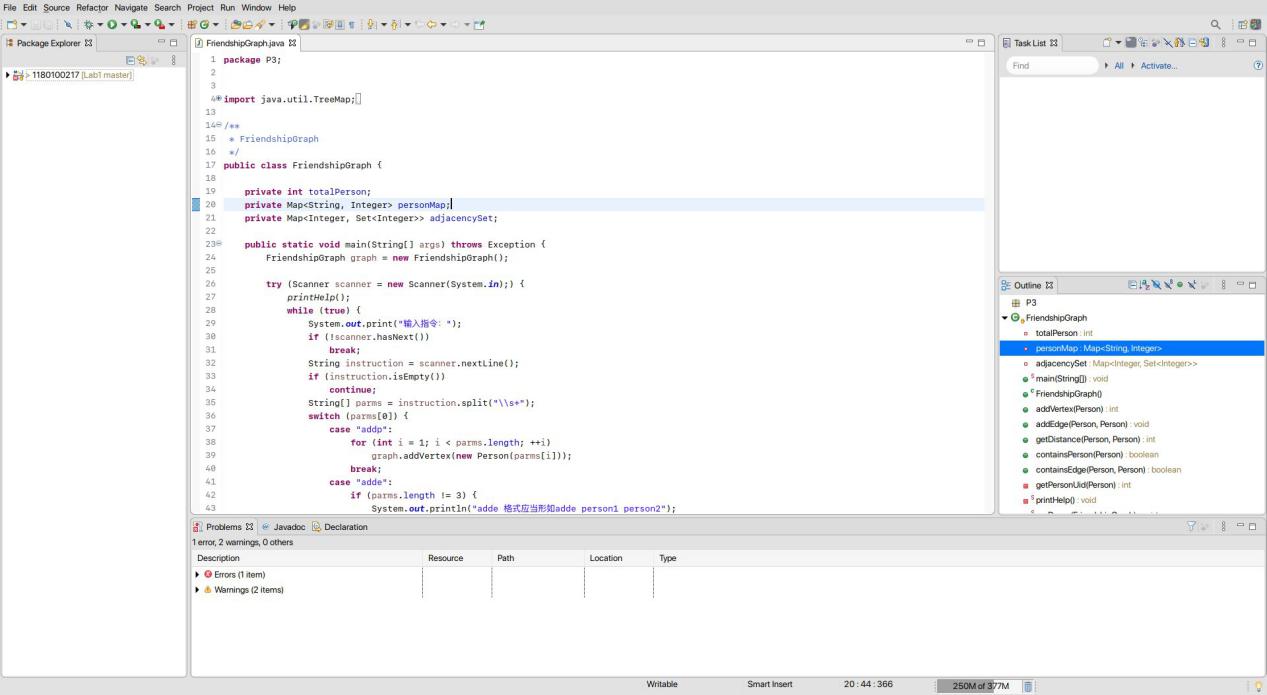


图 2‑2Eclipse运行截图

因为实验测试时使用的是Java8，所以要设置项目的jdk版本。可以在File->Settings中修改Java Compiler版本。

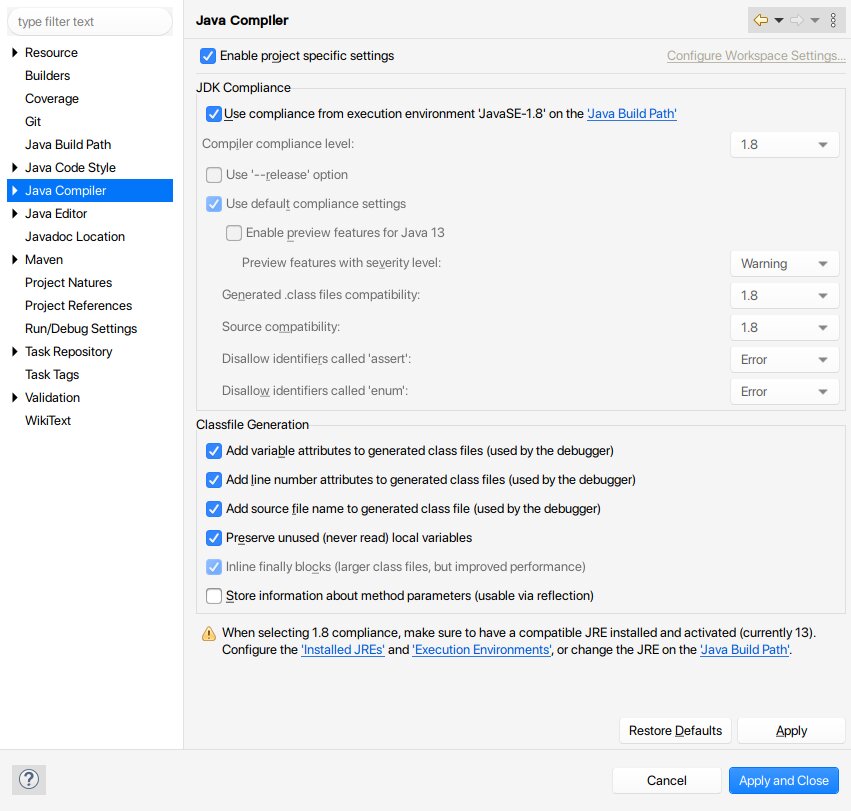


图 2‑3配置JDK版本

## GitHub Lab1地址

https://github.com/ComputerScienceHIT/Lab1-xxxxxxxxxx

# 实验过程

## Magic Squares

本问题具有两个任务：1.判断一个文件包含的是否是一个幻方；2.生成一个奇数阶幻方，并写入文件中。

### isLegalMagicSquare()

作用：判断一个文件包含的是否是一个幻方。

实现：首先将文件按行读入，将每行的字符串按照制表符分割，进而生成一个矩阵。在生成的过程中，需要判断是否有非法字符、“矩阵”的不同行的列数是否相等（即是否是一个合法的矩阵）。之后判断矩阵的行列数是否相等（即是否是一个方阵）。最后判断矩阵的行、列、对角线的和是否相等（即是否是一个幻方）。

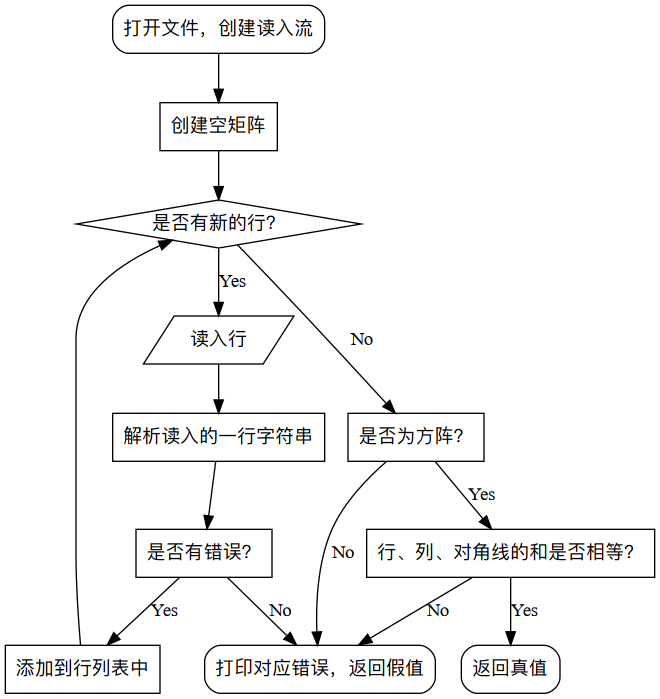


图 3‑1isLegalMagicSquare流程图

### generateMagicSquare()

作用：生成一个奇数阶幻方，并写入文件（src/P1/txt/6.txt）中。

实现：具体实现已经在实验报告中给出，其构造方法Siamese方法，为思想是从第一行中间沿对角线开始填数字，当条对角线填完后填下一行。可以证明这样的方法可以生成奇数阶幻方。

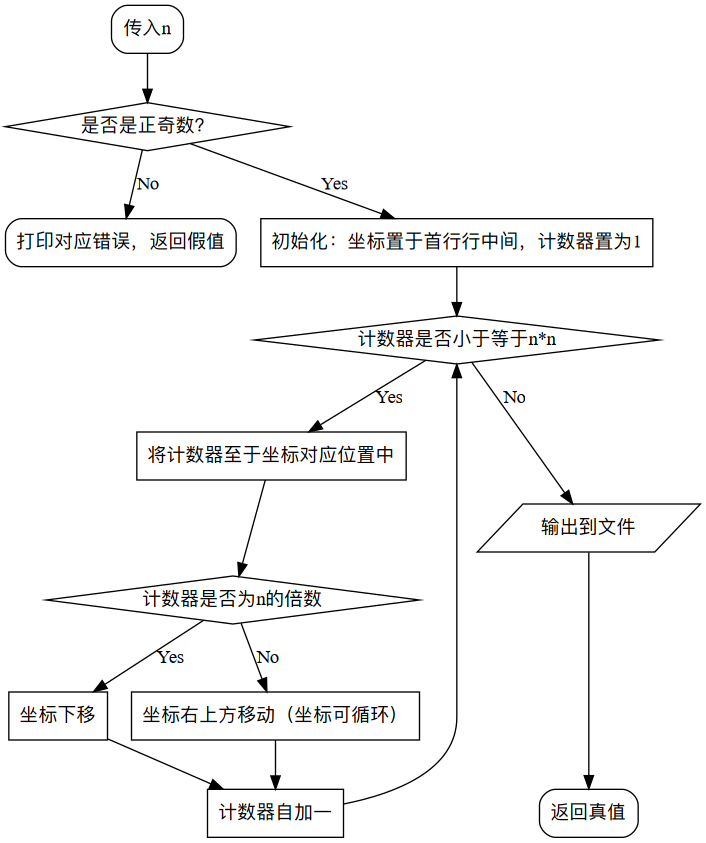


图 3‑2generateMagicSquare流程图

在实验手册的代码中，如果输入的n为偶数，函数运行之后在控制台产生以下输出：

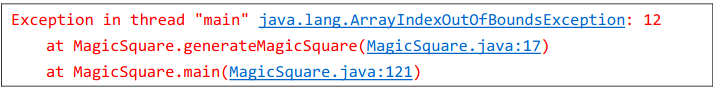


图 3‑3 n为偶数时抛出异常

此异常提示访问到了的索引大于等于数组的长度，数组下标越界。当n为偶数时，程序总会访问到（n，0）进而发生下标越界。

在实验手册的代码中，如果输入的n为负数，函数运行之后在控制台产生以下输出：

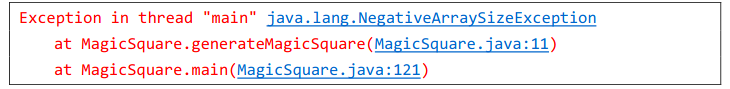


图 3‑4 n为负数时抛出异常

此异常提示尝试创建一个大小为负数的数组。当n为负数时，程序创建一个大小为n的数组，因而抛出异常。

## Turtle Graphics

该任务实现了一个类似“小乌龟画图”logo语言的类turtle，要求通过这个类绘制一些图形，并且实现一些辅助函数和求凸包。

该任务要求实现的函数较多，主要目的是了解Eclipse和git的使用方法，练习基本的Java编程。

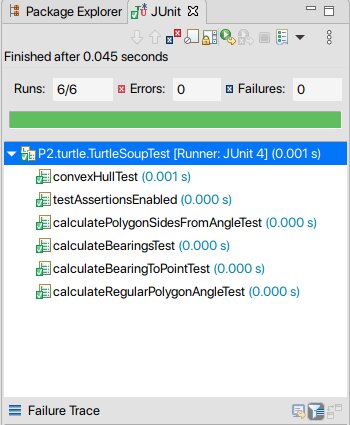


图 3‑5使用Eclipse测试通过

### 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。

在终端中输入如下指令可以从GitHub下载代码

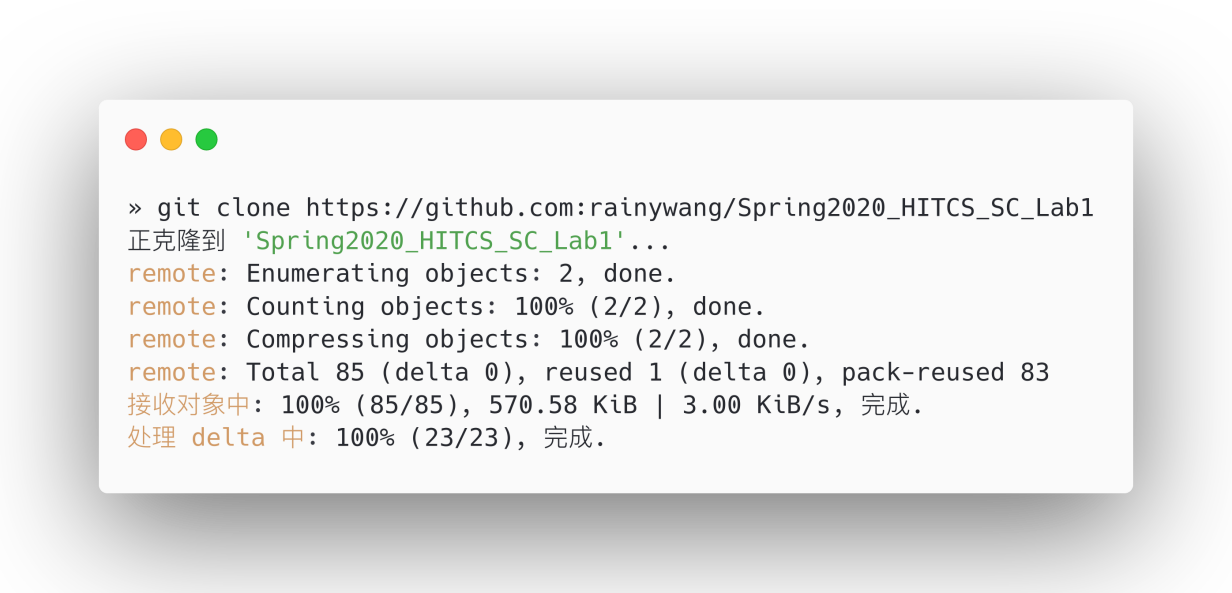


图 3‑6从GitHub下载代码

将下载的代码复制到自己的实验目录，使用git init指令初始化代码仓库。之后通过在终端中输入如下命令将本地仓库与远程远程仓库关联git remote add origin [git@github.com:ComputerScienceHIT/Lab1-xxxxxxxxxx.git。](mailto:git@github.com:ComputerScienceHIT/Lab1-1180100217.git。)

在实现一部分代码功能后，可以使用git add /path/to/the/file来暂存修改，通过git commit来提交，通过git push来上传到远程仓库。更多指令可以通过查询手册（<https://git-scm.com/docs>）获得。

### Problem 3: Turtle graphics and drawSquare

要求实现绘画一个正方形，循环执行四次forward和turn函数即可。注意，每次旋转时旋转的是外角。

### Problem 5: Drawing polygons

此阶段的主要任务是画出正多边形。

实现calculateRegularPolygonAngle函数：由几何知识可以得到内角= 180º - 360º / 多边形边数。

实现drawRegularPolygon函数：绘画多边形的方法同上一任务类似，循环执行多边形边数次forward和turn函数即可，不过每次旋转的角度为360º / 多边形边数。

### Problem 6: Calculating Bearings

此阶段要求实现“小乌龟”计算一条路径的转角。

实现calculateBearingToPoint函数：该函数用于计算两点之间的需要的转角角度。首先计算两点之间的相对座标，之后使用Math.atan2可以计算相对于y轴的角度，最后减去原本的角度就可以得到需要的转角。由于返回的角度必须是角度制而且在[0º,360º)间，所以需要转换。

实现calculateBearings函数：通过已经实现的calculateBearingToPoint函数，逐步计算相邻两点间所需的转角即可。

### Problem 7: Convex Hulls

此阶段要求实现求出一个点集的凸包。我使用的是[Monotone chain, a.k.a. Andrew's algorithm](https://en.wikibooks.org/wiki/Algorithm_Implementation/Geometry/Convex_hull/Monotone_chain)，一种基于坐标水平序的O(nlogn)算法。

此算法分为两个阶段，依次求出上凸包和下凸包，中间要使用栈结构，使用一个辅助函数crossProduct来计算两个向量的叉积，进而判断是否维持凸包性质。

首先将所有点按照xy为一二关键字的方法进行排序。将points[0]和points[1]压入栈中，之后依次遍历剩下的点，依次尝试加入新的点，如果破坏了凸包的结构则弹出当前的栈顶（最后两个向量的叉积非负）。之后倒着求下凸包，注意要跳过上凸包中的点。

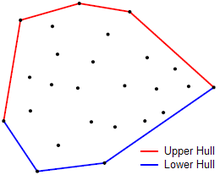


图 3‑7一个使用Monotone chain, a.k.a. Andrew's algorithm求出的凸包

### Problem 8: Personal art

我画了一个通过螺线重复构成的图形，每个阶段的颜色则采用随机函数决定。

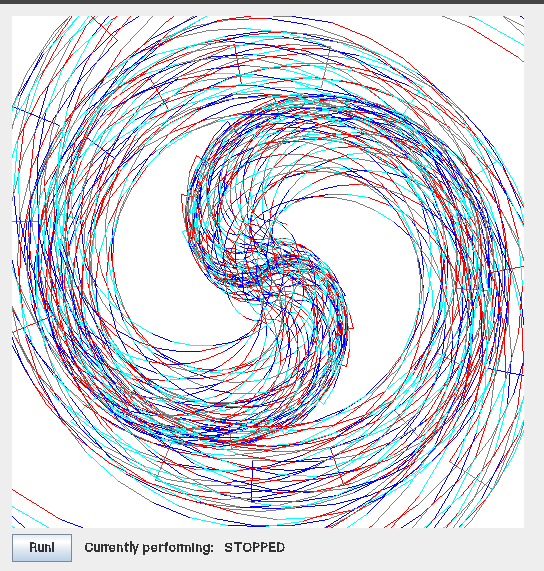


图 3‑8画出的图形

### Submitting

在终端中输入git push origin master即可推送，第一次时要使用git push -u origin master来关联。

## Social Network

要求实现一个简单的网络关系模型，支持节点和有向边的插入和任意两个节点的最短距离的查询。

此任务目的是熟悉OOP的编程思想和了解JUnit进行测试。

图 3‑9使用Maven测试

### 设计/实现FriendshipGraph类

由于节点的数目未知，所以要使用一些大小可变的容器。我在此次实验中使用的是Map容器，personMap容器将每个人与对应的编号（一个整数）对应、adjacencySet将每个人的编号和一个储存边关系的容器对应。存储边关系的容器可以选择List和Set。为了便于测试和提高部分函数的效率，我选择了Set。同时还需要一个计数器来进行标号。使用编号而不是姓名可以提高访问效率，同时减少所需的空间。上述变量均为私有变量。

添加节点时，首先判断要加入的person是否已经在图中（利用Map容器的containsKey方法）：若在，则抛出异常；若不是，则计数器自加，关联姓名和对应的编号，同时初始化对应的Set。同时实现一个containsPerson方法来进行测试。

添加边时，首先获得两个节点对应的编号。如果有两个节点都存在，则在起始点对应的Set中添加终止点。类似的实现containsEdge便于测试。

计算两点之间的最短距离可以使用BFS算法实现即可。不过要注意一些特殊情况：起点和终点相同；起点和终点有一个不在图中；起点和终点并不连通。

### 设计/实现Person类

Person类只是用于存储姓名，值包含一个String变量。所以我实现了一个以String为参数的构造方法和一个getName方法用于获得person的姓名。

### 设计/实现客户端代码main()

客户端支持包含指令addp、adde、dist、list、demo、help和exit。用户可以通过指令来添加节点和边、计算距离、列出图中节点、列出提示和退出。对于错误的输出，程序会提示错误，必要时会退出。



图 3‑10运行示例

### 设计/实现测试用例

我使用的是JUnit4进行测试。

测试使用的样例使用的是实验报告中使用的测试样例，同时通过实现的辅助方法containsPerson和containsEdge进行测试。

在测试时要测试重复添加时是否会抛出异常；是否添加节点和边成功；是否会有意料外的添加等



图 3‑11测试异常

# 实验进度记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 2020-02-25 | 15:30-17:00 | 编写问题1的isLegalMagicSquare和generateMagicSquare函数并进行测试 | 按计划完成 |
| 2020-02-26 | 18:00-20:30 | 编写问题2的TurtleSoup类并进行测试 | 按计划完成 |
| 2020-02-28 | 18:00-19:30 | 编写问题3的Person类和FriendshipGraph类并进行测试 | 按计划完成 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的难点 | 解决途径 |
| 如何从文件中读入并解析出幻方？ | 使用ArrayList套int[]的方式存储数据，使用String.split方法分段，使用Integer.parseInt方法解析出int。 |
| 如何计算出calculateBearingToPoint中需要的转角？ | 首先计算两点之间的相对座标，之后使用Math.atan2可以计算相对于y轴的角度，最后减去原本的角度就可以得到需要的转角。由于返回的角度必须是角度制而且在[0º,360º)间，所以需要转换。 |
| 如何测试重复添加时的异常？ | 使用JUnit的@Test(expected = RuntimeException.class)可以测试是否抛出RuntimeException。 |

# 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

在此次实验，我初步了解了如何Java编程。不过在实验过程中，我在编码花费的时间要少于自己思考设计的时间，编码不是编程的难点，如何设计程序才是重点。同时我在实验过程中查询手册花费的时间过多，一方面说明我对于Java还不够熟悉，另一方面说明我查询手册的方法还可以改进。

## 针对以下方面的感受

1. Java编程语言是否对你的口味？

还好，感觉比较繁琐，没有C++自由，不过很多地方更加方便（比如import和GC）。

1. 关于Eclipse IDE

感觉一般，可能是因为我自己平时使用的是Emacs，更熟悉另一套工作流。个人认为Eclipse的工程管理比较迷，个人更加喜欢Emacs+Maven。

1. 关于Git和GitHub

我自己之前使用过git和GitHub，对于这两个工具比较熟悉。我认为将工业中比较流行和先进的工具加入教学可以提高学生的工程能力，同时这本身也是教育的进步。

1. 关于CMU和MIT的作业

CMU和MIT的作业比较成熟（毕竟他们的计算机教育体系迭代的时间更长），希望工大可以从他们的成果中学习，提高工大自己布置的作业水平。

开源界就不怕“抄袭”，但是新的“代码”应当是更好的更好的，是在巨人的肩膀上进步。

1. 关于本实验的工作量、难度、deadline

此次实验本身的难度并不大。回顾整个实验如果足够认真，我应该可以在一个下午完成（但是总是摸鱼）。在此次实验中我遇到的花费时间较长的问题都不是实验本身，而是使用Maven进行测试、打包等。如果测试时使用的是Maven等测试工具，那么实验的代码结构应当与默认的设置相同，减少学生在这些问题上花费的时间。

1. 关于初接触“软件构造”课程

初步了解了软件构造的过程，不过其实没有太大的感受，可能是因为是初接触。希望以后可以讲解一些具体的案例。