

**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 | 安宏展 |
| 学号 | 1160301023 |
| 班号 | 1603010 |
| 电子邮件 | [2223790237@qq.com](mailto:2223790237@qq.com) |
| 手机号码 | 18846082165 |

**目录**

[1 实验目标概述 1](#_Toc506282558)

[2 实验环境配置 1](#_Toc506282559)

[3 实验过程 1](#_Toc506282560)

[3.1 Magic Squares 1](#_Toc506282561)

[3.1.1 isLegalMagicSquare() 1](#_Toc506282562)

[3.1.2 generateMagicSquare() 1](#_Toc506282563)

[3.2 Turtle Graphics 1](#_Toc506282564)

[3.2.1 Problem 1: Clone and import 2](#_Toc506282565)

[3.2.2 Problem 3: Turtle graphics and drawSquare 2](#_Toc506282566)

[3.2.3 Problem 5: Drawing polygons 2](#_Toc506282567)

[3.2.4 Problem 6: Calculating headings 2](#_Toc506282568)

[3.2.5 Problem 7: Personal art 2](#_Toc506282569)

[3.2.6 Submitting 2](#_Toc506282570)

[3.3 Social Network 2](#_Toc506282571)

[3.3.1 设计/实现FriendshipGraph类 2](#_Toc506282572)

[3.3.2 设计/实现Person类 2](#_Toc506282573)

[3.3.3 设计/实现客户端代码main() 2](#_Toc506282574)

[3.3.4 设计/实现测试用例 2](#_Toc506282575)

[3.4 Tweet Tweet（选作，额外记分） 3](#_Toc506282576)

[4 实验进度记录 3](#_Toc506282577)

[5 实验过程中遇到的困难与解决途径 3](#_Toc506282578)

[6 实验过程中收获的经验、教训、感想 3](#_Toc506282579)

# 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。

基本的 Java OO 编程 

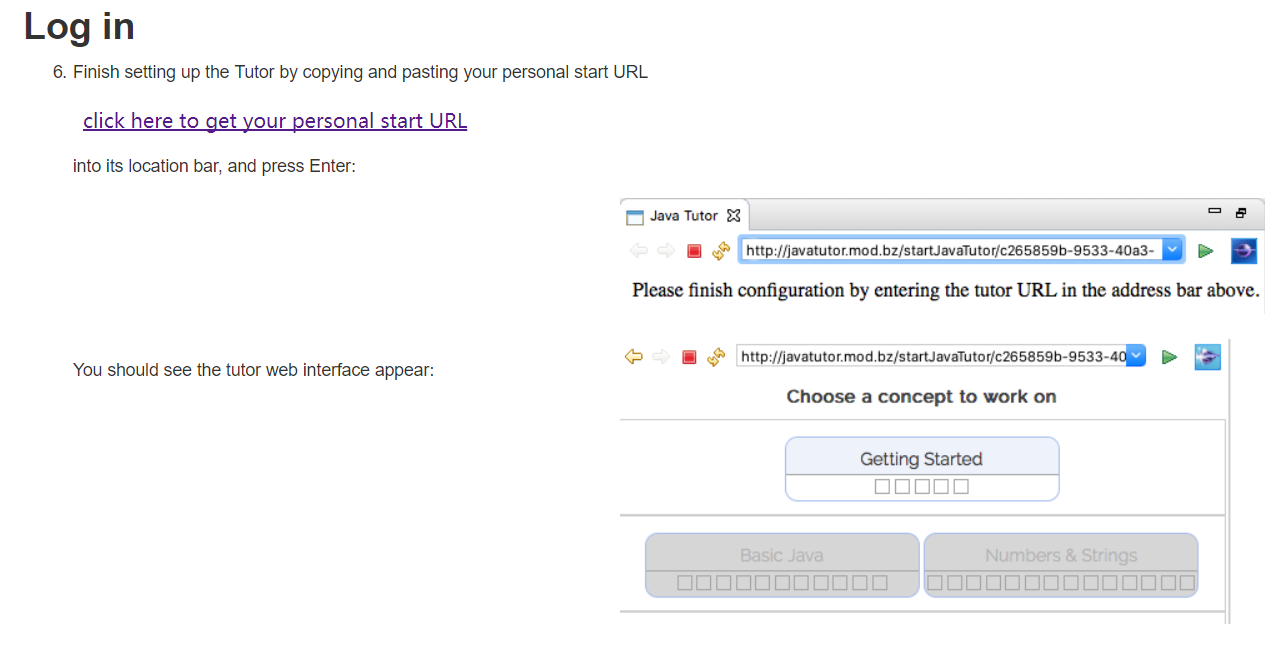
基于 Eclipse IDE 进行 Java 编程 

基于 JUnit 的测试 

基于 Git 的代码配置管理

# 实验环境配置

1. java环境的配置问题：jdk和jre的路径配置混乱，导致不能运行java虚拟机无法运行，需要从头开始检查配置环境，
2. Tutor安装时无法获得MIT的id



1. Git的使用问题

Git的操作比较繁琐，很多命令都无法熟记，全英文的界面对于使用也产生了很大的麻烦，在学习Git操作的时候，很多时候会发生不知道进行到哪一步，或者某一步骤出错然后无法回退的问题，只能重新开始，非常繁琐。

在这里给出你的GitHub Lab1仓库的URL地址（Lab1-学号）。

<https://github.com/ComputerScienceHIT/Lab1-1160301023>

# 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（但千万不要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

幻方(Magic Square)是一个非常有趣的东西，维基百科上说幻方是数字均为整数的正方形矩阵，并且横竖列数字和与两条对角线数字和都需要相等，这就提示了很多关于判断幻方的方法，也就是后面函数的思路。

### isLegalMagicSquare()

1. 文件读取：本题的数据需要使用文件读入，根据要求可知该函数里面存在一个名为filename的String类型的数据作为参数，这就说明文件的路径作为函数的参数，通过输入不同的文件路径来读取不同的数据。文件的读入采用IO输入流的方法进行读取，首先new出一个File类型对象用于作为输入流的源 

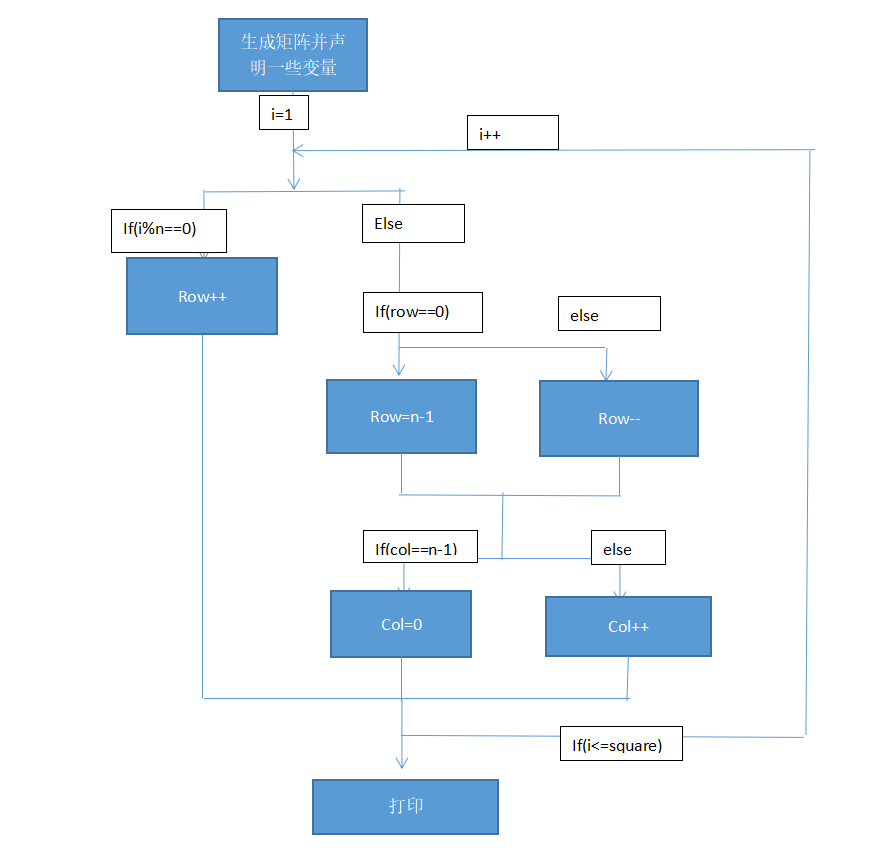
然后将其放入输入流并且进行读取 

1. 判断行与列的大小：在文件的头部设置一个mark，然后开始按行读取文件，知道读取的变量为null来计算行数line，列数的计算使用split方法按照“\t”进行分割，然后使用一个字符串数组来存储分割出的数据，判断数组的长度与line的大小是否相等即可确定行列数的关系 
2. 存储数据：使用reset方法返回文件的头部（之前在文件的头部设置过mark),申请一个长宽均为line的二维数组，按行读取，split方法分割分别进行存储，采用正则表达式判断字符串转换后的数字是否为整数 
3. 计算行与列以及对角线的数字之和，判断是否相等

### generateMagicSquare()

按步骤给出你的设计和实现思路/过程/结果。

程序流程图



生成方法：首先将1放置在第一行的中间位置，然后沿着左下到右上的方向放置接下来的数字，如果沿着该方向延伸超过了该矩阵的范围， 则假想存在一个紧邻着该矩阵的相同的矩阵，然后在原矩阵对应的位置放置数字，如果延伸过程中下一个位置已经存在了数字，则向下移动一格，然后继续按照原方法进行延伸，知道整个矩阵都被填满为止。

异常：n为偶数时的异常为数组越界异常--由上面所描述的生成方法可知（n为矩阵大小，row代表行，cow代表列），当i%n==0时，row++；当row==n-1时，row=0,否则row++;当cow==n-1时，cow=0,否则cow++;当n为奇数时，这个程序在依次的进行下去，当n位偶数时，会存在某一个时间。当row==n-1时i%n==0,此时row==n，会发生数组越界的情况，所以会出现异常



n为负数时，很明显数组里面的参数不能为负



## Turtle Graphics

在这里简要概述你对该任务的理解。

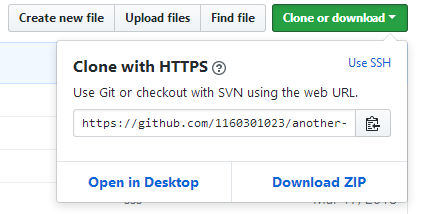
利用正多边形内角外角的关系进行简单的swing编程，简单了解java的graphics类如何进行绘图

### 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。

git clone ssh的地址

或者直接在github的仓库里面直接直接点击clone or download



### Problem 3: Turtle graphics and drawSquare

首先观察turtle接口，可以发现存在forward()、turn()、color()和draw()四种方法

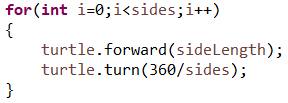


根据drawSquare()方法输入的参数可知我们现在已有一个turtle对象已经需要画的square的长度，所以两种只要forward、turn、forward、turn、forward、turn、forward即可得到所需的图形

### Problem 5: Drawing polygons



绘制正多边形，由正多边形的外角公式可知正多边形的外角大小为360/sides，所以每次forward(sidelength)之后turn(360/sides)即可



### Problem 6: Calculating headings

首先必须明确角度必须是顺时针旋转得到的角度并且处于0--360之间

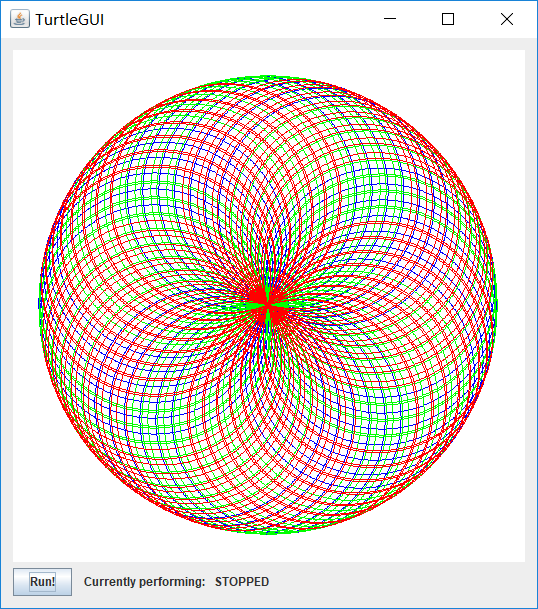
然后就是明确角度的实现方法，分为三部分实现

1. targetX > currentX：由于atan函数返回的数值位于-/2--/2之间，所以将其转换位角度制之后需要+180然后对180除余即可。
2. targetX < currentX：将1所得到的公式+360然后膜360即可
3. targetY == currentY：此时不能使用atan函数，只需要但是此时的角度已知，只需要判断目标点与原点纵坐标的大小即可



### Problem 7: Personal art

由于forward()方法里面的参数定义为整数，所以每次向当前的方向移动距离‘1’，通过微小的移动并且微小的改变方向来画圆，但是由于每次圆是通过直线画出来的，所以每次花园都会出现偏差，改变初始的旋转方向并且改变颜色就可以得到如下的一张图片：



### Submitting

如何通过Git提交当前版本到GitHub上你的Lab1仓库。

找到我们需要上传的项目文件->右键git bush->首先进行初始化



-> 将本地的库与远程的仓库进行链接

**git remote add origin https://gitee.com/\*\*\*/项目名.git**

-> 然后输入git init ->然后将本地文件上传到本地的库里面 git add “文件名”

-> 然后对本次上传进行命名 git commit -m “本次上传名称的描述（仅自己可知）”

-> 将本地库里面的文件上传至远程github的库里面

git push -u origin master 即可

## Social Network

在这里简要概述你对该任务的理解。

加深对数据结构中无向图、bfs（广度优先搜索）的理解，以及对于无向图的构建方法（邻接矩阵、邻接表、多重邻接表等）的复习

### 设计/实现FriendshipGraph类

给出你的设计和实现思路/过程/结果。

根据实验要求可知该类存在以下的方法：addEdge、getDistance、addVertex



addEdge方法：每个Person类里面都存在一个链表用于存储Person，只要将Person

add进去即可



getDistance方法：通过bfs算法遍历所有的Person对象得到两个人之间的最近距离



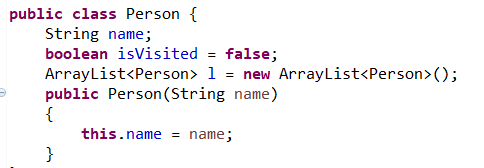
addVertax方法：每一个Graph都存在一个链表用于存储这个图里面的所有Person

### 设计/实现Person类

给出你的设计和实现思路/过程/结果。

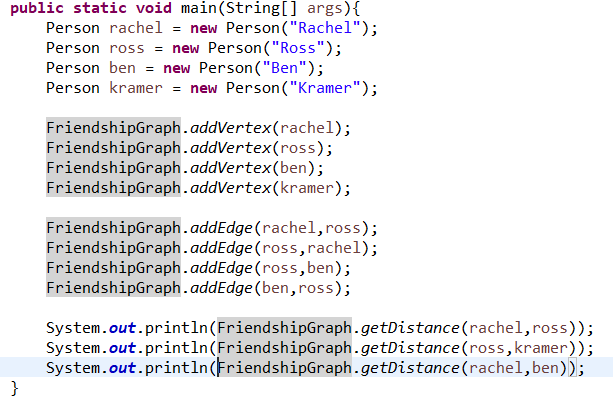
每个Person里面都存在name、用于存放邻接Person的链表、以及isVisited

isVisited用于表示是否被访问过，在bfs里面访问是需要用到



### 设计/实现客户端代码main()

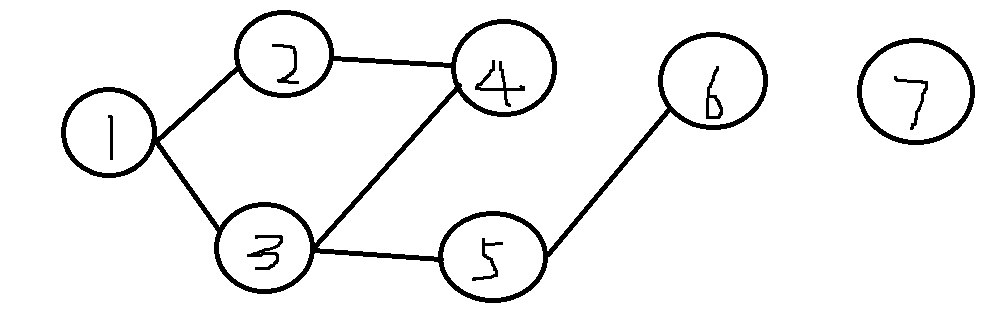
给出你的设计和实现思路/过程/结果。



以上为按照实验手册给出的样例编写main函数

### 设计/实现测试用例

给出你的设计和实现思路/过程/结果。



getDistanceTest：如上图所示

addVertaxTest、addEdgeTest：测试graph的链表与每个Person对象的链表是否相同即可

## Tweet Tweet（选作，额外记分）

## 3.4.1、Extracting data from tweets



getTimespan方法返回一个Timespan对象，他的start end为tweets里面的最小的时间和最大的时间，所以只要选择差最大的那一组时间即可



getMentionedUsers方法返回的是所有tweets所follow的用户名，所以从每一个用户的text里面寻求follow的用户即可，寻求的方法为：将每一个text的内容按照‘空格’分割得到字符串数组，将每一个字符串分为一个字符数组，每个字符数组从第一个字符开始判断，如果为名字的合法字符（‘a'--’z’ ‘A’--’Z’ ‘-’ ‘\_’ ‘0’--’9’）则将该字符串丢弃，如果不是合法字符则继续向下进行，当遇到’@‘时，则将’@‘后面挨着的所有的名字的合法字符保存下来，然后将其add进入set集合即可，即为该Tweet对象所follow的用户名，一直循环直到最后一个字符数组为止。

3.4.2、Filtering lists of tweets



writtenBy()方法返回值为所有的与username相同的字符串list，注意tweet的名字书写时不区分大小写



inTimespan()方法返回存在与timespan的start与end之间的getTimestamp(),将所有满足该条件的getTimestamp()加入到list数组里面返回即可。注意将timespan的start与end转换为按照秒计时的数值，这样才能进行大小的比较



Containing()方法返回所有text分别包含words所有内容的username，注意是只须包含words里面的所有的内容，不需要按照顺序的包含。进行循环的方法为首先遍历words然后遍历tweets，具体的实现方法如下（采用常用的flag的方法）：



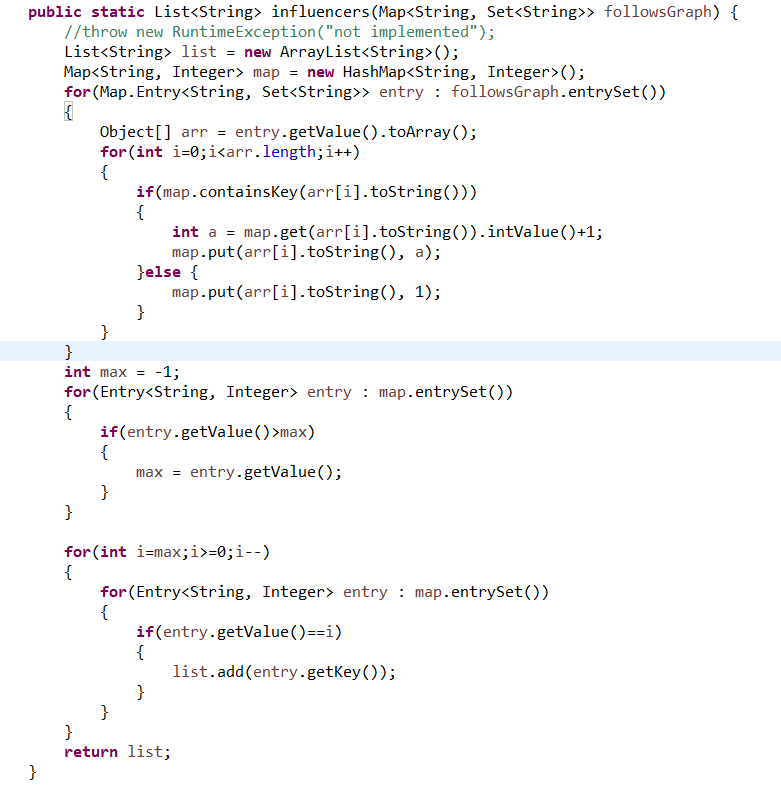
3.4.3、Inferring a social network



guessFollowsGraph()方法返回username与其分别follow的所有的用户名所构成的Map集合类,username即为Map的Key,而其所对应的Set<String>可以直接调用前面的函数getMentionedUsers()即可



Influencers()方法：String-->A Set-->B C D,则相互关系里面B C D都的followers都存在A，所以将函数输入的参数map转换为另一个Map<String,Set<String>>即可，新的Map的Key为原map里面Set集合类里面的内容，Key对应的value为原Map里面的Key，添加的方法如上面所示，最后将Set的size从大到小进行排列并且添加到list里面，然后返回list



3.4.4、Get smarter

这个任务需要的是重新定义influences()方法，本人选择的是common hashtags，每个人所发的文章里面都存在一些hashtag，例如#mit，拥有相同的hashtag的一组人互为influences

实现的方法为：首先找到所有的hashtag，然后将拥有相同的hashtag的人分类到一起，最后将同一类的所有人都互相添加进入自己的Set集合即可

# 实验进度记录

请尽可能详细的记录你的进度情况。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 计划任务 | 实际完成情况 |
| 2018-02-25 | 14:00-15:30 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按计划完成 |
| 2018-02-28 | 9：00-14：00 | 编写3.2的所有 问题 | 延期到当天晚上完成 |
| 2018-3-1 | 9：00-17：00 | 完成3.3的所有方法以及测试 | 按计划完成 |
| 2018-3-5 | 8：00-17：00 | 编写problem4的Extract.java  Filter.java | 延迟到第二天完成 |
| 2018-3-10 | 全天 | 完成了剩余的所有的代码，包括剩下的所有的方法以及所有的 | 按计划完成 |
| 2018-3-15 | 全天 | 使用在线构建的方法对整个项目进行build并且上传，实验完成 | 按计划完成 |

# 实验过程中遇到的困难与解决途径

1. 英语的翻译问题：所有问题的描述都是英文的，完全的理解问题需要一些努力，只有不断的读和翻译才能得到真正的意思
2. JUnit的test代码编写：由于是第一次接触到测试样例的编写，与正常的代码编写格式存在一些不同，并且只是关于单个的函数或方法进行测试
3. 使用各种集合类时存在很多的遍历、指针异常等等的问题。通过查看一些博客以及查看java的帮助文档得到了很多的信息

# 实验过程中收获的经验、教训、感想

本节除了总结你在实验过程中收获的经验和教训，也可就以下方面谈谈你的感受（非必须）：

1. Java编程语言是否对你的口味？
2. 关于Eclipse IDE
3. 关于Git
4. 关于CMU和MIT的作业
5. 关于本实验的工作量、难度、deadline
6. 关于初接触“软件构造”课程

首先还是自己学艺不精，无论是英语还是java的代码编写都存在很多需要提升的地方，还有对于知识的获取途径方面，以前我只知道去网上百度、google，但是想要获得java的大部分的信息最好的途径就是查看java的帮助文档，刚开始查看的时候肯定会存在很多的问题，但是经过一段时间的查看就会发现这个才是最便捷的方法。

java这种编程语言对于我个人来说还是比较喜欢的，因为我们每个人在这个世界上面其实就是一个对象，新的对象从申请到最后jvm处理之间就和人的一生相似，甚至还包含生病以及病好了这种情况，比较人性化。Eclipse这个编译器用起来还是可以的，毕竟当初学习java最先使用的就是Eclipse.Git是我们每个人必须学会使用的一项技能，越早接触越好，这一项非常的好。CMU和MIT的第一次作业还是比较基础的，考察的大概都是一些基础语法以及数据结构的知识，本次实验的工作量个人觉得适中，三周时间完成3+1的实验内容还是可以的，但是不知道没有学过java的同学是否能够接收。