

**2023年春季学期**

**计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 房昕玉 |
| 学号 | 2021113290 |
| 班号 | 2137102 |
| 电子邮件 | 2354598747@qq.com |
| 手机号码 | 18629831988 |

**目录**

[1 实验目标概述 1](#_Toc28717)

[2 实验环境配置 1](#_Toc5112)

[2.1 实验环境配置（所需开发、测试、运行环境） 1](#_Toc22992)

[2.2 GitHub Lab1仓库的URL地址 1](#_Toc3600)

[3 实验过程 1](#_Toc21414)

[3.1 Magic Squares 2](#_Toc16596)

[3.1.1 isLegalMagicSquare() 2](#_Toc20643)

[3.1.2 generateMagicSquare() 2](#_Toc10142)

[3.2 Turtle Graphics 4](#_Toc10793)

[3.2.1 Problem 1: Clone and import 4](#_Toc31751)

[3.2.2 Problem 3: Turtle graphics and drawSquare 5](#_Toc27170)

[3.2.3 Problem 5: Drawing polygons 5](#_Toc25632)

[3.2.4 Problem 6: Calculating Bearings 5](#_Toc6885)

[3.2.5 Problem 7: Convex Hulls 5](#_Toc29516)

[3.2.6 Problem 8: Personal art 5](#_Toc29475)

[3.2.7 Submitting 5](#_Toc6886)

[3.3 Social Network 6](#_Toc9849)

[3.3.1 设计/实现FriendshipGraph类 6](#_Toc20762)

[3.3.2 设计/实现Person类 6](#_Toc10075)

[3.3.3 设计/实现客户端代码main() 7](#_Toc10144)

[3.3.4 设计/实现测试用例 7](#_Toc19815)

[4 实验进度记录 8](#_Toc6459)

[5 实验过程中遇到的困难与解决途径 8](#_Toc11123)

[6 实验过程中收获的经验、教训、感想 9](#_Toc19489)

[6.1 实验过程中收获的经验和教训（必答） 9](#_Toc31430)

[6.2 针对以下方面的感受（必答） 9](#_Toc14522)

# 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。

l 基本的 Java OO 编程

l 基于 Eclipse IDE 进行 Java 编程

l 基于 JUnit 的测试

l 基于 Git 的代码配置管理

# 实验环境配置

//简要陈述你配置本次实验所需开发、测试、运行环境的过程，必要时可以给出屏幕截图。

## 实验环境配置（所需开发、测试、运行环境）

下载git，idea，配置junit插件；

Git和idea是之前已经下过的，只需要在idea加入junit插件即可。

## GitHub Lab1仓库的URL地址

git@github.com:ComputerScienceHIT/HIT-Lab1-2021113290.git

# 实验过程

//请仔细对照实验手册，针对三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

//在这里简要概述你对该任务的理解。

n阶的魔法平方是n×n个数字的排列，通常是不同的整数，这样所有行、所有列和两个对角线中的n个数字和为一个常数。

实验要求:

(1)编写判断文件中的数据是否为魔方的函数

(2)对给出的产生魔方的函数做拓展:使用try catch对输入偶数和负数的情况做处理，将产生的魔方写入6.txt文件

### isLegalMagicSquare()

//按步骤给出你的设计和实现思路/过程/结果。

#### 对函数的理解

函数isLegalMagicSquare()判断一个txt文件中保存的矩阵是否为魔方。输入参数为txt文件的文件路径，函数返回一个boolean值，若符合要求返回true，否则返回false。

#### 实现该函数的具体思路

FileInputStream 的 read() 方法有一个重载版本可以从输入流中读取指定长度字节的数据到字节数组中。我们可以使用这种方法一次将整个文件读入一个字节数组。然后可以使用 String 的构造函数将相应的字节解码为具有指定字符集的字符。指定字符集为UTF-8,是UNICODE 的一种变长字符编码。

使用split(“\n”)分割每一行，使用split(“\t”)分割每一行的每一个字符，并将格式转化为int。

这里注意NumberFormatException的错误处理，以及处理下列情况：文件中的数据不符合 Magic Square 的定义（行列数不相等、并非矩阵等）、矩阵中的某些数字并非正整数、数字之间并非使用\t 分割、等。循环计算每一行的和，每一列的和，两条对角线的和。

判断所有行的和是否为相同的常量，列和对角线是否为相同的常量

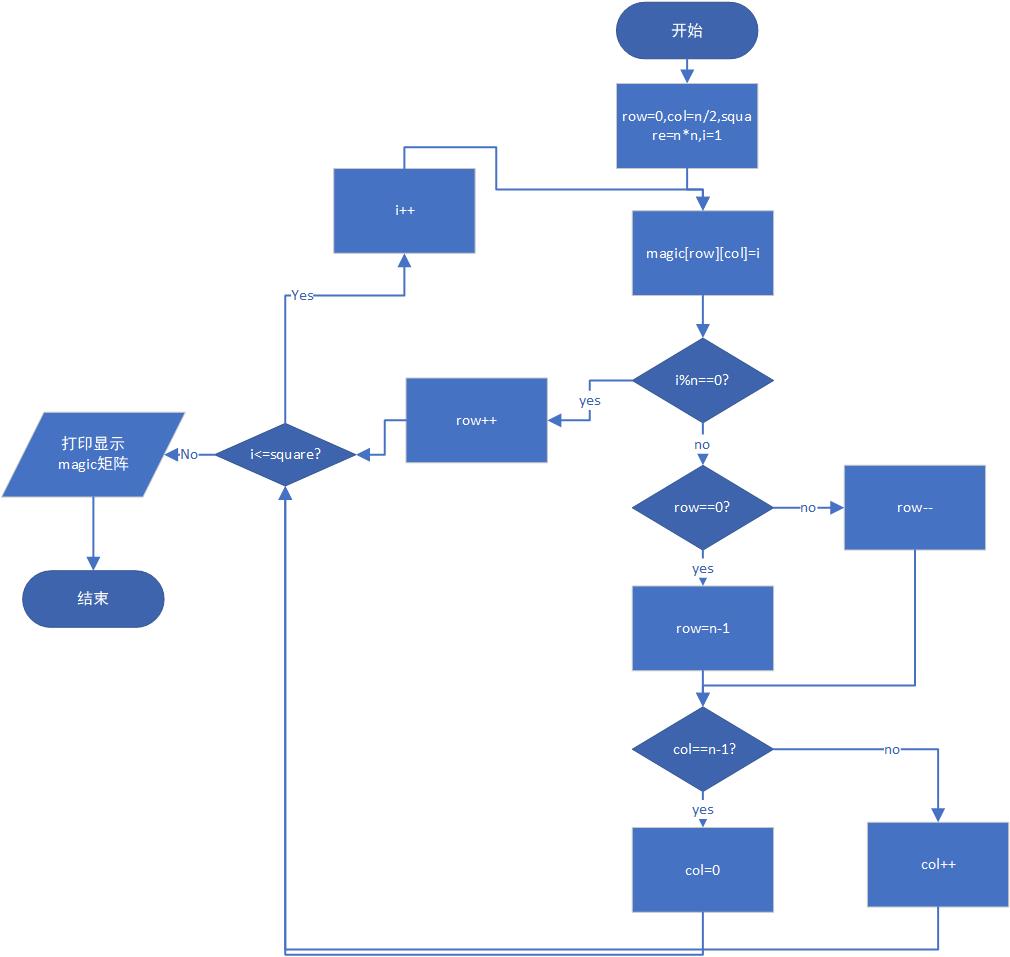
### generateMagicSquare()

//按步骤给出你的设计和实现思路/过程/结果。

#### 对函数的理解

函数generateMagicSquare ( )生成一个符合魔方定义的矩阵。输入参数为奇正数，函数返回一个boolean值，若符合要求返回true，否则返回false。

#### 程序流程图



#### 实现该函数的具体思路

输入的参数为魔方的行或列，是奇数，首先从第一行中间开始赋值i，i的初值为1，每次迭代后加1，判断i是否为行或列的倍数：如果是，则行数加1，因为此时该行该列必被赋值；如果不是，接着分别判断边界情况：如果在第一行，则跳转到最后一行；否则行数减1。如果在最后一列，则跳转到最后一列；否则列数减1，如此循环，产生的矩阵必为幻方。

#### 对函数进行拓展

1. 增加使用try catch的处理输入偶数和负数的情况

try{...

}catch (ArrayIndexOutOfBoundsException e) {

System.out.println("输入为偶数");

return false;

}catch (NegativeArraySizeException e){

System.out.println("输入为负数");

return false;

}

1. 使用FileWriter和BufferedWriter将产生的幻方写入6.txt

try {

File file = new File("src/P1/6.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fileWriter = new FileWriter(file.getAbsoluteFile());

BufferedWriter bw = new BufferedWriter(fileWriter);

for (i = 0; i < n; i++) {

for (j = 0; j < n; j++) {

bw.write(magic[i][j] + "\t");

}

bw.write("\n");

}

bw.close();

System.out.println("finish");

} catch (IOException e) {

e.printStackTrace();

}

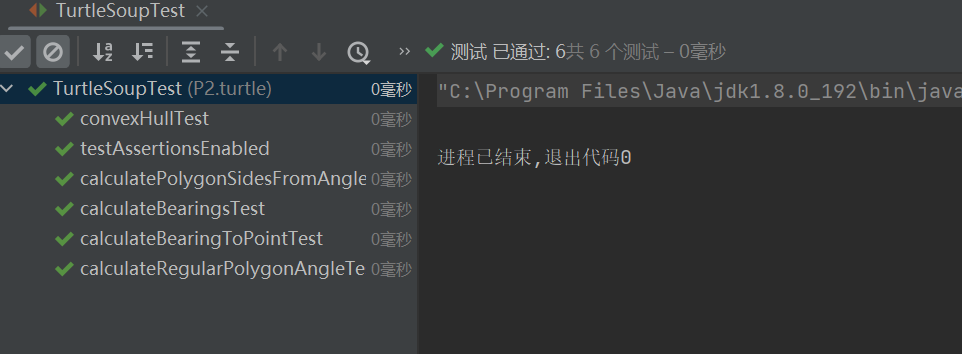
return true;

## Turtle Graphics

在这里简要概述你对该任务的理解。

该实验给出了一个画图程序，给出相关类和方法，将github上的代码clone下来，补全turtlesoup和通过turtlesoupTest。

测试通过：



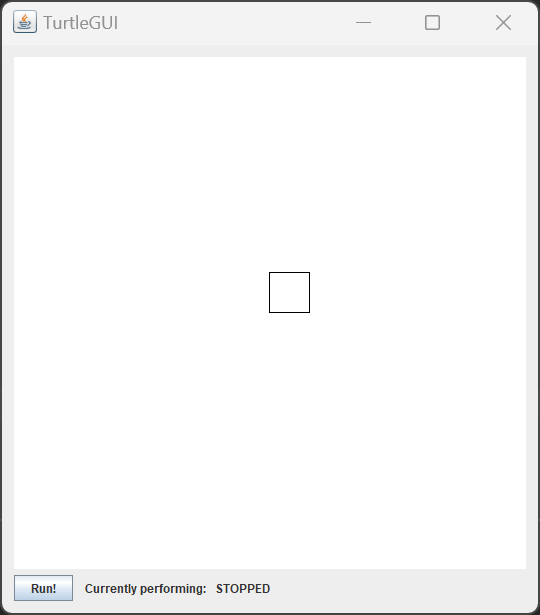
### Problem 1: Clone and import

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。

通过idea上的git，clone远程仓库的代码。

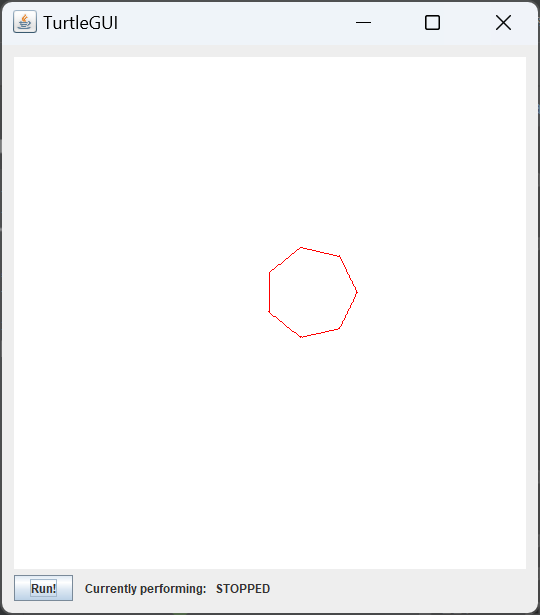
### Problem 3: Turtle graphics and drawSquare

在TurtleSoup.java中实现drawSquare()函数，画一个正方形



### Problem 5: Drawing polygons

在TurtleSoup.java中实现calculateRegularPolygonAngle()函数，返回正多边形的内角和，并通过测试类，逆推公式实现calculatePolygonSidesFromAngle()函数，返回正多边形的边数，并通过测试类，实现drawRegularPolygon()函数，画出一个正多边形



### Problem 6: Calculating Bearings

在TurtleSoup.java中实现calculateBearingToPoiint()函数，返回应旋转的度数，并通过测试类。

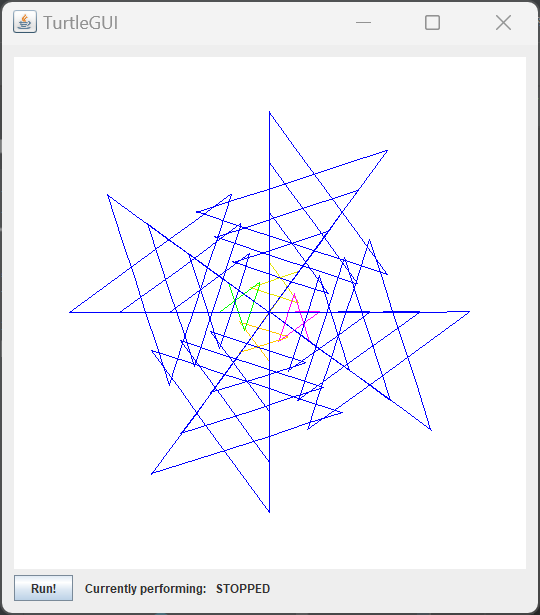
使用Java.util.List,使用calculateBearingToPoiint()函数计算每一次的旋转角度并保存到List中，初始角度为0，计算到达第一个终点需要旋转的角度，记录到List中，将该终点作为下一次的起点...如此循环，直到一个点。

### Problem 7: Convex Hulls

凸包，根据mit上面的提示选择了gift-wrapping算法，首先考虑点比较少的情况，当点少于4个的时候，直接返回这个点集；当点数比较多时，用HashSet建立一个空集合set，首先以最左下角的点作为初始点，添加到set中，重写calculateBearingToPoint，使其坐标点的横纵值可以为double类型（也可以重载），记录初始角度currentBearing为0，循环判断寻找current与p间角度最小的点作为target，如果current与两个点角度相等，则取距离current更大的点作为target，将target加入set，更新current和currentBearing

### Problem 8: Personal art

根据五角星的角度画出一个五角星，在此基础上，循环，增加边长，旋转，最终得到了一片类似“雪花”的图形



### Submitting

如何通过Git提交当前版本到GitHub上你的Lab1仓库。

1.复制仓库地址

2.在本地新建文件夹，右键git bash here

3.git clone 仓库地址

4.把需要上传的文件夹放入到远程仓库文件夹内

5. cd 远程仓库名

git init

git add .

git commit -m “提交信息”

git push

## Social Network

在这里简要概述你对该任务的理解。

设计友谊图，将人作为顶点添加到图中，人与人之间是朋友的加边，并且实现求两个人之间的距离，即图中两顶点的路径长。

### 设计/实现FriendshipGraph类

给出你的设计和实现思路/过程/结果。

使用HashMap来定义图，key值类型为Person，含义是图中包含的顶点；Value值类型为ArrayList<Person>，含义是对应key值的朋友，有连线的顶点。

需要实现三个方法：

addVertex()：参数列表是Person的实例person，首先判断传过来的实例名称是否与哈希图中的已包含的名称重复，如果重复，则报错；否则将该实例及其对应的value值加入到哈希图中。

addEdge():参数列表是Person的两个实例person1，person2，判断图中是否有key值为person1的点，如果有，将person2添加到person1的value值中（person2就是person1的单向的朋友啦）

getDistance():参数列表是Person的两个实例person1，person2,返回两个人之间的距离，实际上是广度优先算法，考虑特殊情况：person1==person2，则返回0，使用一个队列和一个visited数组，数组表示访问过的人，初始时将person1添加到队列中并标记已访问，dis初值为0，当队列不空时，弹出元素，dis++，遍历弹出元素的所有朋友（连线的）。如果等于person2,则返回dis。把访问过的朋友加入到visited数组和队列中，接着查找朋友的朋友，dis++..........循环直到所有点找完。

### 设计/实现Person类

给出你的设计和实现思路/过程/结果。

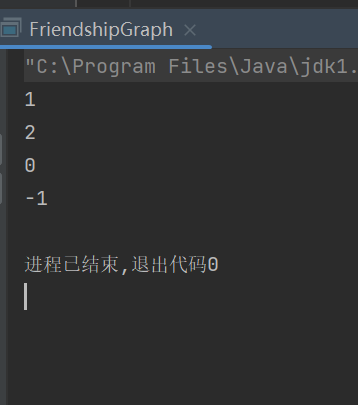
Person类的变量为此人的名字，生成相应构造函数，其中的方法即得到一个人的名字getName()

### 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。

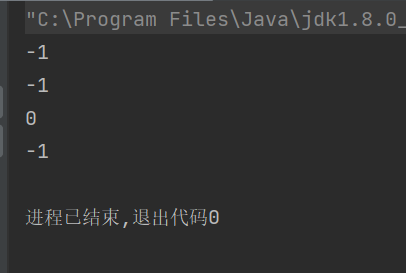
复制实验一要求中的代码运行：

运行结果：



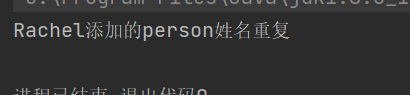
注释掉提供的代码的第10行：因为rachel到ross的边不存在了，所以rachel无法到达其他点，无法计算距离，只有rachel到rachel自己的距离才可以计算。

运行结果：



将第 3 行引号中的“Ross”替换为“Rachel”：

运行结果：

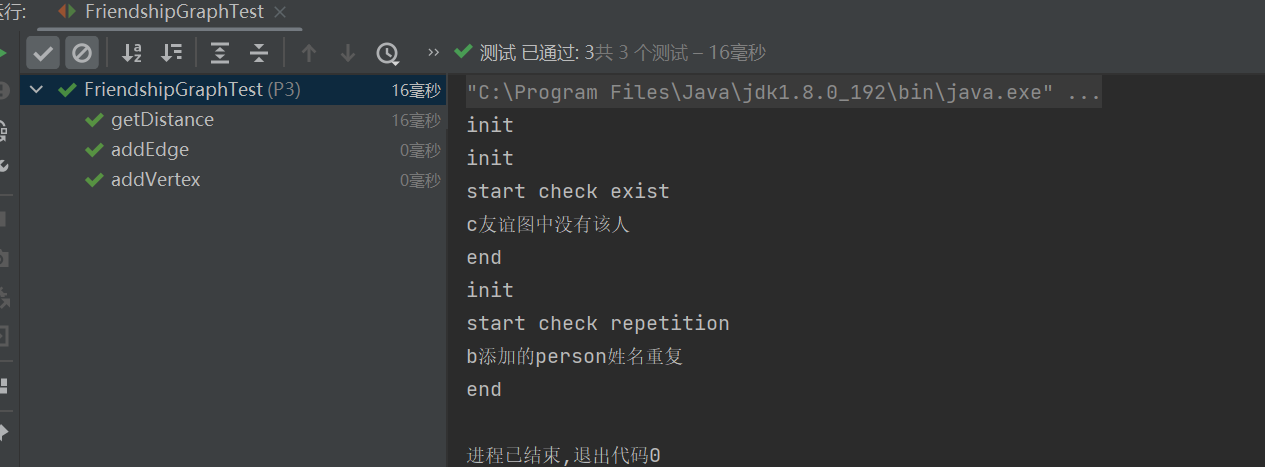
退出程序

### 设计/实现测试用例

给出你的设计和实现思路/过程/结果。

为三个方法分别设计测试用例，addVertex()检查是否能添加点，通过元素个数和是否包含相应key值来检查；同时，检查是否有重名的报错；addEdge()检查是否添加了边，通过key值的value是否包含连线端点来检查，同时，检查是否给不在图中的人加了边的报错；getDistance()检查返回的距离是否正确，设计测试用例检查距离大于0，等于0，小于0的情况。

三个测试通过：



# 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 |
| 2023-03-04 | 8:30-12:30 | 编写问题1的isLegalMagicSquare函数并进行测试 | 遇到困难，未完成 |
| 2023-03-04 | 14:30-17:30 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按计划完成 |
| 2023-03-04 | 19:40-20:10 | 编写问题2的Problem1,3 | 按计划完成 |
| 2023-03-05 | 15:00-16:30 | 编写问题2的Problem5,6 | 按计划完成 |
| 2023-03-06 | 12:00-14:00 | 编写问题2的Problem7,8 | 遇到困难，凸包算法不会，未完成 |
| 2023-03-06 | 15:00-17:00 | 编写问题2的Problem7,8 | 按计划完成 |
| 2023-03-07 | 16:00-17:30  19:00-21:00 | 编写问题3 | 按计划完成 |
| 2023-03-07 | 18:40-19:50 | 完善doc | 按计划完成 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的困难 | 解决途径 |
| 配置idea的远程路径时，无法找到仓库 | 经过同学的帮助，发现是我在IDEA的git集成并没有配置好http形式的身份认证，尝试改为使用ssh的方式完成clone，因为ssh的clone必须要认证，解决了问题 |
| 一些地方没有提前想到try catch的用法 | 在编译运行后报错后，通过搜索引擎查找相关报错原因，使用try catch，提示相关错误并输出false退出。例如：错误：NumberFormatException,查询到是因为三种原因：输入为字母时，即内容不是数字时，例如abc；输入为空时；输入超过int上限时 ，然后使用try catch. |
| 导入代码后，发现许多地方标红 | 检查后发现，是要根据目录结构在import的包前加上P2. |
| 求解凸包问题，不会定义及相关算法 | 百度学习了凸包的定义以及相关算法，经过思考后给出实现 |
| 编写友谊图的测试用例时不知道从何入手 | 认真思考原代码的思想结构，写出相关断言 |
|  |  |

# 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）

配置环境时遇到git的使用不太懂，先粗略学习了git命令行的常见用法，才开始实验。

第一个小实验是对魔方的检验，这里学到了try catch处理异常的方法，这是以前的编程语言没有的；

第二个小实验深刻理解了java面向对象的特点，画图的实现也特别有趣味性，凸包算法也十分精妙，以及使用junit对代码进行测试，这是工程中必不可少的一项能力；

第三个小实验中对java的集合类有了更进一步地掌握：Map/ArrayList/Queue.....

感觉对java有了大致了解

## 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？

合，java面向对象的特点能帮助我们增强代码的灵活性和可重用性；java具有跨平台的特点，号称编译一次，到处运行；同时，它的安全性也很高，java通过避免采用显示的指针以减少安全攻击和风险。

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

本次实验我使用的是idea，我觉得这是一款非常美观/简洁/功能强大的ide，它有着许多插件供我们使用，自动排版和生成相对应语句也十分方便；

没有使用过eclipse，不予评价

1. 关于Git和GitHub，是否感受到了它在版本控制方面的价值；

感受到了，通过这几个小实验，我知道不断的修改代码，提交，是一种很好的版本控制手段，在大的工程项目中，多人协作是不可避免的，此时，版本控制就显得十分重要，git的对项目进行修改/提交/删除等等，在github上的远程仓库上能清晰的看出什么人在什么时间对哪段代码进行修改，这对版本控制起着相当大的作用。

1. 关于CMU和MIT的作业，你有何感受；

真的是精品，实验设置的循序渐进，完成试验后，对java最突出的特点有了进一步深入的了解。

1. 关于本实验的工作量、难度、deadline；

工作量不小，难度对我这个初学者来说也不小，但好在deadline设置时间较为合理

1. 关于初接触“软件构造”课程；

非常不错，学到了许多有用的知识，对实践能力有很大提高，目前来说印象最深刻的是java和git还有junit测试