

Lab Manuals for  
**Software Construction**

**Lab-2  
Abstract Data Type (ADT) and Object-Oriented Programming (OOP)**



Faculty of Computing

Harbin Institute of Technology

Spring 2022

**目录**

[1 实验目标 1](#_Toc72249311)

[2 实验环境 1](#_Toc72249312)

[3 实验要求 2](#_Toc72249313)

[3.1 Poetic Walks (MIT) 2](#_Toc72249314)

[3.2 Re-implement the Social Network in Lab1 3](#_Toc72249315)

[4 实验报告 3](#_Toc72249316)

[5 提交方式 4](#_Toc72249317)

[6 评分方式 4](#_Toc72249318)

# 实验目标

本次实验训练抽象数据类型（ADT）的设计、规约、测试，并使用面向对象编程（OOP）技术实现ADT。具体来说：

* 针对给定的应用问题，从问题描述中识别所需的ADT；
* 设计ADT规约（pre-condition、post-condition）并评估规约的质量；
* 根据ADT的规约设计测试用例；
* ADT的泛型化；
* 根据规约设计ADT的多种不同的实现；针对每种实现，设计其表示（representation）、表示不变性（rep invariant）、抽象过程（abstraction function）
* 使用OOP实现ADT，并判定表示不变性是否违反、各实现是否存在表示泄露（rep exposure）；
* 测试ADT的实现并评估测试的覆盖度；
* 使用ADT及其实现，为应用问题开发程序；
* 在测试代码中，能够写出testing strategy并据此设计测试用例。

# 实验环境

实验环境设置请参见Lab-0实验指南。

除此之外，本次实验需要你在Eclipse IDE中安装配置EclEmma（一个用于统计JUnit测试用例的代码覆盖度的plugin）。请访问<http://www.eclemma.org>，了解EclEmma并学习其安装、配置和使用。(使用Jetbrains IDEA中的测试用例代码覆盖度插件亦可)

本次实验在GitHub Classroom中的URL地址为：

<https://classroom.github.com/a/NL2TjK2z>

请访问该URL，按照提示建立自己的Lab2仓库并关联至自己的学号。

本地开发时，本次实验只需建立一个项目，统一向GitHub仓库提交。实验包含的两个任务分别在不同的目录内开发，具体目录组织方式参见各任务最后一部分的说明。请务必遵循目录结构，以便于教师/TA进行测试。

# 实验要求

针对以下所有两个任务，请为每个你设计和实现的ADT撰写mutability/immutability说明、AF、RI、safety from rep exposure。给出各ADT中每个方法的spec。为每个ADT编写测试用例，并写明testing strategy。

## Poetic Walks (MIT)

请阅读<http://web.mit.edu/6.031/www/sp17/psets/ps2/>，遵循该页面内的要求完成编程任务。

* 在Get the code步骤中，你无法连接MIT的Athena服务器，请从以下地址获取初始代码：  
  <https://github.com/rainywang/Spring2022_HITCS_SC_Lab2/tree/master/P1>
* 在作业描述中若遇到“commit and push”的要求，请将你的代码push到你的GitHub Lab2仓库中。
* MIT作业页面提及的文件路径，请按照下表的目录结构进行调整。例如“test/poet”应为“test/P1/poet”，“src/poet”应为“src/P1/poet”。
* 其他步骤请遵循MIT作业页面的要求。

**项目的目录结构：**

项目名称： Lab2-学号

src

P1

graph

….java

poet

….java

….txt

test

P1

graph

…Test.java

poet

…Test.java

….txt

请使用git指令将符合上述结构的代码push到你的GitHub Lab2仓库中。

## Re-implement the Social Network in Lab1

回顾Lab1实验手册中的3.2节Social Network，你针对所提供的客户端代码实现了FriendshipGraph类和Person类。

在本次实验中，请基于你在3.1节Poetic Walks中定义的Graph<L>及其两种实现，重新实现Lab1中3.3节的FriendshipGraph类。

**注1：**可以忽略你在Lab1中实现的代码，无需在其基础上实现本次作业；

**注2：**在本节FriendshipGraph中，图中的节点仍需为Person类型。故你的新FriendshipGraph类要利用3.1节已经实现的ConcreteEdgesGraph<L>或ConcreteVerticesGraph<L>，L替换为Person。根据Lab1的要求，FriendshipGraph中应提供addVertex()、addEdge()和getDistance()三个方法：针对addVertex()和addEdge()，你需要尽可能复用ConcreteEdgesGraph<L>或ConcreteVerticesGraph<L>中已经实现的add()和set()方法，而不是从0开始写代码实现或者把你的Lab1相关代码直接复制过来；针对getDistance()方法，请基于你所选定的ConcreteEdgesGraph<L>或ConcreteVerticesGraph<L>的rep来实现，而不能修改其rep。

**注3：**不变动Lab1的3.3节给出的客户端代码（例如main()中的代码），即同样的客户端代码仍可运行。重新执行你在Lab1里所写的JUnit测试用例，测试你在本实验里新实现的FriendshipGraph类仍然表现正常。

**项目的目录结构：**

项目名称： HIT-Lab2-学号

src

P2

FriendshipGraph.java

Person.java

...

test

P2

FriendshipGraphTest.java

...

(无需将3.1节实现的Graph<L>的程序源文件复制到P2目录下)

请使用git指令将符合上述结构的代码push到你的GitHub Lab2仓库中。

# 实验报告

针对上述2个编程题目，请遵循**报告模板**，撰写简明扼要的实验报告。

**实验报告的目的是记录你的实验过程，尤其是遇到的困难与解决的途径。**不需要长篇累牍，记录关键要点即可，但需确保报告覆盖了本次实验的所有开发任务。

注意：

* 实验报告不需要包含所有源代码，请根据上述目的有选择的加入关键源代码，作为辅助说明。
* 请确保报告格式清晰、一致，故请遵循目前模板里设置的字体、字号、行间距、缩进；
* 实验报告提交前，请“目录”上右击，然后选择“更新域”，以确保你的目录标题/页码与正文相对应。
* 实验报告文件可采用Word或PDF格式，命名规则：Lab2-学号-Report。

# 提交方式

**截止日期：**第14周周日夜间23:55。

**源代码：**从本地Git仓库推送至个人GitHub的Lab2仓库内。

**实验报告：**随代码仓库（doc）目录提交至GitHub。

# 评分方式

Deadline之后，教师和TA对学生在GitHub上的代码进行测试、阅读实验报告，做出相应评分。