

**2024年春季学期  
计算学部《软件构造》课程**

**Lab 2实验报告**

|  |  |
| --- | --- |
| 姓名 |  |
| 学号 |  |
| 班号 |  |
| 电子邮件 |  |
| 手机号码 |  |

**目录**

[1 实验目标概述 1](#_Toc97648181)

[2 实验环境配置 1](#_Toc97648182)

[3 实验过程 1](#_Toc97648183)

[3.1 Poetic Walks 1](#_Toc97648184)

[3.1.1 Get the code and prepare Git repository 1](#_Toc97648185)

[3.1.2 Problem 1: Test Graph <String> 1](#_Toc97648186)

[3.1.3 Problem 2: Implement Graph <String> 1](#_Toc97648187)

[3.1.3.1 Implement ConcreteEdgesGraph 2](#_Toc97648188)

[3.1.3.2 Implement ConcreteVerticesGraph 2](#_Toc97648189)

[3.1.4 Problem 3: Implement generic Graph<L> 2](#_Toc97648190)

[3.1.4.1 Make the implementations generic 2](#_Toc97648191)

[3.1.4.2 Implement Graph.empty() 2](#_Toc97648192)

[3.1.5 Problem 4: Poetic walks 2](#_Toc97648193)

[3.1.5.1 Test GraphPoet 2](#_Toc97648194)

[3.1.5.2 Implement GraphPoet 2](#_Toc97648195)

[3.1.5.3 Graph poetry slam 2](#_Toc97648196)

[3.1.6 使用Eclemma检查测试的代码覆盖度 2](#_Toc97648197)

[3.1.7 Before you’re done 2](#_Toc97648198)

[3.2 Re-implement the Social Network in Lab1 2](#_Toc97648199)

[3.2.1 FriendshipGraph类 2](#_Toc97648200)

[3.2.2 Person类 3](#_Toc97648201)

[3.2.3 客户端main() 3](#_Toc97648202)

[3.2.4 测试用例 3](#_Toc97648203)

[3.2.5 提交至Git仓库 3](#_Toc97648204)

[4 实验进度记录 3](#_Toc97648205)

[5 实验过程中遇到的困难与解决途径 3](#_Toc97648206)

[6 实验过程中收获的经验、教训、感想 4](#_Toc97648207)

[6.1 实验过程中收获的经验和教训（必答） 4](#_Toc97648208)

[6.2 针对以下方面的感受（必答） 4](#_Toc97648209)

# 实验目标概述

根据实验手册简要撰写。

# 实验环境配置

简要陈述你配置本次实验所需环境的过程，必要时可以给出屏幕截图。

特别是要记录配置过程中遇到的问题和困难，以及如何解决的。

在这里给出你的GitHub Lab2仓库的URL地址（HIT-Lab2-学号）。

# 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（但千万不要把你的源代码全部粘贴过来！）。

## Poetic Walks

在这里简要概述你对该任务的理解。

### Get the code and prepare Git repository

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。

### Problem 1: Test Graph <String>

以下各部分，请按照MIT页面上相应部分的要求，逐项列出你的设计和实现思路/过程/结果。

### Problem 2: Implement Graph <String>

以下各部分，请按照MIT页面上相应部分的要求，逐项列出你的设计和实现思路/过程/结果。

#### Implement ConcreteEdgesGraph

#### Implement ConcreteVerticesGraph

### Problem 3: Implement generic Graph<L>

#### Make the implementations generic

#### Implement Graph.empty()

### Problem 4: Poetic walks

#### Test GraphPoet

#### Implement GraphPoet

#### Graph poetry slam

### 使用Eclemma/Code Coverage for Java检查测试的代码覆盖度

### Before you’re done

请按照[http://web.mit.edu/6.031/www/sp17/psets/ps2/#before\_youre\_done](http://web.mit.edu/6.031/www/sp17/psets/ps2/" \l "before_youre_done)的说明，检查你的程序。

如何通过Git提交当前版本到GitHub上你的Lab2仓库。

在这里给出你的项目的目录结构树状示意图。

## Re-implement the Social Network in Lab1

在这里简要概述你对该任务的理解。

### FriendshipGraph类

给出你的设计和实现思路/过程/结果。

### Person类

给出你的设计和实现思路/过程/结果。

### 客户端main()

给出你的设计和实现思路/过程/结果。

### 测试用例

给出你的设计和实现思路/过程/结果。

### 提交至Git仓库

如何通过Git提交当前版本到GitHub上你的Lab3仓库。

在这里给出你的项目的目录结构树状示意图。

# 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 | 计划任务 | 实际完成情况 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的难点 | 解决途径 |
|  |  |
|  |  |
|  |  |

# 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）

## 针对以下方面的感受（必答）

1. 面向ADT的编程和直接面向应用场景编程，你体会到二者有何差异？
2. 使用泛型和不使用泛型的编程，对你来说有何差异？
3. 在给出ADT的规约后就开始编写测试用例，优势是什么？你是否能够适应这种测试方式？
4. P1设计的ADT在多个应用场景下使用，这种复用带来什么好处？
5. 为ADT撰写specification, invariants, RI, AF，时刻注意ADT是否有rep exposure，这些工作的意义是什么？你是否愿意在以后编程中坚持这么做？
6. 关于本实验的工作量、难度、deadline。
7. 《软件构造》课程进展到目前，你对该课程有何收获和建议？