

**2021年春季学期  
计算学部《软件构造》课程**

**Lab 3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](#_Toc71115456)

[2 实验环境配置 1](#_Toc71115457)

[3 实验过程 1](#_Toc71115458)

[3.1 待开发的三个应用场景 1](#_Toc71115459)

[3.2 面向可复用性和可维护性的设计：IntervalSet<L> 1](#_Toc71115460)

[3.2.1 IntervalSet<L>的共性操作 1](#_Toc71115461)

[3.2.2 局部共性特征的设计方案 2](#_Toc71115462)

[3.2.3 面向各应用的IntervalSet子类型设计（个性化特征的设计方案） 2](#_Toc71115463)

[3.3 面向可复用性和可维护性的设计：MultiIntervalSet<L> 2](#_Toc71115464)

[3.3.1 MultiIntervalSet<L>的共性操作 2](#_Toc71115465)

[3.3.2 局部共性特征的设计方案 2](#_Toc71115466)

[3.3.3 面向各应用的MultiIntervalSet子类型设计（个性化特征的设计方案） 2](#_Toc71115467)

[3.4 面向复用的设计：L 2](#_Toc71115468)

[3.5 可复用API设计 2](#_Toc71115469)

[3.5.1 计算相似度 2](#_Toc71115470)

[3.5.2 计算时间冲突比例 2](#_Toc71115471)

[3.5.3 计算空闲时间比例 2](#_Toc71115472)

[3.6 应用设计与开发 2](#_Toc71115473)

[3.6.1 排班管理系统 3](#_Toc71115474)

[3.6.2 操作系统的进程调度管理系统 3](#_Toc71115475)

[3.6.3 课表管理系统 3](#_Toc71115476)

[3.7 基于语法的数据读入 3](#_Toc71115477)

[3.8 应对面临的新变化 3](#_Toc71115478)

[3.8.1 变化1 3](#_Toc71115479)

[3.8.2 变化2 3](#_Toc71115480)

[3.9 Git仓库结构 3](#_Toc71115481)

[4 实验进度记录 3](#_Toc71115482)

[5 实验过程中遇到的困难与解决途径 4](#_Toc71115483)

[6 实验过程中收获的经验、教训、感想 4](#_Toc71115484)

[6.1 实验过程中收获的经验和教训 4](#_Toc71115485)

[6.2 针对以下方面的感受 4](#_Toc71115486)

# 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 Lab3仓库的URL地址（Lab3-学号）。

# 实验过程

请仔细对照实验手册，针对每一项任务，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！）。

## 待开发的三个应用场景

简要介绍三个应用。

分析三个应用场景的异同，理解需求：它们在哪些方面有共性、哪些方面有差异。

## 面向可复用性和可维护性的设计：IntervalSet<L>

该节是本实验的核心部分。

### IntervalSet<L>的共性操作

### 局部共性特征的设计方案

### 面向各应用的IntervalSet子类型设计（个性化特征的设计方案）

## 面向可复用性和可维护性的设计：MultiIntervalSet<L>

### MultiIntervalSet<L>的共性操作

### 局部共性特征的设计方案

### 面向各应用的MultiIntervalSet子类型设计（个性化特征的设计方案）

## 面向复用的设计：L

## 可复用API设计

### 计算相似度

### 计算时间冲突比例

### 计算空闲时间比例

## 应用设计与开发

利用上述设计和实现的ADT，实现手册里要求的各项功能。

### 排班管理系统

### 操作系统的进程调度管理系统

### 课表管理系统

## 基于语法的数据读入

修改“排班管理”应用以扩展该功能。

## 应对面临的新变化

### 变化1

评估之前的设计是否可应对变化、代价如何

如何修改设计以应对变化

### 变化2

评估之前的设计是否可应对变化、代价如何

如何修改设计以应对变化

## Git仓库结构

请在完成全部实验要求之后，利用Git log指令或Git图形化客户端或GitHub上项目仓库的Insight页面，给出你的仓库到目前为止的Object Graph，尤其是区分清楚change分支和master分支所指向的位置。

# 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

## 针对以下方面的感受

1. 重新思考Lab2中的问题：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？本实验设计的ADT在五个不同的应用场景下使用，你是否体会到复用的好处？
2. 重新思考Lab2中的问题：为ADT撰写复杂的specification, invariants, RI, AF，时刻注意ADT是否有rep exposure，这些工作的意义是什么？你是否愿意在以后的编程中坚持这么做？
3. 之前你将别人提供的API用于自己的程序开发中，本次实验你尝试着开发给别人使用的API，是否能够体会到其中的难处和乐趣？
4. 你之前在使用其他软件时，应该体会过输入各种命令向系统发出指令。本次实验你开发了一个解析器，使用语法和正则表达式去解析输入文件并据此构造对象。你对语法驱动编程有何感受？
5. Lab1和Lab2的大部分工作都不是从0开始，而是基于他人给出的设计方案和初始代码。本次实验是你完全从0开始进行ADT的设计并用OOP实现，经过五周之后，你感觉“设计ADT”的难度主要体现在哪些地方？你是如何克服的？
6. “抽象”是计算机科学的核心概念之一，也是ADT和OOP的精髓所在。本实验的五个应用既不能完全抽象为同一个ADT，也不是完全个性化，如何利用“接口、抽象类、类”三层体系以及接口的组合、类的继承、设计模式等技术完成最大程度的抽象和复用，你有什么经验教训？
7. 关于本实验的工作量、难度、deadline。
8. 到目前为止你对《软件构造》课程的评价。