

**2022年春季学期  
计算学部《软件构造》课程**

**Lab 3实验报告**

|  |  |
| --- | --- |
| 姓名 | 瞿久尧 |
| 学号 | 120L022314 |
| 班号 | 2003012 |
| 电子邮件 | 2668136485@qq.com |
| 手机号码 | 15390311738 |

**目录**

[1 实验目标概述 1](#_Toc102826659)

[2 实验环境配置 1](#_Toc102826660)

[3 实验过程 1](#_Toc102826661)

[3.1 待开发的三个应用场景 1](#_Toc102826662)

[3.2 ADT识别与设计 1](#_Toc102826663)

[3.2.1 任务1：投票类型VoteType 1](#_Toc102826664)

[3.2.2 任务2：投票项VoteItem<C> 1](#_Toc102826665)

[3.2.3 任务3：选票Vote 2](#_Toc102826666)

[3.2.4 任务4：投票活动Poll<C>的测试 2](#_Toc102826667)

[3.2.5 任务5：投票活动Poll<C>的实现类GeneralPollImpl 2](#_Toc102826668)

[3.2.6 任务6：投票活动Poll<C>的子类型 2](#_Toc102826669)

[3.3 ADT行为的设计与实现 2](#_Toc102826670)

[3.3.1 任务7：合法性检查 2](#_Toc102826671)

[3.3.2 任务8：采用Strategy设计模式实现灵活的计票规则 2](#_Toc102826672)

[3.3.3 任务9：采用Strategy设计模式实现灵活的遴选规则 2](#_Toc102826673)

[3.3.4 任务10：处理匿名和实名投票 2](#_Toc102826674)

[3.3.5 任务11：采用Visitor设计模式实现功能扩展 2](#_Toc102826675)

[3.3.6 任务12：基于语法的数据读入 2](#_Toc102826676)

[3.4 任务13：应用设计与开发 2](#_Toc102826677)

[3.4.1 商业表决系统 2](#_Toc102826678)

[3.4.2 代表选举系统 2](#_Toc102826679)

[3.4.3 聚餐点菜系统 2](#_Toc102826680)

[3.5 任务14：应对面临的新变化 2](#_Toc102826681)

[3.5.1 商业表决应用：可以一次表决多个商业提案 2](#_Toc102826682)

[3.5.2 代表选举应用：遴选规则变化 3](#_Toc102826683)

[3.5.3 聚餐点菜应用：取消权重设置、只计算“喜欢”的票数 3](#_Toc102826684)

[3.6 Git仓库结构 3](#_Toc102826685)

[4 实验进度记录 3](#_Toc102826686)

[5 实验过程中遇到的困难与解决途径 3](#_Toc102826687)

[6 实验过程中收获的经验、教训、感想 4](#_Toc102826688)

[6.1 实验过程中收获的经验和教训（必答） 4](#_Toc102826689)

[6.2 针对以下方面的感受（必答） 4](#_Toc102826690)

# 实验目标概述

目标是编写具有可复用性和可维护性的软件，主要使用以下软件构造技术：

* 子类型、泛型、多态、重写、重载
* 继承、委派、CRP
* 语法驱动的编程、正则表达式
* 设计模式

本次实验给定了多个具体应用，学生不是直接针对每个应用分别编程实现，而是通过ADT和泛型等抽象技术，开发一套可复用的ADT及其实现，充分考虑这些应用之间的相似性和差异性，使ADT有更大程度的复用（可复用性）和更容易面向各种变化（可维护性）。

# 实验环境配置

实验环境设置请参见Lab-0实验指南。

本次实验在GitHub Classroom中的URL地址为：

<https://classroom.github.com/a/ZAJ8w2eC>

请访问该URL，按照提示建立自己的Lab3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。实验包含的多个任务分别在不同的包内开发，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。

在这里给出你的GitHub Lab3仓库的URL地址（HIT-Lab3-学号）。

[ComputerScienceHIT/HIT-Lab3-120L022314: HIT-Lab3-120L022314 created by GitHub Classroom](https://github.com/ComputerScienceHIT/HIT-Lab3-120L022314)

（HIT-Lab3-120L022314）

# 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

1. 商业表决

在某公司中，需要各股东进行投票，而各股东占据不同的股权，只有当投支持票的股东所持股权比例总和超过2/3才算通过。

因此主要是需要考虑投票加权的问题

1. 代表选举

是一群投票人对几个候选人进行投票，每个人有支持/反对/弃权的权力，但是投支持票的数量不能超过一定限额，最后根据支持票的数量进行排序选举。

主要是考虑多对多以及投票数量超过后不计票的问题。

1. 聚餐点菜

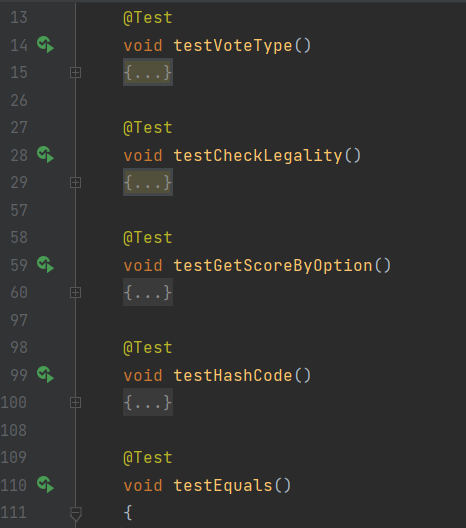
我认为可以理解为以上两个应用场景的合并，需要考虑加权、限制、多对多的投票以及统计。

## ADT识别与设计

该节是本实验的核心部分。

### 任务1：投票类型VoteType

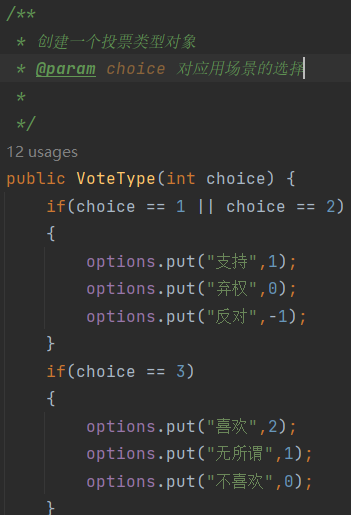
该ADT用于管理一次投票活动中允许使用的合法投票选项，是Immutable的。



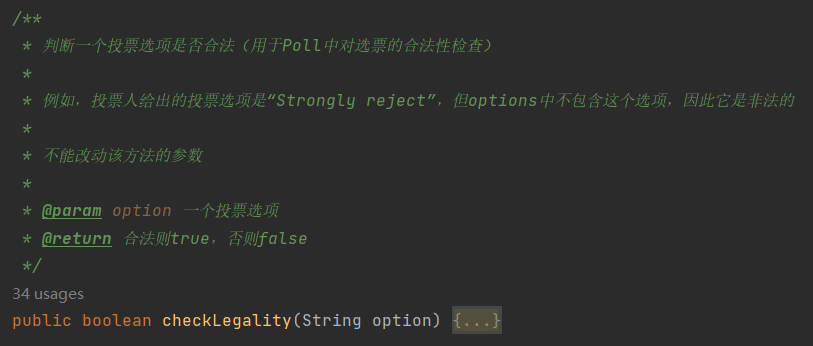
需要在完善ADT之前先将测试写好，并且在完善ADT之后测试通过。



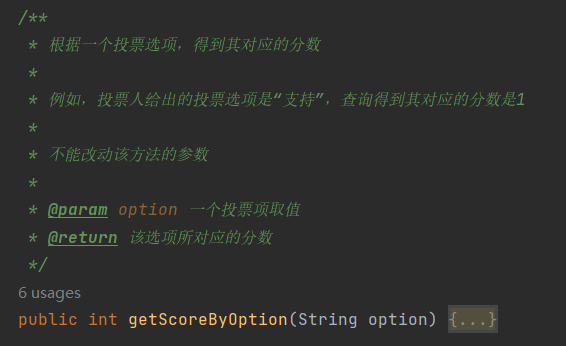
在完善ADT的时候先将RI,AF,SRE写清楚，并根据rep写好checkRep



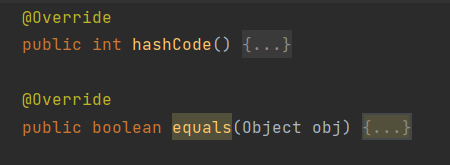
根据应用场景划分创建的对象



只需要检查option是否在构建对象中的options中即可



根据在构造时创建好的options，搜索key查找键值即可



尽量重写两个方法

### 任务2：投票项VoteItem<C>

VoteItem<C>是一个immutable的ADT，描述了一个“投票项”，表征了一个投票人对一个候选对象的具体评价。该ADT内部的数据和行为包括：

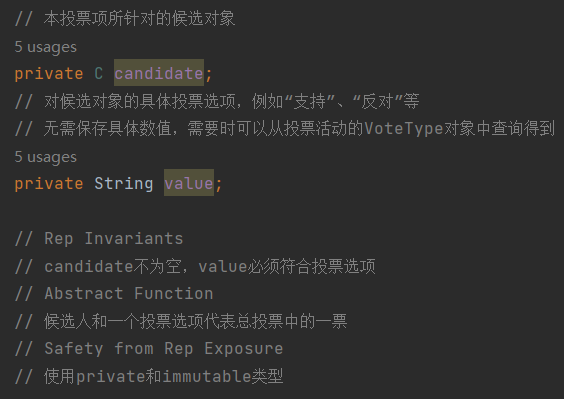
【数据】候选对象（类型为C）

【数据】投票选项（例如“支持”）。

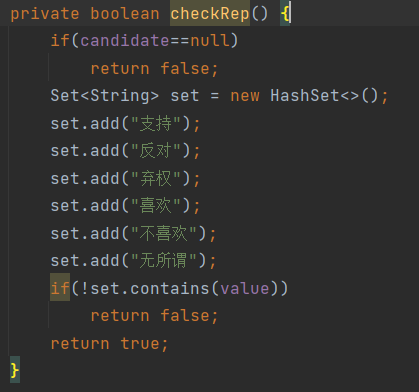
【行为】创建投票记录



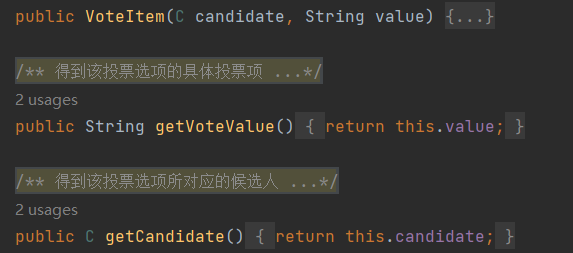
编写测试用例并在之后通过测试，testVoteItem中已经测试了三个方法



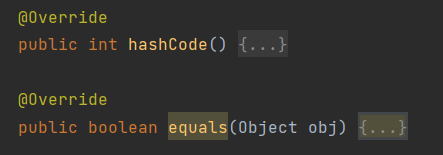
根据spec和rep完善RI,AF, Safety from Rep Exposure



写出checkRep



此三个方法已经给出，一个构造函数和两个观察器



最后重写这两个方法即可

### 任务3：选票Vote

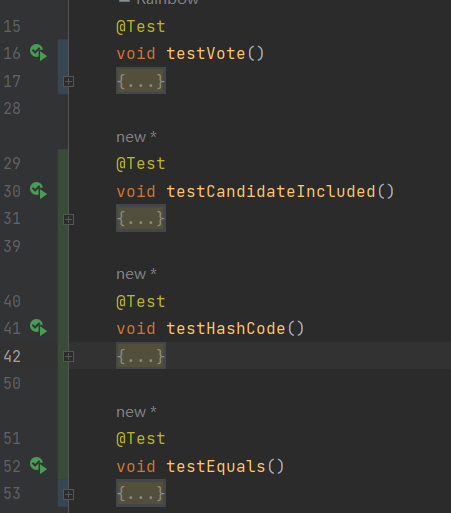
这是代表一个投票人对所有候选对象的投票项的聚合体，是Immutable的。该ADT内部应维护的数据和行为包括：

【数据】对所有候选对象的投票记录（相当于多个VoteItem对象）

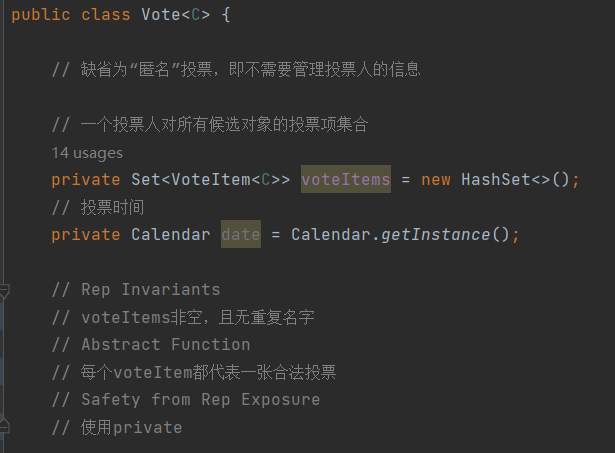
【数据】投票时间

【行为】创建选票

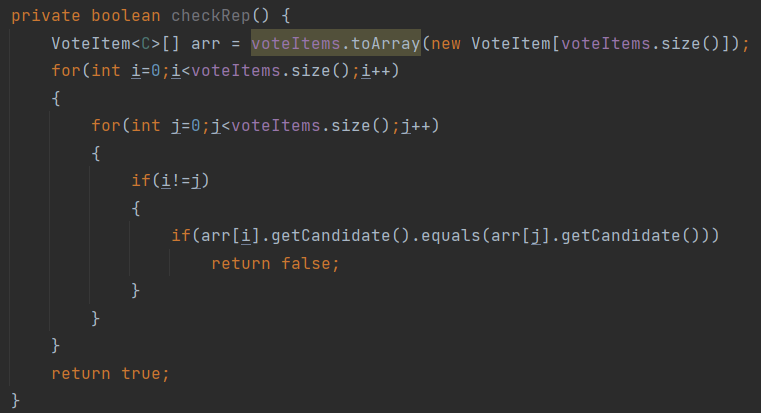
【行为】查询该选票中包含的所有投票项



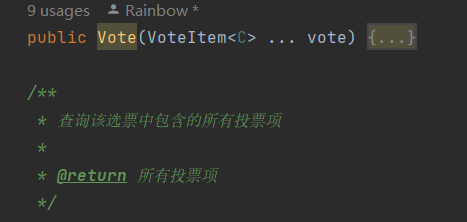
编写测试，并在之后通过测试



根据Vote所给的Rep以及Vote的功能完善RI、AF、Safety from Rep Exposure



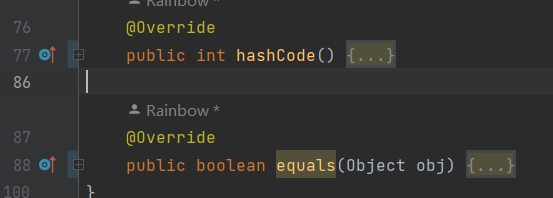
根据所写的RI完善checkRep()



因为参数不确定，所以需要写参数未定的构造函数



根据spec完善两个方法函数



重写hashCode和equals方法

### 任务4：投票活动Poll<C>的测试

### 任务5：投票活动Poll<C>的实现类GeneralPollImpl

### 任务6：投票活动Poll<C>的子类型

## ADT行为的设计与实现

### 任务7：合法性检查

### 任务8：采用Strategy设计模式实现灵活的计票规则

### 任务9：采用Strategy设计模式实现灵活的遴选规则

### 任务10：处理匿名和实名投票

### 任务11：采用Visitor设计模式实现功能扩展

### 任务12：基于语法的数据读入

## 任务13：应用设计与开发

### 商业表决系统

### 代表选举系统

### 聚餐点菜系统

## 任务14：应对面临的新变化

### 商业表决应用：可以一次表决多个商业提案

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

### 代表选举应用：遴选规则变化

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

### 聚餐点菜应用：取消权重设置、只计算“喜欢”的票数

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（必答）

## 针对以下方面的感受（必答）

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在三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ADT/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ADT/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简单的解析器，使用语法和正则表达式去解析一个遵循特定规则的字符串并据此构造对象。你对语法驱动编程有何感受？
5. Lab1和Lab2的工作都不是从0开始，而是基于他人给出的设计方案和初始代码。本次实验中也提供了一部分基础代码。假如本实验要求你完全从0开始进行ADT的设计并用OOP实现，你觉得自己是否能够完全搞定？你认为“设计ADT”的难度主要体现在哪些地方？
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。本实验的三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、委派、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. 课程结束了，你对《软件构造》课程内容和任课教师的评价如何？