

内部资料，请勿发布到互联网上，违者必究

Lab2 Reusability and Maintainability Oriented Software Construction

School of Computer Science and Technology

Harbin Institute of Technology

Spring 2024

1. 实验目标

本次实验涵盖课程授课的第 5、6、7 和 8 内容。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：

- 使用 ADT 及其实现，为应用问题开发程序；
- 子类型、泛型、多态、重写、重载
- 继承、代理、组合
- 面向对象的七原则：SOLID+2EX

2. 实验环境

实验环境设置请参见 Lab-0 实验指南。

3. 实验要求

本次实验给出两个题目，均为开放型题目，没有标准答案，其中 3.2 为选做题目。判断是否设计良好的标准有两个必要条件：其一，必须满足开闭原则(Open-Closed Principle, OCP)；其二，必须能够复用 ReUse。这两个条件没有做到，教师认为你的设计及实现就不具备可复用性和可维护性。

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。

利用教师给定的 StarUML 工具，绘制设计类图。在设计时，充分考虑可复用性和可维护性。需要提醒各位同学：

(1) Windows 用户必须使用教师提供的 StarUML 工具，否则在撰写实验报告时做不到直接从 StarUML 中 copy，并 paste 到实验报告中，切记不得在 StarUML 中截图，不遵守该规定者，格式分零分；

(2) Mac 和其他操作系统的用户，建议采用 Microsoft office 系列的 Visio 绘图工具，该工具也能直接将绘制的类图复制并拷贝到 word 中

(3) 也可以采用其他 UML 绘图工具，只要能直接 copy 类图，并粘贴到 word 格式的实验中即可。

不要改变报告的格式，此外，为了减少报告厚度，请正反面打印。

3.1. 待开发的应用场景-文件解析（必做）

在软件开发过程中，为了实现系统的灵活性，经常采用配置文件的方式对系统进行配置。现有某项目要进行文件解析，需要根据文件的类型，去使用不同的文件解析器 Parser。具体如下：

- Json 类型的文件需要用 Json 解析器 JsonTypeParser
- Xml 类型文件使用 Xml 解析器 XmlTypeParser。
- ...

请按照如下小节的顺序和要求，完成本应用场景的设计与实现，并进行测试。

3.1.1. 设计

需要注意的是，设计的过程中，需要充分考虑未来存在的多种类型的配置文件的可能性，

例如 .yaml 格式, yml 格式, .properties 等数据格式, 需要能够应对变化。

对你的类图进行说明, 可以采用文字描述、类图扩展等方式, 说明你的设计能够复用并满足 OCP。

3.1.2. 实现

在实现过程中, 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。在实际操作中, ADT 对应接口或者抽象类, 因此相关的 spec 定义应该在此编写; 而 AF、RI、safety from rep exposure 需要在抽象类的子类、接口的实现类中编写。

3.1.3. 测试

为每个 ADT 编写测试用例, 并写明 testing strategy。具体的编写方法与策略, 请仔细阅读 MIT 的阅读材料: <https://web.mit.edu/6.031/www/sp21/classes/03-testing/>

3.2. 待开发的应用场景 (选做)

基于 3.1 的开发场景, 需要根据文件环境来分类。在测试环境 (testing environment) 下, 所有的格式都要被解析, 而在某生产环境 (production environment) 中, 只需要解析 Json 和 XML。

请按照如下小节的顺序和要求, 完成本应用场景的设计与实现, 并进行测试。

3.2.1. 设计

同 3.1.1 一致, 设计的过程中, 需要充分考虑未来存在的多种类型的配置文件的可能性, 例如 .yaml 格式, yml 格式, .properties 等数据格式, 需要能够应对变化。对你的类图进行说明, 可以采用文字描述、类图扩展等方式, 说明你的设计能够复用并满足 OCP。

需要注意的是, 在 Lec08-Design Patterns for Reuse and Maintainability Part ① 讲义中, 采用抽象工厂的方法实现了生产多个等级的产品, 但抽象工厂并不满足 OCP。因此, 需要你充分考虑如何才能满足 OCP

3.2.2. 实现

在实现过程中, 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。在实际操作中, ADT 对应接口或者抽象类, 因此相关的 spec 定义应该在此编写; 而 AF、RI、safety from rep exposure 需要在抽象类的子类、接口的实现类中编写。

3.2.3. 测试

为每个 ADT 编写测试用例, 并写明 testing strategy。具体的编写方法与策略, 请仔细阅读 MIT 的阅读材料: <https://web.mit.edu/6.031/www/sp21/classes/03-testing/>

3.3. 待开发的应用场景-饮料计费系统 (必做)

ReFine 咖啡店在全球范围内迅速发展的一个著名的咖啡馆。因为它发展的很快, 为了适应饮料消费量, 它急需更新结账系统。

ReFine 卖的咖啡种类有: Espresso、Decaf、DarkRost 和 HouseBlend 等, 为了满足客户的口味, 调料包括 Steamed Milk (蒸过的牛奶)、Soy (豆酱)、Mocha (糊状巧克力)、Whip (生牛奶) 等调料。

在更新后的计费系统中, 咖啡和调料都需要收费。即咖啡的价格加上调料的价格即为客户最后需要结账的费用。目前面临的问题:

- 咖啡的品种不断增加；
- 调料的品种逐步更新；
- 需求五花八门，每位顾客点的咖啡和调料不尽相同

ReFine 咖啡店希望尽快找到解决方案，实现能够满足其需求的计费系统。

需要注意的是，客户可能不加调料；也可能加一份某种调料；可能加多份同一种调料；可能加多份、多种调料。请你构建这套系统，使其能够适应咖啡和调料品种的增加，并且咖啡和调料都能计费。

3.3.1. 设计

同 3.1.1 一致，设计的过程中，需要充分考虑未来存在咖啡的变化和调料的变化，并且能够满足开闭原则和复用。

3.3.2. 实现

在实现过程中，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。在实际操作中，ADT 对应接口或者抽象类，因此相关的 spec 定义应该在此编写；而 AF、RI、safety from rep exposure 需要在抽象类的子类、接口的实现类中编写。

3.3.3. 测试

为每个 ADT 编写测试用例，并写明 testing strategy。具体的编写方法与策略，请仔细阅读 MIT 的阅读材料：<https://web.mit.edu/6.031/www/sp21/classes/03-testing/>

4. 实验报告

请遵循实验报告模板，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、美观，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。
- 电子版本的实验报告文件可采用 Word 或 PDF 格式，命名规则：Lab1-学号-Report。放到工程目录的根目录下。

5. 提交

截止日期：**第 15 周周六 17:00（2024 年 6 月 15 日）**。各班学委将纸质版报告放到 M 楼 107 教师休息室的书架上即可。电子版各班学委在第 15 周周六下午 18:00 之前将本班的电子报告打包发给老师，不用等，到点就交，根据约定，纸质版和电子版超过规定时限不接受提交。