

实验课4：开闭原则

实验课4：开闭原则

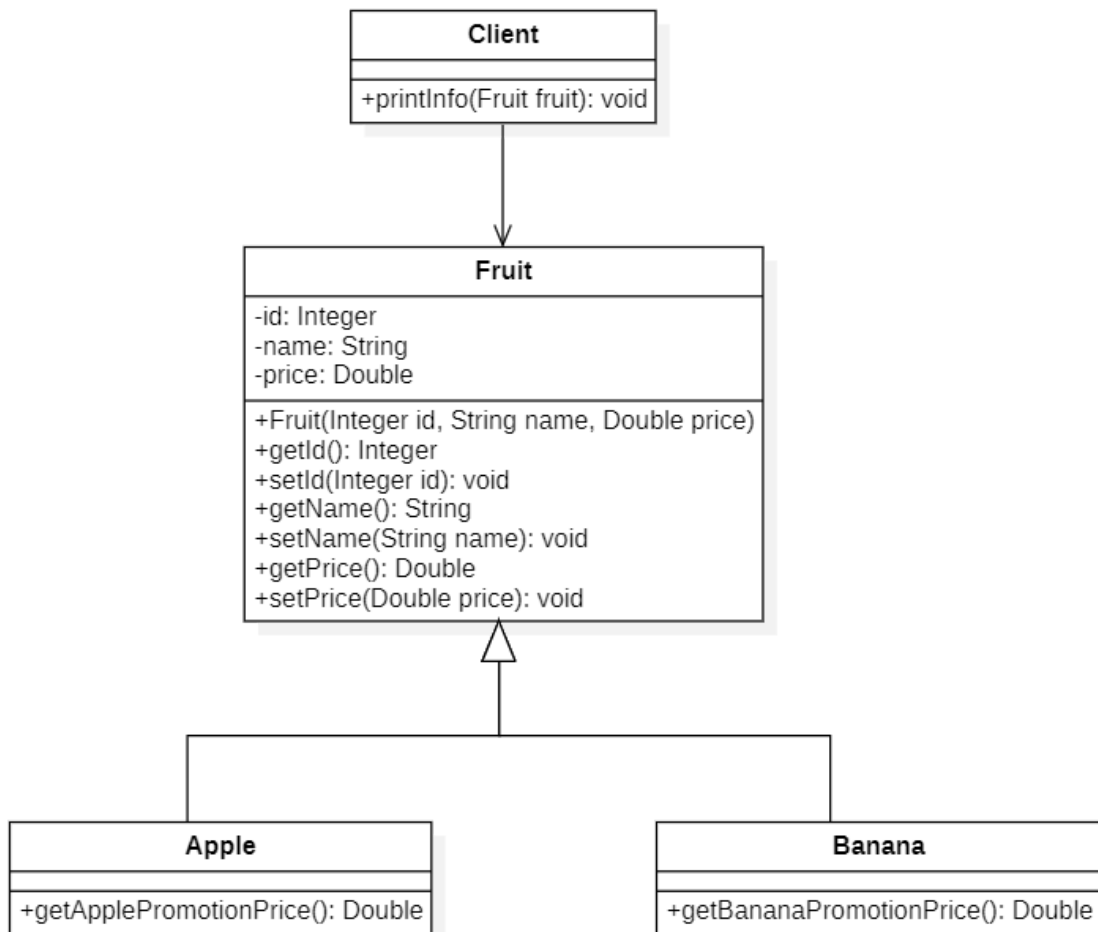
实验任务

实验任务

- 请对下面的设计进行重构，使其符合**开闭原则**。

有一家水果店，其信息系统中有个功能是**打印出某种水果的信息**，因店老板经常会搞些促销打折活动，故打印的信息中包括了**原价**也包括**促销价**。不同类型的水果折扣是不一样的，比如苹果八折，香蕉九折，且这些折扣也都是会变动的。

目前的设计是这样的：



这个设计不符合开闭原则，因为：（1）目前折扣价是写死在程序 `getXxxPromotionPrice()` 里面的，当折扣率变化时需修改原有实现代码；（2）当添加新的水果种类时，客户代码也需要修改，以判断打印的水果是不是新加的种类。

要求：

- 根据当前的设计写出代码，实现功能，例如：能正确输出如下信息：

Id: 1; 名称: 苹果; 原价格: 20.0
促销价格: 16.0

- 重构原有设计, 使其符合开闭原则, 画出类图并编码实现 (提示: 针对问题 (1), 可考虑在Fruit中设置折扣率成员属性, 并提供方法进行动态设置; 针对问题 (2), 可抽象出计算折扣的方法放在Fruit类中, 具体实现由子类来做, 当然由于这个案例过于简单, 这样做看起来也有点傻, 不过为了学习目的就将就一下吧)
- 本实验将作为一次作业在超星平台布置, 请同学们记得要提交。
- 建议同学们自行学会使用Git, 在Gitee代码托管平台创建代码仓库, 将程序代码保存在Gitee, 通过云IDE (如 <https://cloudstudio.net>) 编程并用Git管理代码库。
- (选做) 思考一下: 如果促销方式不仅仅是简单地打折, 而是还有其他方式, 比如: 单价超过50元的水果额外再多打一折; VIP会员额外又再多打一折。这种情形下, 你的重构方案是否还符合开闭原则呢? 如果不符合, 应该怎样做呢? (提示: 可考虑每种促销策略为单独的具体类, 它们继承自抽象的促销策略类/接口, 由具体促销策略类来计算促销优惠额。实际上就是设计模式的一种: 策略模式)