20XX

实际项目中的设计模式分析

授课人：胡秋琼
答辩人：翁维斌

1. 项目概述 01

2. 设计模式的应用分析 02

3技术分析 03

4. 使用心得和总结 04

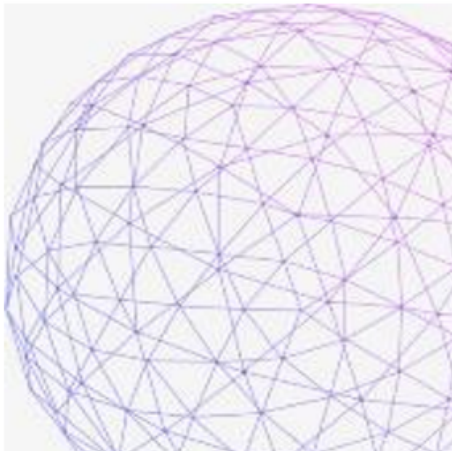
1. 项目概述

PART
1

1. 项目概述

项目的简介

设计模式在外卖营销"邀请下单"业务中设计模式的分析

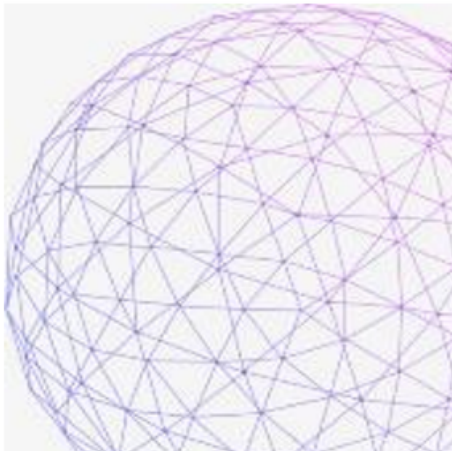


1. 项目概述

项目中用到的设计模式概述

工厂模式：用于创建不同的返奖策略对象，让子类决定实例化哪一个类

策略模式：用于定义一系列的返奖算法，将每个算法封装起来，并让它们可以互相替换



2. 设计模式的应用分析

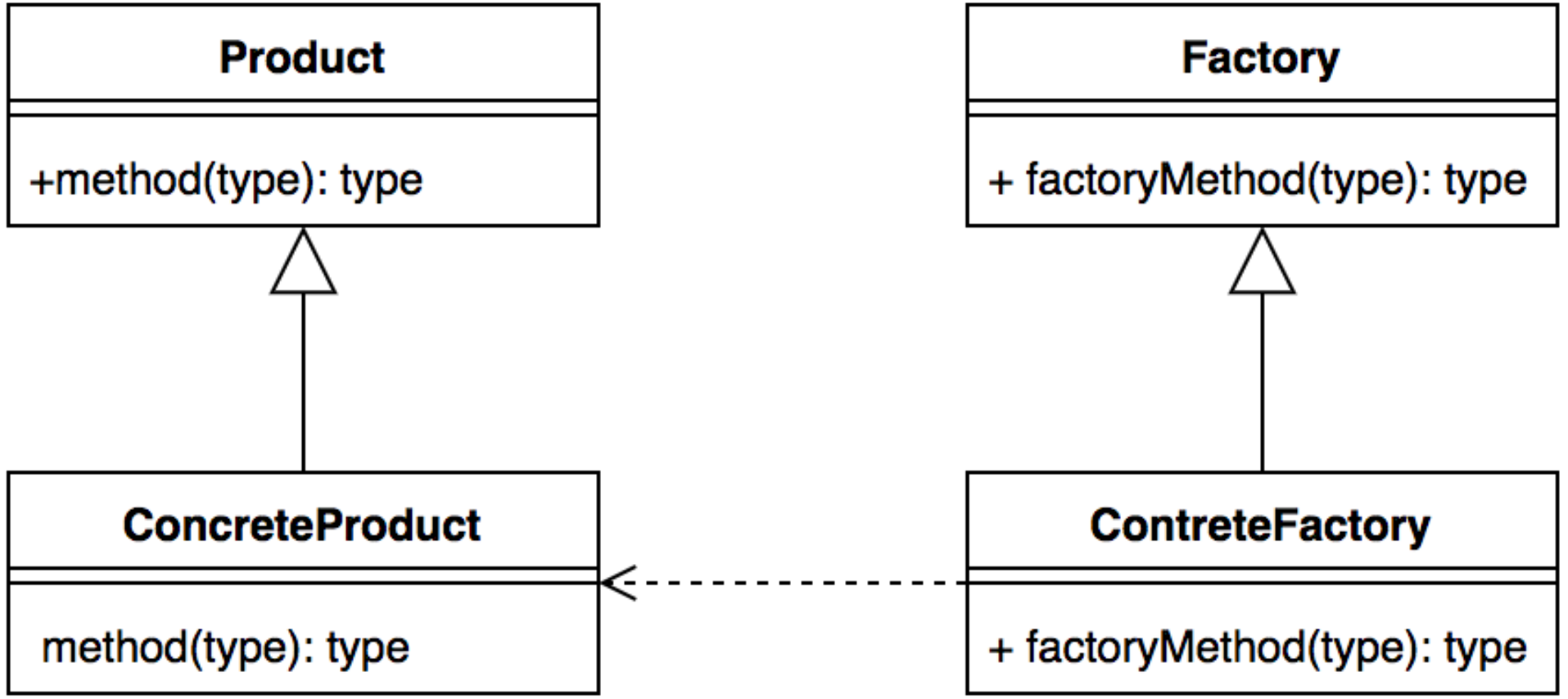
PART 2

2. 设计模式的应用分析

”邀请下单”是美团外卖用户邀请其他用户下单后给予奖励的平台。即用户A邀请用户B，并且用户B在美团下单后，给予用户A一定的现金奖励(以下简称返奖)。同时为了协调成本与收益的关系，返奖会有多个计算策略。为什么这么写，用了这个模式产生了什么样的效果或者作用

2. 设计模式的应用分析

工厂模式通用类图

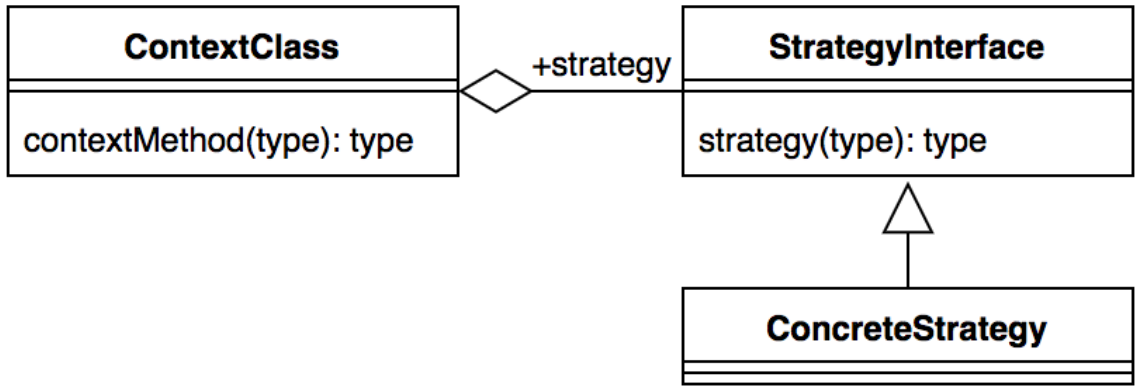


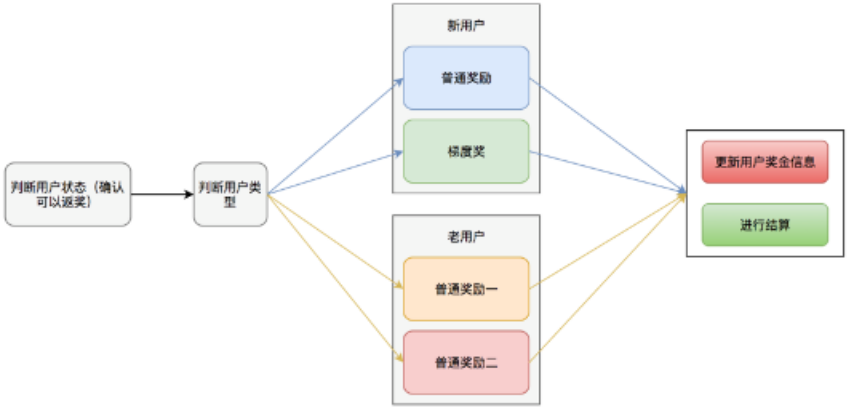
2. 设计模式的应用分析

策略模式通用类图

我们可以看到返奖的主流程就是选择不同的返奖策略的过程，每个返奖策略都包括返奖金额计算、更新用户奖金信息、以及结算这三个步骤。 我们可以使用工厂模式生产出不同的策略，同时使用策略模式来进行不同的策略执行。首先确定我们需要生成出n种不同的返奖策略，其编码如下

//抽象策略





2. 设计模式的应用分析

```
public abstract class RewardStrategy {
public abstract void reward(long userId)
public void insertRewardAndSettlement(long userId, int
reward) {} ; //更新用户信息以及结算
}
//新用户返奖具体策略A
public class newUserRewardStrategyA extends RewardStrategy
{
@Override
public void reward(long userId) {}  //具体的计算逻辑，
}
```

2. 设计模式的应用分析

```
//老用户返奖具体策略A
public class OldUserRewardStrategyA extends
RewardStrategy {
@Override
public void reward(long userId) {}  //具体的计算
逻辑,
}

//抽象工厂
public abstract class StrategyFactory<T> {
```