

[Tsubasa Plus – MD04 - S08] - BTUD - Refactor ứng dụng xét khuyến mãi

1. Mục tiêu

- Nhận biết *Smell Code*, *Clean Code*
- Refactor code

2. Mô tả

Khi viết module trong ứng dụng Java Console kiểm tra xem khách hàng có đủ điều kiện nhận khuyến mãi không, tuy nhiên do lập trình viên viết nhanh mà không chú trọng cấu trúc mã nguồn.

- Thông tin đầu vào:
 - n: Tên khách hàng (String)
 - t: Loại khách hàng (int – 1 là VIP, 2 là thường)
 - l: Điểm tích lũy (loyaltyPoints- int)
 - d: Số tiền nợ (debit - int)
- Logic yêu cầu:
 - Nếu loại khách là VIP (t==1) và có điểm tích lũy > 0 và không có nợ (d==0) thì in ra “Khách <tên> là VIP và đủ điều kiện nhận khuyến mãi.”
 - Trường hợp khách thường hoặc không có điểm hoặc có nợ thì in ra “Khách <tên> là khách thường hoặc không đủ điều kiện.”

```
public class U {  
    public void p(String n, int t, int l, int d) {  
        if (t == 1 && l > 0 && d == 0) {  
            System.out.println("Khách " + n + " là VIP và đủ điều kiện nhận khuyến mãi.");  
        } else if (t == 2 || l == 0 || d > 0) {  
            System.out.println("Khách " + n + " là khách thường hoặc không đủ điều kiện.");  
        }  
    }  
}
```

Yêu cầu:

- **Đổi tên** biến/hàm/lớp cho rõ nghĩa
- **Extract method** – tách điều kiện logic thành các phương thức nhỏ
- **Extract class** – tạo Customer để đóng gói thông tin
- **Move method** – đẩy logic kiểm tra điều kiện vào Customer
- **Đơn giản hóa điều kiện** – tách thành nhiều biểu thức dễ đọc
- Tách tối thiểu 2 lớp **Customer**, **PromotionChecker**

3. Đánh giá

- Để hoàn thành bài thực hành, học viên cần:
 - Đưa mã nguồn lên GitHub.
 - Dán link của repository lên phần nộp bài trên hệ thống.