I . Thêm cách tính khoảng cách nữa.

Nếu thay đổi cách tính khoảng cách và sử dụng thư viện mới thì thiết kế hiện tại vi phạm nguyên lý OCP (Open Closed Principle). Vì khi đó mình sẽ phải chỉnh sửa code ở lớp cũ là DeliveryInfo.

1. **Cải tiến thiết kế thế nào?**

Sử dụng FactoryPattern : bằng cách tạo Abstract Class DistanceCalculatorFactory , và class AltDistanceApiCalculatorFactory và class DistanceApiCalculatorFactory.

Và tạo 3 class DistanceCalculator, AltDistanceApiCalculator , DistanceApiCalculator.

1. **Sử dụng mẫu thiết kế:**

* Sử dụng mẫu thiết kế là Factory Method
* Cho thay đổi yêu cầu: Thay đổi cách tính khoảng cách , sử dụng thư viện mới.
* Lý do: Để khi chúng ta sử dụng một thư viện tính khoảng cách mới thì chỉ cần tạo class mới mà không tác động vào class cũ.

**5. Thêm phương thức thanh toán mới: Thẻ nội địa ( Domestic Card )**

Nếu thêm phương thức thanh toán mới thì sẽ vi phạm nguyên lý DIP và OCP vì tác động vào code của subsystem.interbank, . Vi phạm control coupling vì phải xử lý xem hình thức thanh toán nào sẽ được thực hiện

**Cải tiến thiết kế thế nào?**

Sử dụng template method: Tạo ra Abstract Class Payment Type và 2 class Credit Card và Domestic Card. 2 Class này extend Payment Type và giao tiếp với các class còn lại thông qua Payment Type. Từ đó tránh được việc vi phạm nguyên lý OCP và DIP