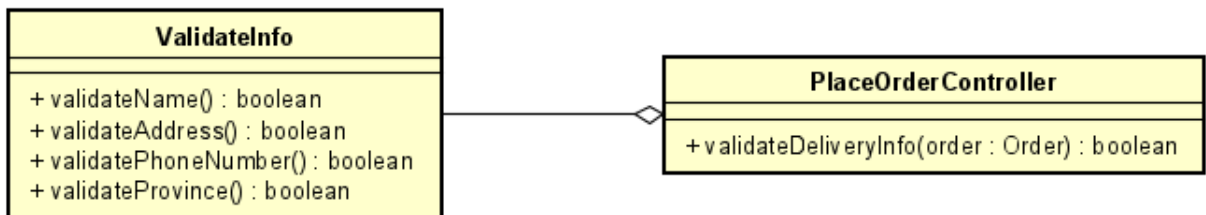


SOLID

1. Single Responsibility Principle

Related modules	Description	Improvement
PlaceRushOrder	PlaceRushOrder chỉ nên có các funtion liên quan đến việc tạo đơn hàng và thanh toán nhưng lại đảm nhiệm cả việc xác thực, tính tiền.	Các method dành cho việc validate hay tính phí ship nên ở một class khác.



2. Open/Closed Principle

Related modules	Description	Improvement
PlaceRushOrder	Phương thức tính tiền và xác thực thông tin vận chuyển khi thay đổi thì khó sửa chữa	Tạo interface và triển khai các instance là phương thức tính tiền hay xác thực nào đó

3. Liskov Substitution Principle

Related modules	Description	Improvement
PlaceRushOrder	PlaceRushOrderController kế thừa lớp cha PlaceOrderController và có thể thay thế cha trong mọi trường hợp mà không gây lỗi (override cách tính tiền và xác thực)	

4. Interface Segregation Principle

Related modules	Description	Improvement
PlaceRushOrder	Không vi phạm vì các interface đang sử dụng đã đủ chuyên biệt	

5. Dependency Inversion Principle

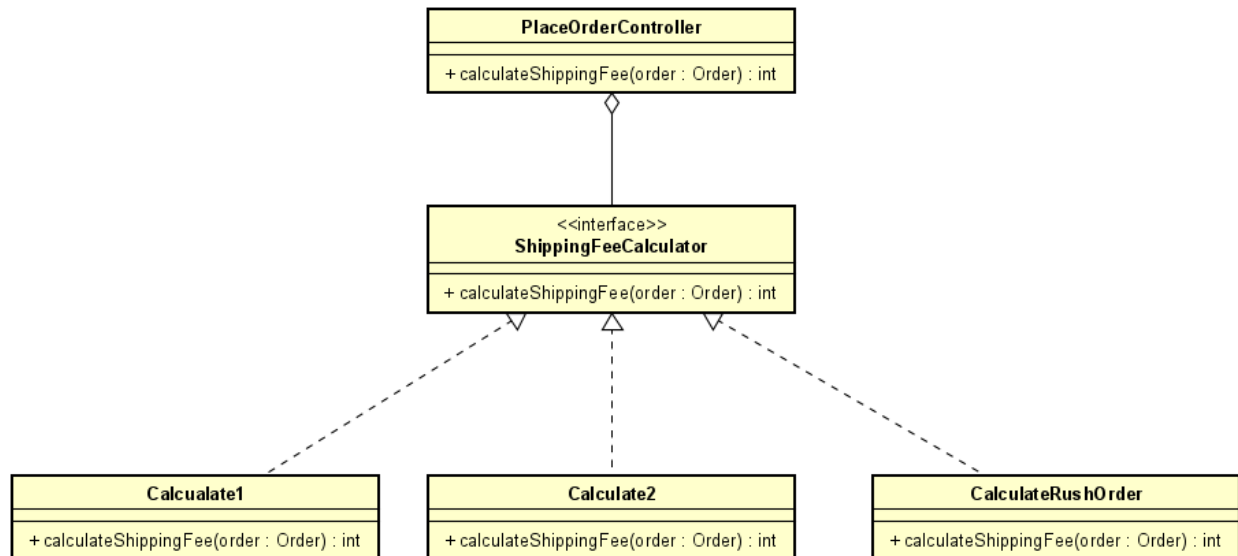
Related modules	Description	Improvement
PlaceRushOrder	Không vi phạm	

Yêu cầu bổ sung

1. Thay đổi phí ship

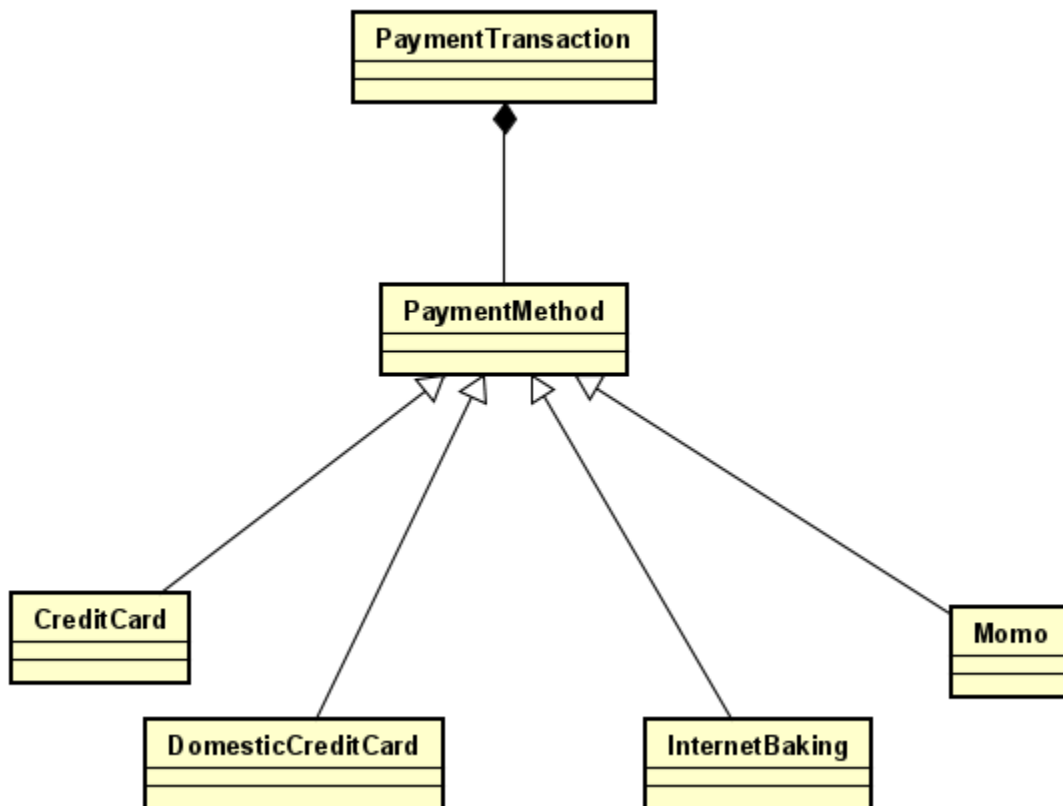
Ta sẽ tạo ra một interface `ShippingFeeCalculator` với phương thức trừu tượng là `calculateShippingFee`. Khi muốn thêm một hay nhiều cách tính phí ship thì ta chỉ cần viết thêm một lớp mới implement interface trên. Khi đó, `PlaceOrderController` khởi tạo một đối tượng là interface `ShippingFeeCalculator` với tương ứng loại chiến lược tính phí ship mà chúng ta muốn.

Class `PlaceRushOrderController` khi kế thừa cha cũng chỉ cần thay đổi instance tính tiền tương ứng với cách tính hàng giao nhanh.



2. Thêm phương thức tính tiền

Tạo một lớp abstract là PaymentMethod và lớp PaymentTransaction chỉ quan tâm đến lớp PaymentMethod này



3. Sử dụng bank khác

Vì đã sử dụng InterbankInterface để giao tiếp nên ta không cần quá quan tâm sự thay đổi bên trong interbank