

Design principle

1. Single responsibility principle

In class PlaceOrderController and PlaceRushOrderController, they are responsible for multiple jobs, so we create a new class ValidateInfo to validate information of delivery.

2. Open closed principle

The case is that there is new case rush order we need to develop, but we should not modify class order. Therefore, we create new class RushOrder extends Order, PlaceRushOrderController extends PlaceOrderController.

3. Liskov substitution principle

There is no problem in the design.

4. Interface segregation principle

There is no problem in the design.

5. Dependency inversion principle

In the future, we can use a different type of connection instead of JDBC, so we can create a ConnectionDb interface and types of connections can implement it.

We will need many kinds of payment instead of credit card such as Momo wallet, so we can create new methods in InterbankInterface that pass a parameter like paymentMethod and CreditCard or Momo wallet extends it.