

1. Single Responsibility Principle
 - PlaceOrderController: This class controls the flow of place order use case in AIMS project, but we can see that it's also responsible for validate delivery information, which looks like is the responsible of another class
2. Open Closed Principle
 - PlaceOrderController: Assume that for validating delivery information, there is a new kind of name, phone number, ... that requires the validate constraints also changes. So everytimes the new kind appear, we need to modify the validate method in PlaceOrderController class
3. Liskov Substitution Principle
 - FXMLScreenHandler: this class has attribute content of type AnchorPane, while there are many different types like BorderPane, GridPane, ...
4. Interface Segregation Principle
 - InterbankInterface: In this class, the payOrder and refund is only accepted CreditCard. If our project extends and accepts more payment method. At some time, this InterbankInterface will will have too many methods that implemented class might not need.
5. Dependency Inversion Principle
 - InterbankSubsystem & InterbankSubsystemController: The InterbankSubsystem has an attribute ctrl of type InterbankSubsystemController. The payOrder and refund operations in InterbankSubsystemController only accepts credit card. In the future, the system might accept more type of payment.