Lab 11: Solid

1. Single responsibility

Some methods do not work properly in class:

- PaymentController: emptyCart
- PlaceOrderController: createInvoice
- ScreenHandler is in charge of the controller handling the role
- ShippingScreenHandler creates Order, gets and sets information for the Order

```
// calculate shipping fees
int shippingFees = getBaseController().calculateShippingFee(order);
order.setShippingFees(shippingFees);
```

- The cart is calculated the payment when CartScreen updates it

```
void updateCartAmount() {
    // calculate subtotal and amount
    int subtotal = getBaseController().getCartSubtotal();
    int vat = (int) ((Configs.PERCENT_VAT / 100) * subtotal);
    int amount = subtotal + vat;
    LOGGER.info("amount" + amount);
```

For conclusion, it needs to refactor the methods of the appropriate class.

- 2. Open close
- PlaceOrderController has a computeShippingFee method, a defined calculation method. If there is a change in calculation. You have to change the code of the method.

```
public int calculateShippingFee(Order order) {
   Random rand = new Random();
   int fees = (int) (((rand.nextFloat() * 10) / 100) * order.getAmount());
   LOGGER.info("Order Amount: " + order.getAmount() + " -- Shipping Fees: " + fees);
   return fees;
}
```

- The PaymentController is passed a payOrder method, processed according to credit card parameters and card payment. However, if you want to add another payment method, you need to change payOrder

So, there is no solution yet

- 3. Interface segregation (no solution)
- 4. Liskov substitutes (no solution)
- 5. Dependency inversion
- PaymentController contains two properties of type CreditCard and InterbankInterface for credit card payments. As a result, PaymentController relies on cards and interbanks.

```
private CreditCard card;

/**
    * Represent the Interbank subsystem.
    */
private InterbankInterface interbank;
```

So, Move these 2 properties to a higher level of abstraction, let CreditCard inherits PaymentEntity and InterbankInterface inherits PaymentInterface