

# Assignment #3

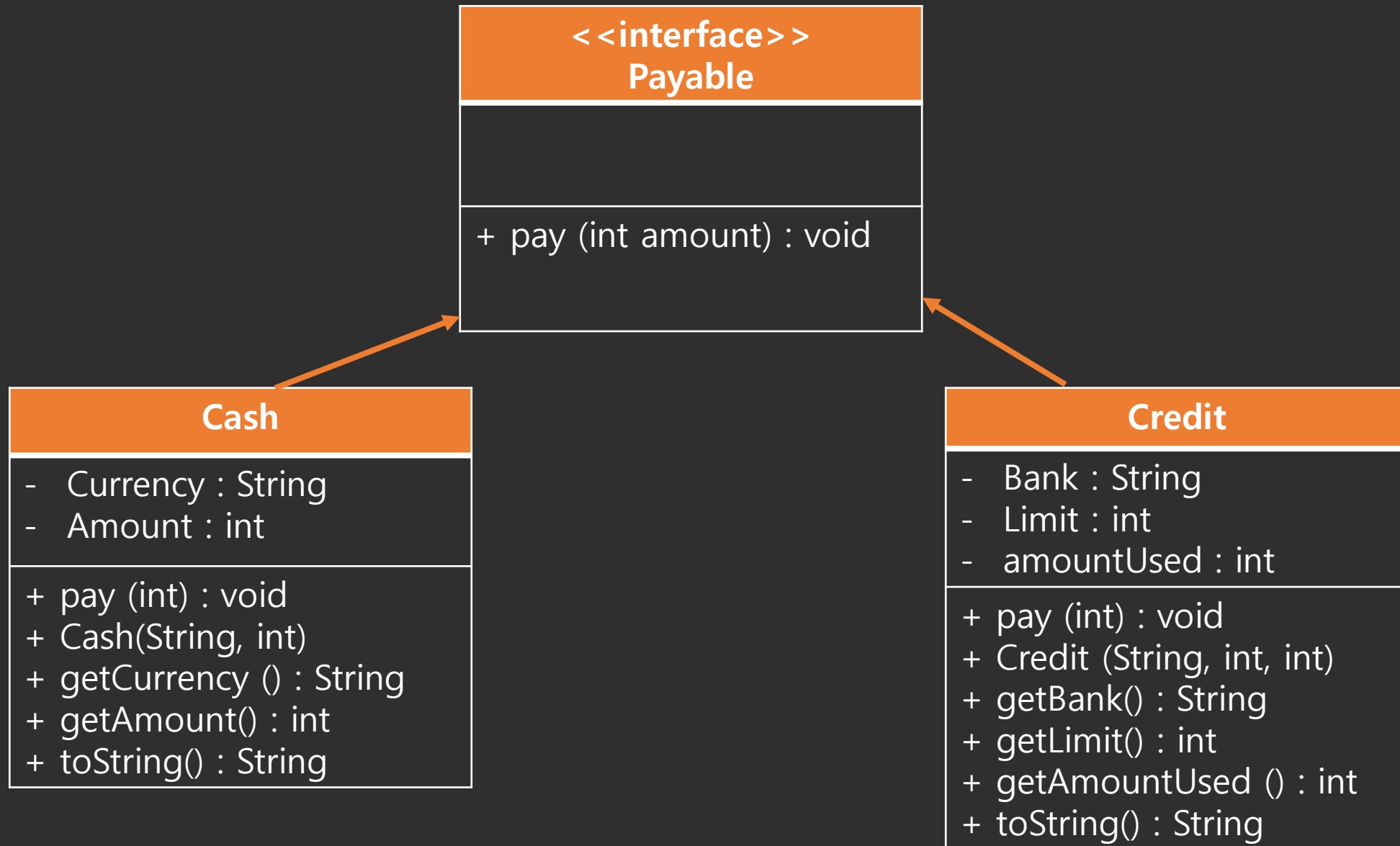
2019006171 김은민

운영체제  
Windows 10

이클립스 버전  
4.18.0.v20201202-1800  
Build id: I20201202-1800

자바 버전  
java 15.0.2 2021-01-19

# Classes & Methods



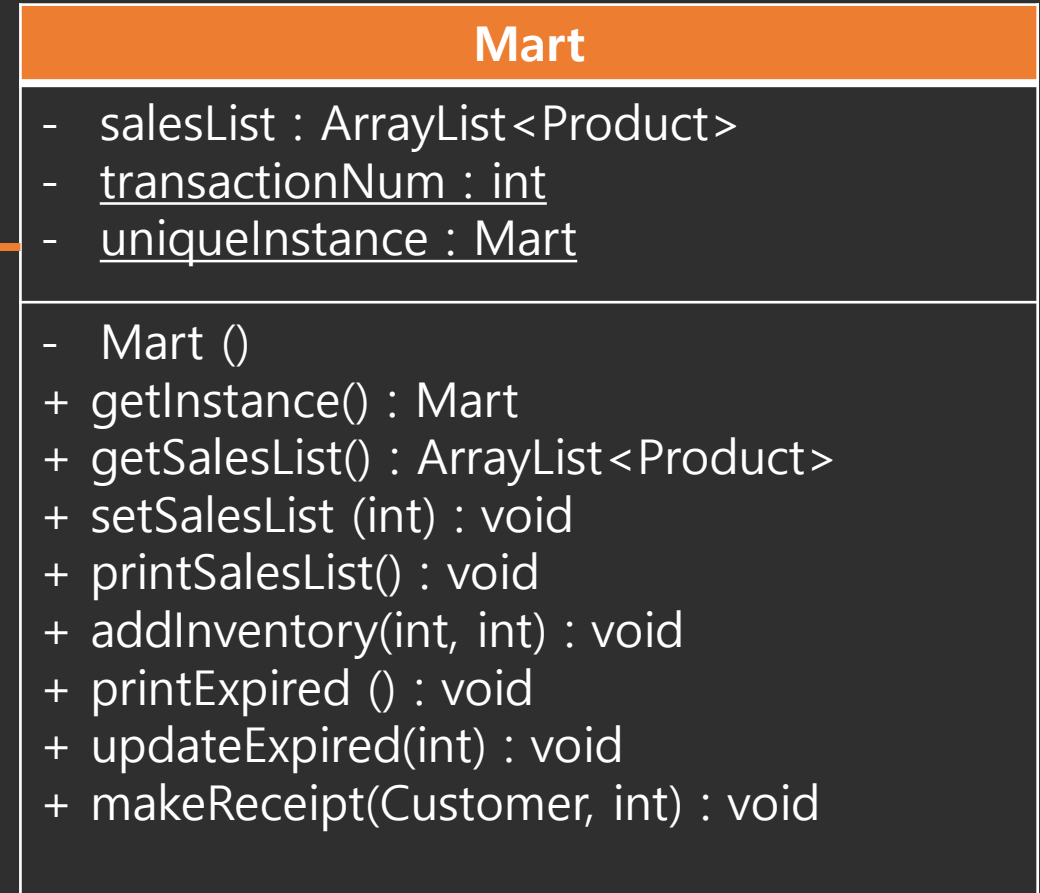
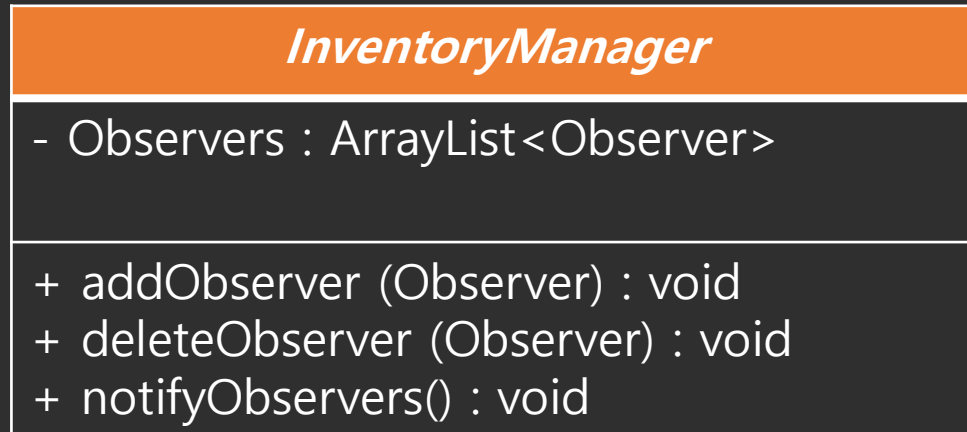
# Classes & Methods

Product
<ul style="list-style-type: none"><li>- Name : String</li><li>- Price : int</li><li>- Quantity : int</li><li>- waitList : ArrayList&lt;Customer&gt;</li></ul>
<ul style="list-style-type: none"><li>+ Product (String, int, int)</li><li>+ getQuantity () : int</li><li>+ getPrice() : int</li><li>+ getName () : String</li><li>+ addQuantity(int) : void</li><li>+ setQuantity(int) : void</li><li>+ wait(Customer) : void</li><li>+ getWaitList () : ArrayList&lt;Customer&gt;</li></ul>

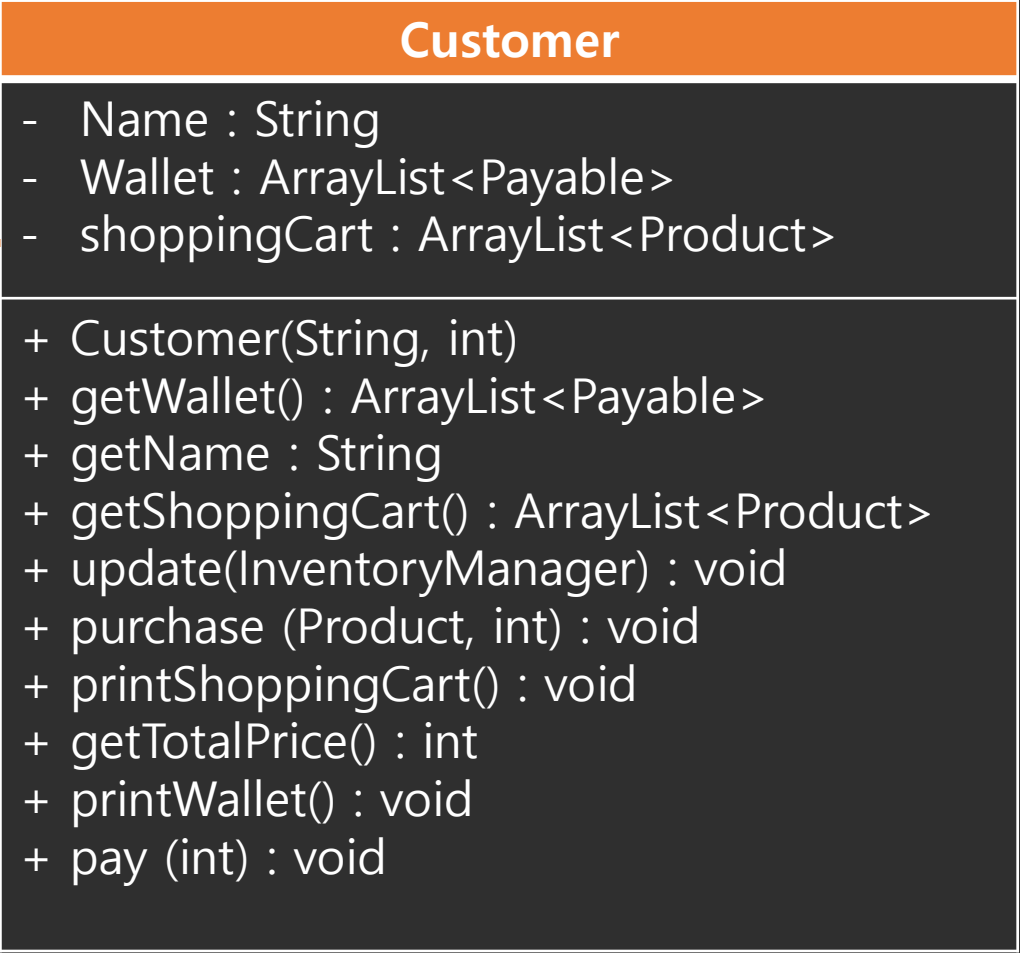
Food
<ul style="list-style-type: none"><li>- expirationDateTime : LocalDateTime</li></ul>
<ul style="list-style-type: none"><li>+ isExpired(LocalDateTime) : Boolean</li><li>+ Food(String, int, int, int, int, int, int, int)</li><li>+ getExpirationDateTime() : LocalDateTime</li><li>+ setExpirationDateTime(LocalDateTime) : void</li></ul>

Manufactured
<ul style="list-style-type: none"><li>- Brand : String</li></ul>
<ul style="list-style-type: none"><li>+ Manufactured (String, int, String, int)</li><li>+ getBrand() : String</li></ul>

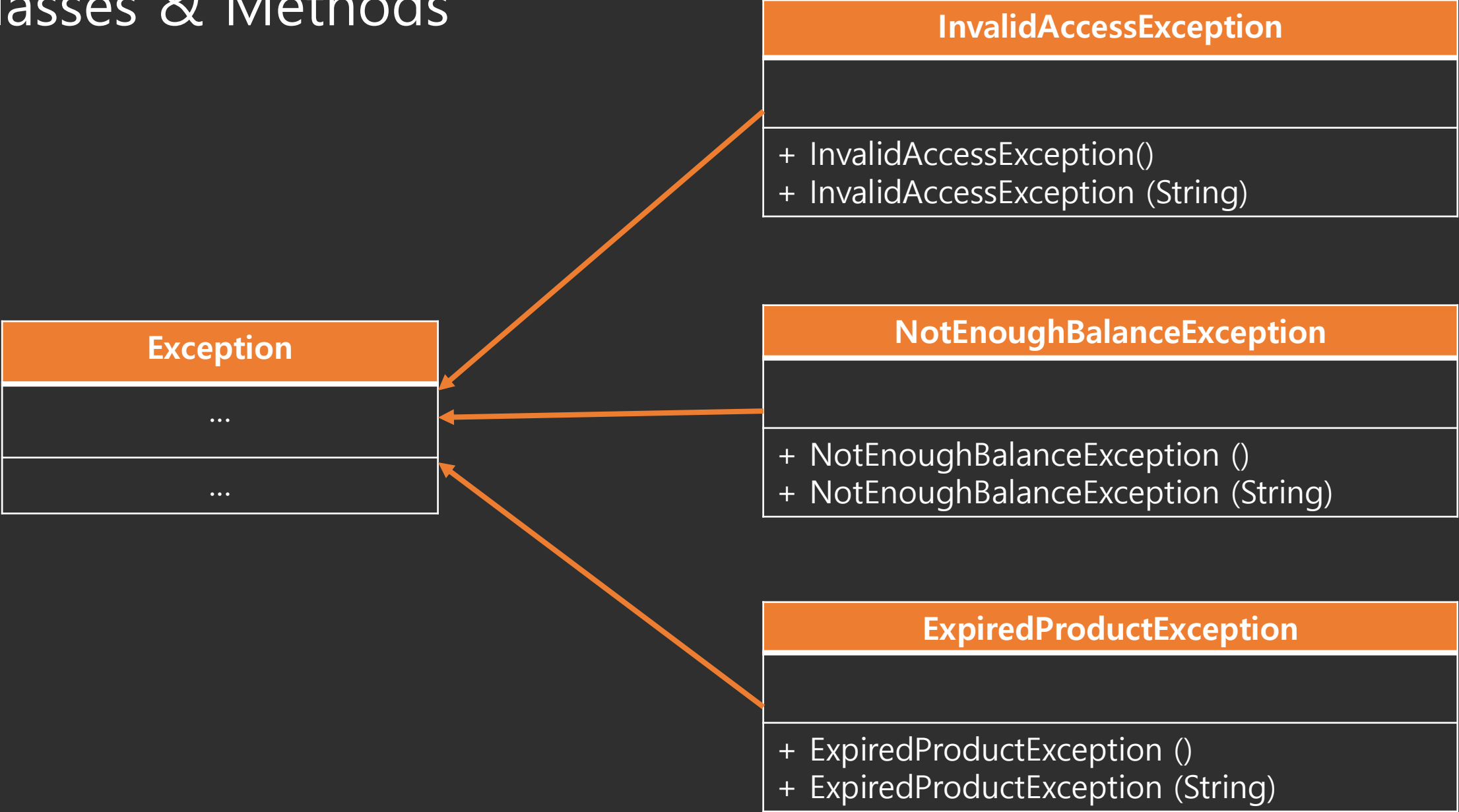
# Classes & Methods



# Classes & Methods



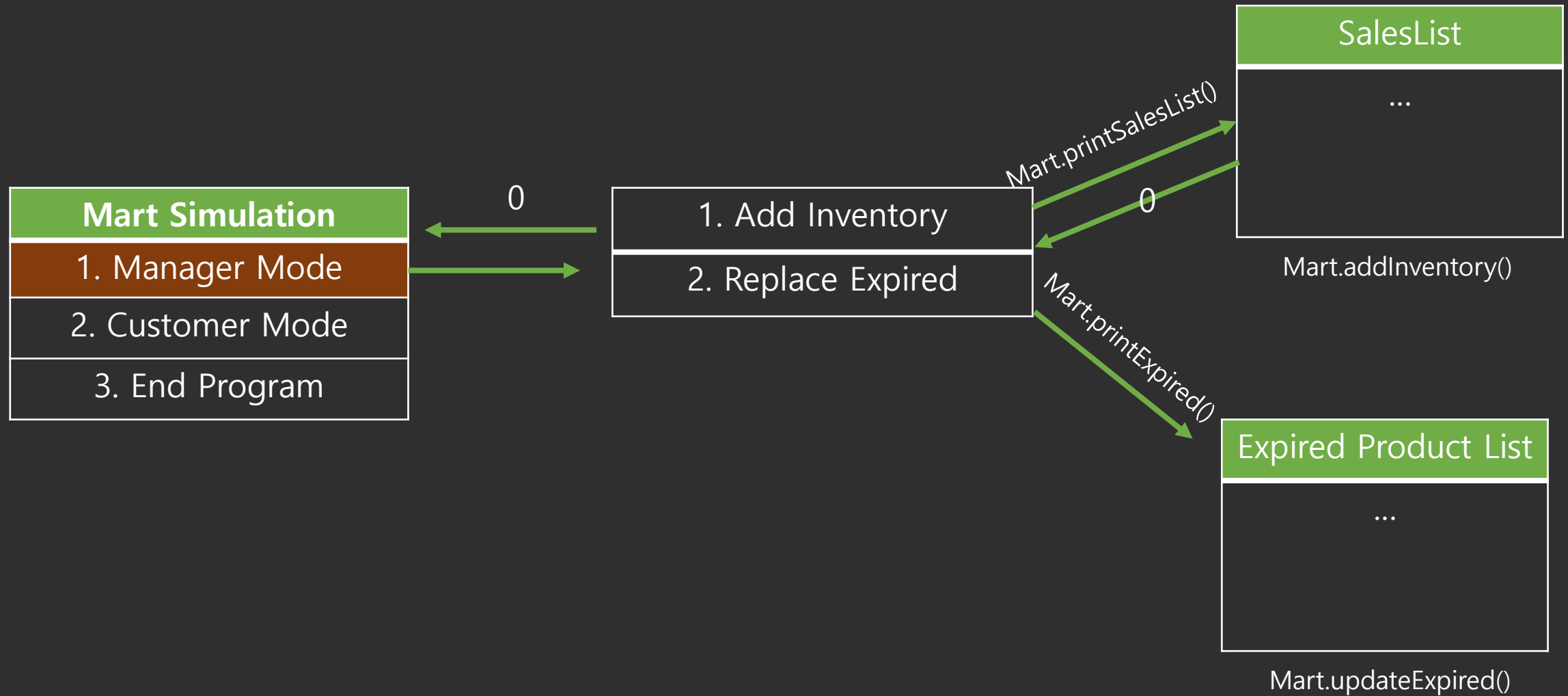
# Classes & Methods



# Execution



# Execution

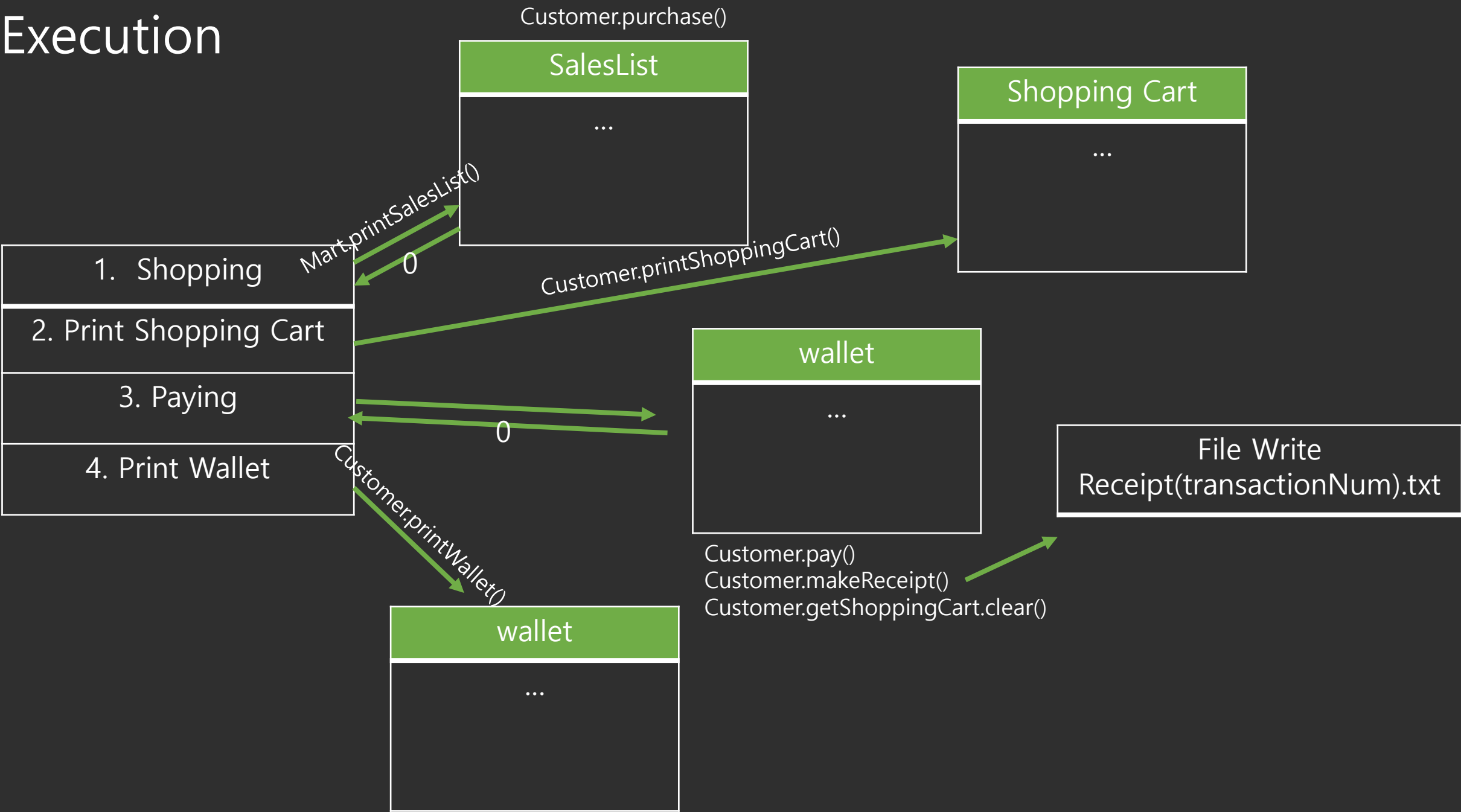




# Execution



# Execution



# Design Pattern

## Singleton Pattern

Mart.java

```
public class Mart extends InventoryManager {  
    private ArrayList<Product> salesList;  
    public static int transactionNum = 1000;  
  
    private static Mart uniqueInstance;  
    private Mart() {};  
    public synchronized static Mart getInstance() {  
        if (uniqueInstance == null)  
            uniqueInstance = new Mart();  
        return uniqueInstance;  
    }  
}
```

- Private Constructor of class Mart
- Unique instance of class Mart
- getInstance() method returns the unique instance of Mart

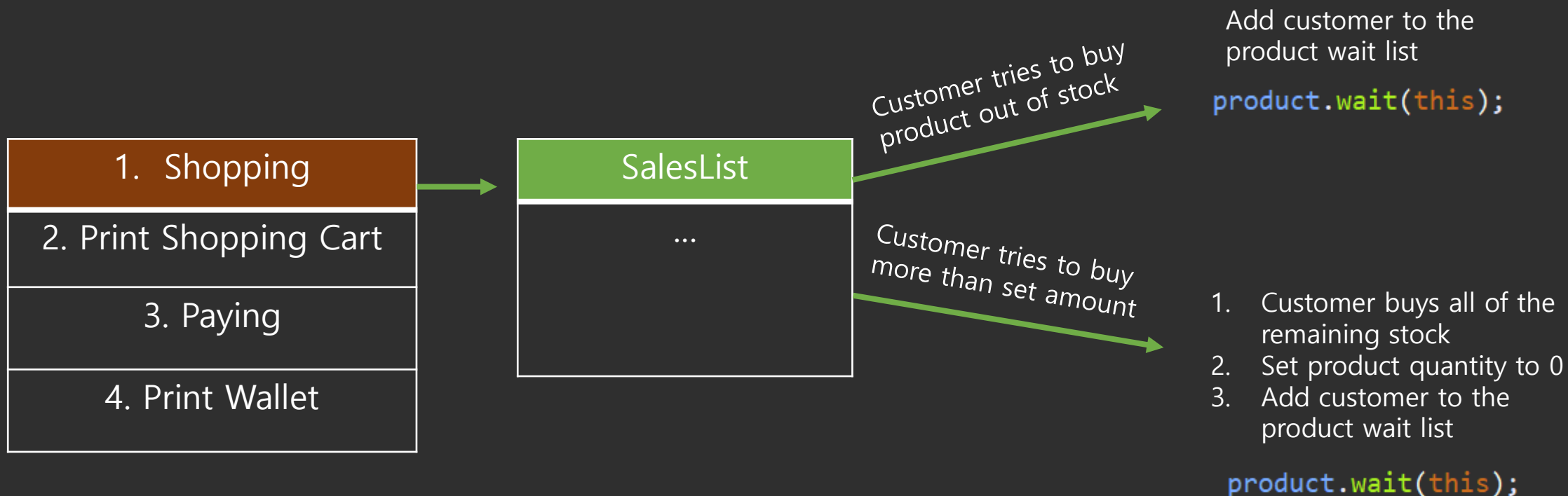
Shopping.java

```
Mart mart = Mart.getInstance();
```

- Unique instance of Mart is referenced in shopping.java

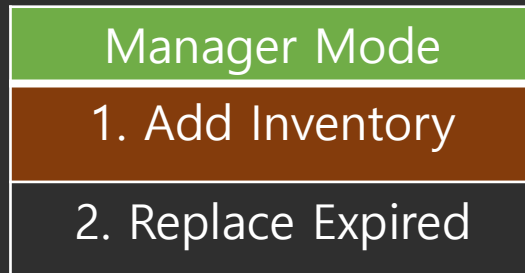
# Design Pattern

## Observer Pattern



# Design Pattern

## Observer Pattern



Stock refilled

`mart.notifyObservers();`

Update all of the  
observers(customers)

```
for (int i=0;i<observers.size();i++) {  
    observers.get(i).update(this);  
}
```

If there were 2 customers on the wait list, and first customer bought all of the remaining stock, The second customer remains on the wait list

For customers on the waiting list of restocked item, let the item be purchased by those customers

And remove customers from the waiting list

Customer tries to buy more than the amount of refilled stock  
- Remains on the wait list