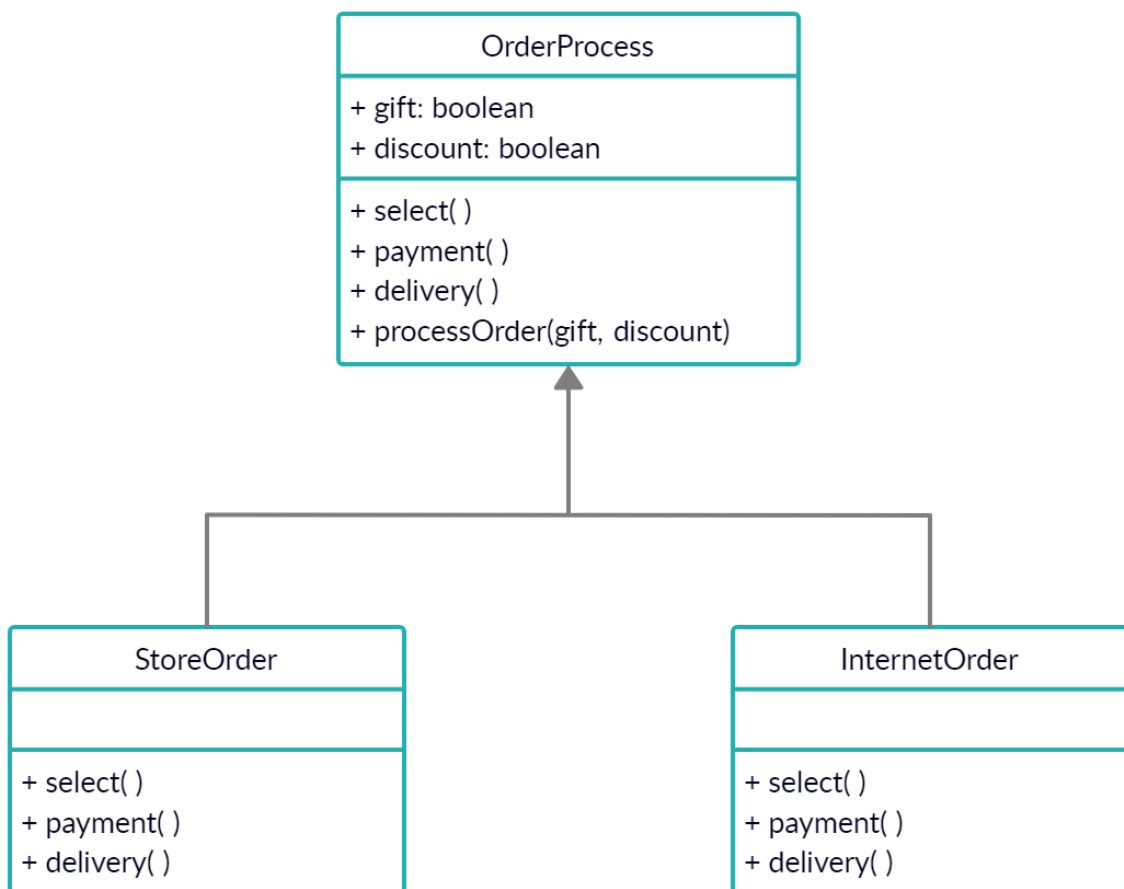


## Template Method Pattern

Template Method Pattern คือ ต้นแบบที่สร้างขึ้นเพื่อลดโค้ดที่มียการทำงานซ้ำซ้อนกันในแต่ละคลาส ในกรณีที่แต่ละคลาสมี Method ที่ทำงานเหมือนกัน ให้สร้างคลาสขึ้นมาเก็บ Method ที่ทำงานเหมือนกันไว้เป็นเหมือน Template แล้วให้แต่ละคลาสดึงไปใช้โดยการ extends แทนการเขียน Method ซ้ำกันในทุกๆ คลาส

กรณีศึกษา “กระบวนการการสั่งซื้อสินค้า” เป็นการแสดงขั้นตอนการซื้อสินค้าว่ามีอะไรบ้าง และอธิบายแต่ขั้นตอน เช่น การเลือกสินค้า การชำระสินค้า การขนส่ง สิทธิพิเศษต่างๆ (เช่น การห่อของขวัญ ส่วนลด) และได้มีการกำหนดวิธีสั่งซื้อสินค้าเป็น 2 แบบ คือ แบบหน้าร้าน กับแบบออนไลน์



รูปที่ 1. Class Diagram ของกระบวนการการสั่งซื้อสินค้า

จาก Class Diagram ข้างต้นแสดงให้เห็นว่า คลาส OrderProcess เป็นคลาสที่สร้างไว้เก็บ Template method จึงกำหนดให้ OrderProcess เป็น Abstract class ส่วนคลาส StoreOrder กับ InternetOrder เป็นวิธีการสั่งซื้อซึ่งจะ extends มาจาก Abstract class โดยมีขั้นตอนเหมือนกันแต่วิธีในแต่ละขั้นตอนต่างกัน จึงทำการ Override methodมาใช้ และกำหนดการทำงานใน method

โค้ดโปรแกรม Oder Process มีรายละเอียดดังรูป ตามลำดับ

```
OrderProcess.java
1  package v1;
2  public abstract class OrderProcess
3  {
4      public boolean gift;
5      public boolean discount;
6      public abstract void select();
7      public abstract void payment();
8      public abstract void delivery();
9
10     public final void processOrder(boolean gift,boolean discount)
11     {
12         select();
13         payment();
14         if (gift)
15         {
16             System.out.println("Gift wrap successful");
17         }
18         if (discount)
19         {
20             System.out.println("Use discount successful");
21         }
22         delivery();
23     }
24 }
```

รูปที่ 2. คลาส OrderProcess

```
StoreOrder.java
1  package v1;
2  public class StoreOrder extends OrderProcess
3  {
4      @Override
5      public void select()
6      {
7          System.out.println("Chooses products from shelf in the store.");
8      }
9
10     @Override
11     public void payment()
12     {
13         System.out.println("Pays at cashier counter by cash or credit card.");
14     }
15
16     @Override
17     public void delivery()
18     {
19         System.out.println("Customer pick up products");
20     }
21 }
```

รูปที่ 3. คลาส StoreOrder

```

InternetOrder.java
1 package v1;
2 public class InternetOrder extends OrderProcess
3 {
4     @Override
5     public void select()
6     {
7         System.out.println("Add products to online shopping cart.");
8         System.out.println("Get gift wrap preference");
9         System.out.println("Get delivery address.");
10        System.out.println("Can use discount if you have.");
11    }
12
13    @Override
14    public void payment()
15    {
16        System.out.println("Online Payment through Internetbanking, Paypal or credit card.");
17    }
18
19    @Override
20    public void delivery()
21    {
22        System.out.println("Ship products to delivery address");
23    }
24 }

```

รูปที่ 4. คลาส InternetOrder

```

Main.java
1 package v1;
2 class Main
3 {
4     public static void main(String[] args)
5     {
6         OrderProcess storeOrder = new StoreOrder();
7         storeOrder.processOrder(true,false);
8         System.out.println();
9         OrderProcess netOrder = new InternetOrder();
10        netOrder.processOrder(true,true);
11    }
12 }

```

รูปที่ 5. โปรแกรมหลัก

```

Chooses products from shelf in the store.
Pays at cashier counter by cash or credit card.
Gift wrap successful
Customer pick up products

Add products to online shopping cart.
Get gift wrap preference
Get delivery address.
Can use discount if you have.
Online Payment through Internetbanking, Paypal or credit card.
Gift wrap successful
Use discount successful
Ship products to delivery address

```

รูปที่ 6. ผลลัพธ์ของโปรแกรม

## เอกสารอ้างอิง

1. Template Method, <https://refactoring.guru/design-patterns/template-method>
2. Design Patterns RECAP, <https://medium.com/rootusercc/design-patterns-recap-1dc10302d399>
3. Template Method Design Pattern in Java, <https://www.journaldev.com/1763/template-method-design-pattern-in-java>
4. Template Method Design Pattern, <https://www.geeksforgeeks.org/template-method-design-pattern/>
5. Design Patterns แบบพื้นฐานสุด ๆ มาดูมันคืออะไร,  
<https://www.borntodev.com/2020/04/05/design-patterns-แบบพื้นฐานสุด-ๆ/#Template>