

## แบบฝึกหัดที่ 2

1. ให้ปรับปรุง CustomerMailApplication ให้สมบูรณ์ โดยประยุกต์ใช้ Factory Design Pattern โดยให้สร้าง

คลาสชื่อ CustomerFactory ซึ่งมีโครงสร้างดังนี้

```
public class CustomerFactory {  
    public static Customer createCustomer(String type) {  
        ...  
    }  
}
```

โดยให้ Fork Github repository ของผม <https://github.com/ajsarun/custmailwofactory> ไปที่ Github repository ของนักศึกษา จากนั้น Clone repository มายังเครื่องของนักศึกษา แก้โปรแกรมจนสมบูรณ์แล้วโหลดขึ้น Github repository ของนักศึกษา และส่งลิงก์ของ Github repository ของนักศึกษาใน Google Classroom

2. สมมติมีคลาส ScheduleServer และ main Program repository ที่จะให้ต่อไป ประเด็นคือทุกครั้งที่จะให้ Server เริ่มทำงานจะต้องทำงานตามขั้นตอนต่าง ๆ ตามที่แสดงใน main program เช่นเดียวกับการหยุดการทำงานของ Server ก็ต้องทำตามขั้นตอนต่าง ๆ ใน Main Program เช่นกัน นักศึกษาจะใช้ Façade Design Pattern ช่วยแก้ปัญหานี้ได้อย่างไร เพื่อให้การเริ่ม Server ทำได้ด้วยการเรียกคำสั่งเดียว และการหยุด Server ทำได้ด้วยการเรียกคำสั่งเดียว

โดยให้ Fork Github repository ของผม <https://github.com/ajsarun/serverwofacade> ไปที่ Github repository ของนักศึกษา จากนั้น Clone repository มายังเครื่องของนักศึกษา แก้โปรแกรมจนสมบูรณ์แล้วโหลดขึ้น Github repository ของนักศึกษา และส่งลิงก์ของ Github repository ของนักศึกษาใน Google Classroom

3. เขียนโปรแกรมดังนี้ สมมติมีออบเจกต์ซึ่งรายงานผลสกอร์ฟุตบอล โดยผู้ใช้จะป้อนผลฟุตบอล เช่น Thai 1-0 UAE เมื่อป้อนเสร็จแล้ว ก็จะส่งผลไปยังออบเจกต์ที่ลงทะเบียนไว้และแสดงผลลัพธ์ให้ประยุกต์ใช้ Observer Design Patterns โดยใน main program ให้มีผู้ลงทะเบียนรับผลอย่างน้อยสองคน และให้มีส่วนที่ผู้ป้อนข้อมูลผลวนลูปป้อนข้อมูลไปเรื่อย ๆ โดยสามารถจบโปรแกรมด้วยการกดปุ่ม Enter โดยไม่ป้อนข้อมูลใด ๆ ออฟโปรแกรมขึ้น Github แล้วส่งลิงก์ไปยัง Github เข้า Google Classroom

Enter Score Thai 1-0 UAE <- คนป้อน

live result: Thai 1-0 UAE <- แสดงผล

live result: Thai 1-0 UAE <- แสดงผล

Enter Score Thai 2-0 UAE <- คนป้อน

live result: Thai 2-0 UAE <- แสดงผล

live result: Thai 2-0 UAE <- แสดงผล

...