

ให้นิสิตสรุปเนื้อหาคาบบรรยายวันที่ 22 ธ.ค. 63 เรื่อง Introduction to Object and Class II

คาบปฏิบัติการ

ให้นิสิตอ่าน และปฏิบัติตาม แล้วตอบคำถาม ส่งภายในคาบเรียน

ร้านสะดวกซื้อร้านหนึ่ง ซึ่งมีหลายสาขาทั่วประเทศ มีระบบบัตรสมาชิกสำหรับการซื้อสินค้าภายในร้าน เพื่อให้สมาชิกได้รับสิทธิพิเศษบางประการ ในคาบปฏิบัติการนี้ จะมาเรียนรู้การเขียนโปรแกรมเชิงวัตถุ เพื่อสร้างระบบบัตรสมาชิกของร้านสะดวกซื้อร้านนี้

บัตรสมาชิก จะเก็บข้อมูลดังนี้

- ชื่อสมาชิก
- หมายเลขโทรศัพท์ของสมาชิก
- จำนวนยอดซื้อสินค้าสะสม (บาท)
- จำนวน E-Stamp

1. Model Class คือคลาสที่จัดการกับการจัดเก็บและการเปลี่ยนแปลง state ของข้อมูล ให้นิสิตสร้าง Model Class ชื่อ MemberCard เพื่อเก็บข้อมูลของบัตรสมาชิกข้างต้น ให้ถูกต้องตามหลักการตั้งชื่อ และกำหนด access specifier ของทุก field เป็น private

```
// ส่งโค้ดของคลาส MemberCard ในช่องนี้
```

2. เพิ่ม method ในคลาส MemberCard เพื่อจัดการเรียกดูข้อมูลใน instance โดยเพิ่ม getter ของข้อมูลทั้ง 4 ค่า โดย access specifier ของ getter เป็น public

```
// ส่งโค้ดของคลาส MemberCard ในช่องนี้
```

3. เพิ่ม method ในคลาส MemberCard เพื่อจัดการการเปลี่ยนแปลง state ของข้อมูลใน instance โดยให้มี method ต่อไปนี้

3.1 public void addPurchase (double purchase)

เป็น method สำหรับเพิ่มจำนวนยอดซื้อสินค้าสะสม

และทุกยอดซื้อ purchase 50 บาท จะได้รับ E-Stamp เพิ่มอีก 1 ดวง

// สก๊อตของคลาส MemberCard ในช่องนี้

3.2 public boolean useStamp(int stamp)

เป็น method สำหรับใช้ E-Stamp ซึ่งต้องระบุจำนวน E-Stamp ที่จะใช้ และไปลดจำนวน E-Stamp
ก็ต่อเมื่อจำนวน E-Stamp เพียงพอ และคืนค่า true
แต่หากจำนวน E-Stamp ไม่เพียงพอที่จะใช้ ให้คืนค่า false โดยไม่ลดจำนวน E-Stamp

```
// ส่งโค้ดของคลาส MemberCard ในช่องนี้
```

4. เพิ่ม Constructor ในคลาส MemberCard เพื่อกำหนดค่าเริ่มต้นให้ instance ของคลาส MemberCard โดย Constructor จะรับข้อมูลชื่อสมาชิก และหมายเลขโทรศัพท์ (ไม่ต้องรับ จำนวนยอดซื้อสะสม และ จำนวน E-Stamp)

```
// สก๊อตของคลาส MemberCard ในช่องนี้
```

5. เพิ่มอีก Constructor ในคลาส MemberCard เพื่อกำหนดค่าเริ่มต้นให้ instance ของคลาส MemberCard

โดย Constructor รูปแบบที่ 2 จะรับข้อมูลชื่อสมาชิก หมายเลขโทรศัพท์ และจำนวน E-Stamp เริ่มต้น (เนื่องจากลูกค้าบางคน ได้รับ Stamp ไปก่อนหน้าที่จะทำบัตรสมาชิก) (ไม่ต้องรับยอดซื้อสะสม)

```
// สก๊ิดของคลาส MemberCard ในช่องนี้
```

6. ปรับปรุง Constructor ทั้ง 2 รูปแบบ ให้ใช้งาน Constructor Chain

// ส่งโค้ดของคลาส MemberCard ในช่องนี้

7. Constructor Chain คืออะไร มีประโยชน์อย่างไร

8. เขียน Main Class ให้มี main method เพื่อทดสอบ instance ของคลาส MemberCard ดังนี้

8.1 ลูกค้า John เป็นลูกค้าใหม่ ต้องการทำบัตรสมาชิกใหม่ ลงทะเบียนด้วยหมายเลข a

วันที่ 1 สมาชิก John ซื้อสินค้าเป็นจำนวนเงินรวม 360 บาท

วันที่ 2 สมาชิก John ซื้อสินค้าเป็นจำนวนเงินรวม 486.50 บาท

ผ่านไป 2 วัน สมาชิก John มียอดซื้อสะสมเท่าใด และมีจำนวน E-Stamp อยู่เท่าใด

8.2 ลูกค้า Smith เป็นลูกค้าเก่า มี Stamp อยู่ 35 ดวง ต้องการทำบัตรสมาชิกใหม่

Smith ลงทะเบียนด้วยหมายเลข 0678901234

วันที่ 1 สมาชิก Smith ซื้อสินค้าเป็นจำนวนเงินรวม 35 บาท

วันที่ 2 สมาชิก Smith ซื้อสินค้าเป็นจำนวนเงินรวม 186 บาท

วันที่ 3 สมาชิก Smith ซื้อสินค้าเป็นจำนวนเงินรวม 311 บาท

ผ่านไป 3 วัน สมาชิก Smith มียอดซื้อสะสมเท่าใด และมีจำนวน E-Stamp อยู่เท่าใด

// ส่องโค้ดของ Main Class ในช่องนี้

ผลลัพธ์จากการประมวลผล Main Class คือ

ร้านสะดวกซื้อร้านนี้ มีเครื่องคิดเงิน ซึ่งจะคำนวณยอดซื้อสะสม และคำนวณส่วนลดจากการใช้ E-Stamp ของสมาชิก

เครื่องคิดเงิน จะเก็บข้อมูลดังนี้

- จำนวนสินค้าทั้งหมดที่ซื้อ
- จำนวนราคาสินค้ารวม
- บัตรสมาชิกที่ลูกค้าใช้ในการซื้อสินค้าครั้งนี้
- ยอดชำระสินค้าที่ลูกค้าต้องจ่ายจริง
- เงินที่ลูกค้าจ่าย
- เงินทอน

9. ให้นักศึกษาร่าง Model Class ชื่อ CashRegister เพื่อเก็บข้อมูลของเครื่องคิดเงินข้างต้น ให้ถูกต้องตามหลักการตั้งชื่อ และกำหนด access specifier ของทุก field เป็น private

```
// สก๊อตของคลาส CashRegister ในช่องนี้
```

10. คลาส CashRegister จะใช้ Default Constructor เพื่อกำหนดค่าเริ่มต้นทุก field เป็น 0 ยกเว้น field บัตรสมาชิกที่ลูกค้าใช้ในการซื้อสินค้าครั้งนี้ (MemberCard)
Default Constructor จะกำหนดให้ field ที่เป็น instance ของคลาสอื่น มีค่าเริ่มต้นเป็นค่าอะไร

11. กำหนด setter ของข้อมูลบัตรสมาชิกที่ลูกค้าใช้ในการซื้อสินค้าครั้งนี้ในคลาส CashRegister (generate ในโปรแกรม IntelliJ IDEA)
เพื่อให้พนักงานแคชเชียร์ใส่ข้อมูลบัตรสมาชิกที่ลูกค้าใช้

```
// ส่งโค้ดของคลาส CashRegister ในช่องนี้
```

12. คลาส CashRegister มี method ที่จัดการการเปลี่ยนแปลง state ของข้อมูลดังนี้

12.1 public void purchaseItem (double purchase)

เป็น method สำหรับระบุงบราคาสินค้าแต่ละชิ้นที่ซื้อ ซึ่งจะไปเพิ่มจำนวนสินค้าที่ซื้อ และเพิ่มราคาสินค้ารวม

```
// ส่งโค้ดของคลาส CashRegister ในช่องนี้
```

12.2 public void pay (double cash, int stamp)

เป็น method สำหรับรับเงินสดจากลูกค้า และจำนวน stamp ที่ลูกค้าต้องการใช้

method นี้ทำงาน 2 ขั้นตอน คือ 1. คำนวณยอดชำระสินค้าที่ลูกค้าต้องจ่ายจริง (ถ้ามีส่วนลด) และ 2. คำนวณเงินทอน

กรณีที่มีการระบุบัตรสมาชิก ให้นำ E-Stamp มาใช้เป็นส่วนลดในการซื้อสินค้า โดย E-Stamp 1 ดวง มีมูลค่า 2 บาท (หากจำนวน E-Stamp ไม่เพียงพอ จะต้องไม่มีการคิดส่วนลด) ก่อน การเพิ่มยอดซื้อ สะสมในบัตรสมาชิก

กรณีที่ไม่มีบัตรสมาชิก ลูกค้าอาจจะใช้ stamp มาใช้เป็นส่วนลดได้เช่นกัน โดย stamp 1 ดวง มีมูลค่า 1 บาท

```
// ส่งโค้ดของคลาส CashRegister ในช่องนี้
```

12.3 public void clear ()

เป็น method สำหรับการเคลียร์ค่าทั้งหมดในเครื่องคิดเงิน เพื่อรอรับการซื้อสินค้าจากลูกค้าคนต่อไป

```
// สังกัดของคลาส CashRegister ในช่องนี้
```

13. เขียน Main Class ต่อจากข้อที่ 8 เพื่อทดสอบ instance ของคลาส CashRegister ดังนี้

13.1 ลูกค้า John ซื้อสินค้า 5 ชิ้น ราคา

15, 16.50, 24, 18 และ 30 บาท ตามลำดับ

ใส่ข้อมูลบัตรสมาชิกของ John ไปในเครื่องคิดเงิน

แสดงจำนวนสินค้าทั้งหมดที่ John ซื้อ และราคาสินค้ารวม

ลูกค้า John จ่ายเงิน 500 บาท และต้องการใช้ E-Stamp 4 ดวง เพื่อเป็นส่วนลด

แสดงราคาสินค้ารวม, ราคาที่ John ต้องจ่ายจริง และเงินทอน

สั่งเคลียร์เครื่องคิดเงิน

13.2 ลูกค้า Belle ไม่มีบัตรสมาชิก ซื้อสินค้า 3 ชิ้น ราคา 34, 65, 20 บาท ตามลำดับ

เนื่องจากลูกค้า Belle ไม่มีบัตรสมาชิก จะต้องไม่มีการเรียก method ใส่ข้อมูลบัตรสมาชิก

แสดงจำนวนสินค้าทั้งหมดที่ Belle ซื้อ และราคาสินค้ารวม

ลูกค้า Belle จ่ายเงิน 100 บาท และต้องการใช้ stamp 30 ดวง เพื่อเป็นส่วนลด

แสดงราคาสินค้ารวม, ราคาที่ Belle ต้องจ่ายจริง และเงินทอน

สั่งเคลียร์เครื่องคิดเงิน

// ส่งโค้ดของ Main Class ในช่องนี้

ผลลัพธ์จากการประมวลผล Main Class คือ