

ชุดที่ 2

1. จาก Class diagram ที่กำหนดให้ จงเขียนโปรแกรมเพื่อสร้าง MemberCard ซึ่งประกอบด้วย

```
class MemberCard
+static final int POINTS2CASH100 = 800;
+static final int BAHT2POINTS = 25;
-long memberId;
-double totalPurchaseAmount;
-int totalPoints;
-int numberOfRedeem;

+MemberCard(long idCard)
+buy(double purchaseAmount):void
-addPoints(double purchaseAmount):void
+redeem():int
+getRemainingPoints():int
+getUsagePoints():int
+getTotalPoints():int
+getTotalPurchaseAmount():double
+getMemberId():long
+toString():String
```

• Attributes

- ค่าคงที่ POINTS2CASH100=800 (800 คะแนน สำหรับแลกเงินสด 100 บาท)
- ค่าคงที่ BAHT2POINTS=25 (25 บาท สำหรับแลก 1 คะแนน)
- memberId สำหรับเก็บรหัสสมาชิก
- totalPurchaseAmount สำหรับเก็บยอดซื้อสะสมตามใบเสร็จรับเงิน
- totalPoints สำหรับเก็บคะแนนสะสมทั้งหมด
- numberOfRedeem สำหรับเก็บจำนวนครั้งที่มีการแลกคะแนน

• Methods

- constructor รับเลขบัตรประชาชนของสมาชิกกำหนดเป็นค่าเริ่มต้นให้กับ memberId attribute
- addPoints() คำนวณและบันทึกคะแนนสะสมที่ได้จากใบเสร็จรับเงินที่ซื้อสินค้า
- buy() ทำการสะสมยอดซื้อสินค้าตามใบเสร็จรับเงิน และทำการเรียก addPoints() เพื่อสะสมยอดคะแนนโดย 25 บาทเท่ากับ 1 คะแนน
- redeem() นำคะแนนแลกเงินสด โดยถ้าคะแนนมีมากกว่าหรือเท่ากับ 800 คะแนน จะทำการแลก 800 คะแนน โดยปรับปรุงจำนวนครั้งของการแลกคะแนน (numberOfRedeem) ขึ้น 1 และคืนค่าเป็นจำนวนเงินสด 100 บาท ถ้ามีคะแนนน้อยกว่าจะคืนค่าเป็น 0
- getRemainingPoint() คำนวณและคืนค่าคะแนนคงเหลือ
- getUsagePoint() คืนค่าเป็นคะแนนทั้งหมดที่มีทำการแลก
- toString() แสดงข้อมูลรหัสสมาชิก ยอดซื้อสะสม คะแนนสะสมทั้งหมด จำนวนครั้งที่มีการแลกคะแนน คะแนนคงเหลือ

ตัวอย่างการเรียกใช้คำสั่ง

```
MemberCard mc1 = new MemberCard (310001234671L);
mc1.buy(100000); //ซื้อสินค้า 100000 บาท สะสมคะแนนได้ 4000 คะแนน
mc1.redeem();//แลกคะแนน 800 คะแนน ได้เงินสด 100 บาท เหลือคะแนน 3200 คะแนน
mc1.getUsagePoint(); //คืนค่า 800 คะแนน
```

Object State

MemberCard {memberId=310001234671, totalPurchaseAmount=100000.0, totalPoints=4000, numberOfRedeem=1, remainingPoints=3200}