

PRINCE OF SONGKLA UNIVERSITY  
FACULTY OF ENGINEERING

Midterm Examination: ภาคการศึกษาที่ 2/2558

Date: 28 ก.พ. พ.ศ. 2559 13.30 – 15.30 น.

Subject Number: 242-210

Room: หัวหุ่น, R201

Subject Title: Programming Fundamentals II

ทฤษฏีในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทฤษฏี และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

รหัสนักศึกษา .....

เวลา 2 ชั่วโมง (120 คะแนน: 120 นาที)

ชื่อ-สกุล .....

เอกสารมีทั้งหมด 7 หน้า (ไม่รวมหน้านี้)

สำหรับอาจารย์

คำถามจำนวน 4 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียน

ไม่อนุญาต: หนังสือ และเครื่องคิดเลข

คำแนะนำ:

- เขียนชื่อ (ไม่ต้องเขียนนามสกุล) และรหัสนักศึกษา ในทุกหน้า
- พยายามทำทุกข้อ และคำตอบทั้งหมดให้ทำในข้อสอบชุดนี้
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนเมธอด หรือเขียนทั้งโปรแกรม รวมไปถึงข้อกำหนดเพิ่มเติม และหมายเหตุในข้อนั้นๆ
- การเขียนโปรแกรมในแต่ละข้อ อาจไม่ต้องเขียนตามคำสั่งย่อยทั้งหมด แต่คะแนนจะลดลงตามส่วน
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่างๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี

ข้อ	คะแนน
1 (20)	
2 (40)	
3 (25)	
4 (35)	

อ.สุรน: ผู้ออกข้อสอบ

ข้อที่ 1 ความรู้พื้นฐาน

(20 คะแนน : 20 นาที)

1.1 จงอธิบายความสัมพันธ์ของ Class กับ Object (ใช้ภาพประกอบ)

(10 คะแนน)

.....

.....

.....

.....

.....

.....

1.2 จงอธิบายความแตกต่างของ Primitive Variables กับ Reference Variables (ใช้ภาพประกอบ) (10 คะแนน)

.....

.....

.....

.....

.....

.....

## ข้อที่ 2 Classes & Collections

(40 คะแนน : 40 นาที)

จากโค้ดของฟังก์ชัน main() ที่กำหนดให้ จงเขียนคลาส Dice และ DiceBox อย่างเหมาะสม เพื่อให้โปรแกรมทำงานได้ตามคาดหวัง

```
public class Foo {
    public static void main(String args[]){
        Dice d1 = new Dice();
        System.out.println(d1);
        d1.roll();
        System.out.println(d1);

        DiceBox box = new DiceBox();
        box.add(d1);
        box.add(new Dice());
        box.add(new Dice());

        System.out.println("Before shaking: total = " + box.getTotal());
        box.shake();
        System.out.println("After shaking: total = " + box.getTotal());

        box.printDices();
    }
}
```

ดย. ผลการทำงาน

```
Face = 1
Face = 5
Before shaking: total = 7
After shaking: total = 9
1. Face = 4
2. Face = 1
3. Face = 4
```

คำอธิบายเพิ่มเติม

- Dice คือ ลูกเต๋า ที่เมื่อเริ่มการทำงานจะมีหน้าแต้มเป็น 1 แต่เมื่อถูกเขย่า หน้าแต้มจะเป็นค่าสุ่มระหว่าง 1-6
- DiceBox เปรียบเสมือน กล่องที่สามารถใส่ลูกเต๋าดได้หลายลูก เมื่อเขย่า ลูกเต๋าทั้งหมดในกล่อง ก็จะถูกกลิ้งไปมา (roll)

2.1 คลาส Dice มีข้อมูลเพิ่มเติมดังนี้

(20 คะแนน)

- a) นิยาม Constructor ที่กำหนดแต้มเริ่มต้นเป็น 1
- b) เมธอด roll() เพื่อสุ่มหน้าของลูกเต๋า
- c) นิยามฟังก์ชัน getFace() เพื่อคืนแต้มหน้าปัจจุบันของลูกเต๋า
- d) เมธอด toString() (Override เมธอดของคลาส Object) เพื่อคืนค่าข้อความในรูปแบบ F = แต้ม

\*\* โค้ดตัวอย่างการสุ่มค่าเลขจำนวนเต็ม ระหว่าง 0-9 คือ (int) (Math.random() \* 10)

2.2 คลาส DiceBox มีข้อมูลเพิ่มเติมดังนี้

(20 คะแนน)

- a) นิยาม Constructor
- b) เมธอด `add()` เพื่อเพิ่มลูกเต๋ไปในกล่อง
- c) นิยามฟังก์ชัน `shake()` เพื่อเขย่ากล่อง ทำให้ลูกเต๋าทิ้ง roll
- d) นิยามเมธอด `getTotal()` เพื่อหาผลรวมของแต้มของลูกเต๋าทิ้ง
- e) นิยามเมธอด `printDices()` เพื่อแสดงลูกเต๋าทิ้ง โดยการแสดงผลจะต้องสอดคล้องกับผลการรันที่แสดงไว้ข้างต้น

### ข้อที่ 3 ไลบรารี (String & Scanner)

(25 คะแนน : 25 นาที)

จงเขียนฟังก์ชันที่เหมาะสม เพื่อรับข้อมูลที่มีข้อความในลักษณะดังนี้ (ไม่มี // ในไฟล์ ใช้เพียงเพื่ออธิบายการทำงาน)

```
A+3      //บวก A ไป 3
B-1      //ลบ B ไป 1
C+7      //บวก C ไป 7
B+12     //บวก B ไป 12
.        //สิ้นสุดการทำงาน จะได้ A=3, B=11, C=7
```

\*\* ตัวอักษร A-C ตามด้วยเครื่องหมาย + หรือ - และตัวเลขจำนวนเต็ม

ทั้งนี้ฟังก์ชันจะประมวลผล แล้วคืนค่าเป็น HashMap โดยมี key คือ ตัวอักษร A-C และ value คือ ผลบวกของตัวเลขของตัวอักษรนั้นๆ หากตัวอักษรใด ไม่ปรากฏในข้อมูลที่รับเข้ามา ให้ถือว่า ตัวอักษรนั้นมีค่าผลบวกเป็น 0

```
s1.substring(5)      returns the substring starting at index position 5
s.charAt(0)          returns the char value at the specified index.

sc.nextLine()        Advances this scanner past the current line and returns the input that was
                    skipped.
```

#### คำตอบข้อ 3

```
public static HashMap<String, Integer> process(){
}
}
```

#### ข้อที่ 4 การสืบทอด

(35 คะแนน : 35 นาที)

คลาส Shape เป็นตัวแทนของรูปทางคณิตศาสตร์ ซึ่งประกอบด้วยชื่อ และสามารถคำนวณพื้นที่ ดังแสดงในโค้ดข้างล่าง

```
public class Shape{
    private String name;
    public Shape(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public double getArea() {
        return 0;
    }
}
```

4.1 คลาส Circle และ Rectangular ซึ่งสืบทอดจากคลาส Shape และมีการคำนวณพื้นที่อย่างเหมาะสม จงเลือกเขียนนิยามของคลาสเพียงคลาสเดียว (10 คะแนน)

- Circle มีคุณสมบัติเป็นชื่อและรัศมี โดยคำนวณพื้นที่จาก  $\text{Math.PI} * \text{radius} * \text{radius}$
- Rectangular มีคุณสมบัติเป็นชื่อ ความกว้างและความยาว โดยคำนวณพื้นที่จาก  $\text{width} * \text{height}$

คำตอบข้อ 4.1

4.2 จากโค้ดที่กำหนดให้ข้างล่าง จง (1) เติมโค้ดส่วนที่ขาดหายไป (2) เขียนนิยามฟังก์ชัน findBiggest() เพื่อแสดงชื่อของ Shape ที่มีพื้นที่ของรูปทรงเรขาคณิตใหญ่ที่สุดในลิสต์ (15 คะแนน)

```

import java.util.ArrayList;
public class ShapeMain {
    public static void main(String args[]){
        Shape shapes[] = new Shape[3];
        shapes[0] = new Rectangle("A", 5.0, 10.0);
        shapes[1] = new Circle("B", 3.0);
        shapes[2] = new Circle("C", 4.0);
        System.out.println("Biggest = " + .....);
    }
    //You may assumed that size of shapes will be at least 1
    private static Shape findBiggest(Shape[] shapes){

    }
}

```

4.3 จงอธิบายโดยใช้หลักการของ Dynamic Binding เพื่อแสดงให้เห็นเข้าใจว่า เหตุใดโค้ดข้างล่าง จึงไม่แสดงค่า 0 (10 คะแนน)

```

Shape s = new Circle("B", 3.0);
System.out.println(s.getArea());

```