## Lab 4: Class Collection and Library

## 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();

}

}

ตย. 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 กล่องที่สามารถใส่ลูกเต๋าได้หลายลูก เมื่อเขย่า ลูกเต๋าทั้งหมดในกล่อง ก็จะถูกกลิ้งไปมา (roll)

1.1 คลาส Dice มีข้อมูลเพิ่มเติมดังนี้ (20 คะแนน)

1. นิยาม Constructor ที่กำหนดแต้มเริ่มต้นเป็น 1
2. เมธอด roll() เพื่อสุ่มหน้าของลูกเต๋า
3. นิยามฟังก์ชัน getFace() เพื่อคืนแต้มหน้าปัจจุบันของลูกเต๋า
4. เมธอด toString() (Override เมธอดของคลาส Object) เพื่อคืนค่าข้อความในรูปแบบ F = แต้ม

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

1.2 คลาส DiceBox มีข้อมูลเพิ่มเติมดังนี้ (20 คะแนน)

1. นิยาม Constructor
2. เมธอด add() เพื่อเพิ่มลูกเต๋าไปในกล่อง
3. นิยามฟังก์ชัน shake() เพื่อเขย่ากล่อง ทำให้ลูกเต๋าทุกลูก roll
4. นิยามเมธอด getTotal() เพื่อหาผลรวมของแต้มของลูกเต๋าทุกลูก
5. นิยามเมธอด printDices() เพื่อแสดงลูกเต๋าทุกลูก โดยการแสดงผลจะต้องสอดคล้องกับผลการรันที่แสดงไว้ข้างต้น

**Class diagram**

**Dice**

**//Attribute**

- face : int

**//Method**

+ Dice()

+ int getFace()

+ void roll()

+ String toString()

**DiceBox**

**//Attribute**

- DiceList : ArrayList<Dice>

**//Method**

+ DiceBox()

+ void add()

+ void shake()

+ void getTotal()

+ void printDices()

**Object Diagram**

//Dice d1 = new Dice()

d1

**Dice**

face = 1

//d1.roll

d1

**Dice**

face = 5

//DiceBox box = new DiceBox()

box

**DiceBox**

ø

//box.add(d1)

box

**DiceBox**

d1

**Dice**

face = 5

//box.add(new Dice())

//box.add(new Dice())

box

**DiceBox**

d1

**Dice**

face = 1

**Dice**

face = 1

**Dice**

face = 5

//box.shake()

box

**DiceBox**

d1

**Dice**

face = 1

**Dice**

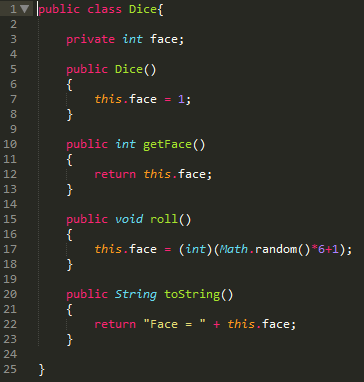
face = 4

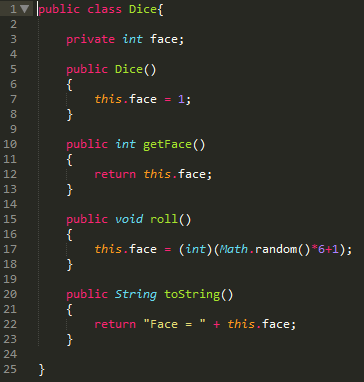
**Dice**

face = 4

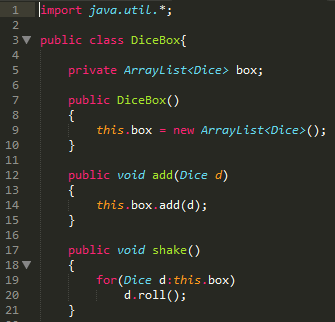
**โค๊ดโปรแกรม**

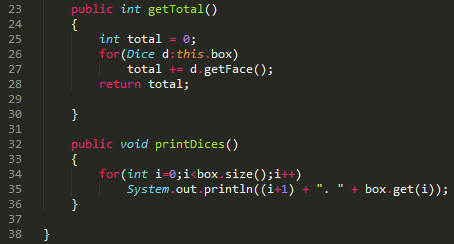
Dice.java



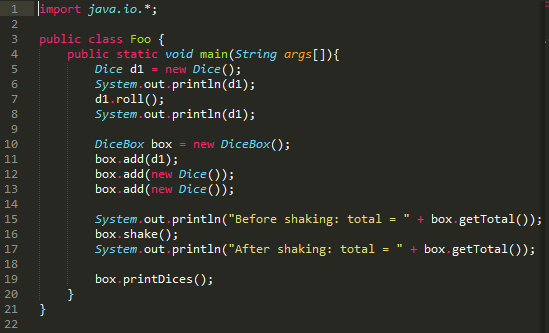


DiceBox.java

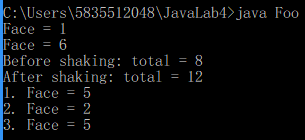




Foo.java



**ผลการรัน**



## จงเขียนนิยามฟังก์ชันชื่อ extractIt เพื่อแปลงข้อความที่กำหนดให้ เป็น String

* ข้อความที่ต้องการแปลงอยู่ในรูปแบบ url
* String ที่ฟังก์ชันส่งกลับเป็นชื่อไฟล์ และ
* หากชื่อไฟล์ขึ้นต้นด้วย ‘ \* ‘ ให้แทนที่ ‘ \* ‘ ด้วย ‘\_’ (แทนที่เฉพาะตัวอักษรแรกที่เป็น ‘\*’ เท่านั้น

ต้นแบบ method extractIt:String extractIt(String url)

ตัวอย่างข้อความ

http://example.com/files/public\_files/test.zip

www.abc.com/pub/\*aaa.html

phuket.psu.ac.th/aaa.html

www.phuket.psu.ac.th/study/study.php

ผลลัพธ์ที่ได้จากการส่งค่ากลับจาก method จะเป็น

test.zip

\_aaa.html (แทนที่ ‘\*’ ด้วย ‘\_’)

aaa.html

study.php

study.php

s1.startsWith("text") returns true if s1 starts with “text”

s1.substring(5) returns the substring starting at index position 5

s1.substring(1, 4) returns substring between positions 1 and 3

note: second argument is end position + 1

s1.split("/") returns array of strings splitting by given regular expression

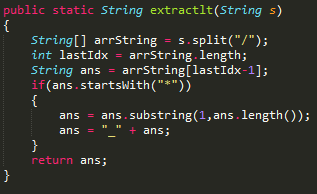
s1.trim() return new String object where any white space before or after

the s1 text has been removed

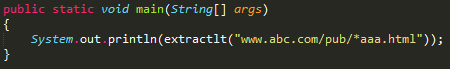
s1.length() returns the length of this string.

**โค๊ดโปรแกรม**

Method extractlt()



เรียกใช้ Method



ผลการรัน

