

แบบฝึกปฏิบัติการครั้งที่ 5

1. **Zeller's congruence** เป็นอัลกอริทึมที่คิดค้นโดย **Christian Zeller** เพื่อคำนวณหาวันของสัปดาห์ โดยสูตรคือ

$$h = \left(q + \frac{26(m+1)}{10} + k + \frac{k}{4} + \frac{j}{4} + 5j \right) \% 7$$

โดย

h คือวันของสัปดาห์ (0: วันเสาร์, 1: วันอาทิตย์, 2: วันจันทร์, 3: วันอังคาร, 4: วันพุธ, 5: วันพฤหัสบดี, 6: วันศุกร์)

q คือวันของเดือน (1-31)

m คือเดือนโดย 3: มีนาคม, 4: เมษายน, ..., 12: ธันวาคม สำหรับเดือนมกราคมและกุมภาพันธ์จะนับเป็นเดือนที่ 13 และ 14 ของปีก่อนหน้า

j คือ $\text{year}/100$

k คือ $\text{year}\%100$

ทั้งนี้ / หมายถึง **integer division**

จงเขียนคลาส **Zeller** ที่มีตัวแปรดังนี้

```
private int dayOfMonth; //เก็บวันของเดือน
```

```
private int month; //เก็บเดือน
```

```
private int year; //เก็บปี
```

และมี **enum type** ชื่อ **Day** ที่กำหนดดังนี้ **SUNDAY("Sunday")**, **MONDAY("Monday")**, **TUESDAY("Tuesday")**, **WEDNESDAY("Wednesday")**, **THURSDAY("Thursday")**, **FRIDAY("Friday")**, **SATURDAY("Saturday")**;

จากนั้นเขียนเมธอด **public Day getDayOfWeek()** ซึ่งคำนวณหาวันของสัปดาห์โดยใช้ **Zeller's congruence algorithm**

แล้วเขียนคลาสทดสอบที่รับ ปี เดือนและ วันของเดือน แล้วแสดงวันของสัปดาห์

หมายเหตุในข้อนี้ขอให้ฝึกใช้ **switch** ในส่วนของโปรแกรมที่เหมาะสม

ตัวอย่างผลลัพธ์การรัน

```
Enter year (e.g. , 2012): 2015
Enter month (1-12): 1
Enter day of the month (1-31): 25
Day of the week is Sunday
```

ตัวอย่างผลลัพธ์การรัน

```
Enter year (e. g. , 2012): 2012
Enter month (1-12): 5
Enter day of the month (1-31): 12
Day of the week is Saturday
```

2. จงเขียนคลาส **Line** ซึ่งมี **constructor** 4 แบบ

public Line(double x, double y, double m) //รับจุดหนึ่งจุดที่อยู่บนเส้น (x, y) และความชันของเส้น (m)

public Line(double x1, double y1, double x2, double y2) //รับจุดสองจุดที่อยู่บน-เส้น (x1, y1), (x2, y2)

public Line(double m, double b) //รับความชัน (m) และจุดตัดแกน (b) ในรูปสมการ $y = mx+b$

public Line(double a) //สำหรับ **vertical line** ซึ่งสมการอยู่ในรูป $x = a$

การเก็บ **instance variable** ทำได้หลายแบบแล้วแต่นิสิตจะออกแบบ แต่หากใครคิดไม่ออก ลองทำตามด้านล่างดังนี้

เก็บ **instance variable** สามตัวได้แก่

m เก็บความชันของเส้นตรง (มีค่าเป็น NaN สำหรับ **vertical line**)

b เก็บจุดตัดแกน (มีค่าเป็น NaN สำหรับ **vertical line**)

x เก็บจุดตัดแกน **x** สำหรับ **vertical line** (มีค่าเป็น NaN สำหรับเส้นตรงปกติ)

จากนั้นเขียนเมธอดดังนี้

public boolean isParallel(Line line) //ตรวจสอบว่าเส้นตรงสองเส้นขนานกันหรือไม่ ให้ตรวจสอบโดยดูจากเส้นตรงสองเส้นขนานกันหากมีความชันเท่ากัน

public boolean equals(Line line) //ตรวจสอบว่าเส้นตรงสองเส้นเท่ากันหรือไม่ ให้ตรวจสอบโดยดูจากเส้นตรงสองเส้นเท่ากันหากมีความชันเท่ากัน และจุดตัดแกนเท่ากัน

public boolean isIntersect(Line line) //ตรวจสอบว่าเส้นตรงสองเส้นตัดกันหรือไม่ ให้ตรวจสอบโดยดูจากเส้นตรงสองเส้นตัดกันหากมันไม่ขนานกัน

public Point2D.Double getIntersectionPoint(Line line) //หาจุดตัดของเส้นตรงสองเส้นหากตัดกัน ในกรณีที่เส้นตรงสองเส้นเป็นเส้นเดียวกันถือว่าจุดตัดคือทุกจุดบนเส้น (**infinity**) (ใช้ความรู้คณิตศาสตร์ม.ปลายในการหาจุดตัด)

หมายเหตุ ลองเปิด **java api** เพื่อศึกษาคلاس **Point2D.Double**

จากนั้นเขียนคลาสทดสอบให้ได้ผลลัพธ์ตามตัวอย่างด้านล่าง

ตัวอย่างผลลัพธ์การรันเมื่อสร้างเส้นตรงดังนี้

```
Line l1 = new Line(-2, 1, 1, -2);
```

```
Line l2 = new Line(-6, -2, -2, 0);
```

```
Are the two lines equals?: false  
Are the two lines parallel?: false  
Do the two lines intersect?: true  
Point of intersection: -1.33, 0.33
```

ตัวอย่างผลลัพธ์การรันเมื่อสร้างเส้นตรงดังนี้

```
Line l1 = new Line(0, 2);
```

```
Line l2 = new Line(2, 1);
```

```
Are the two lines equals?: false  
Are the two lines parallel?: false  
Do the two lines intersect?: true  
Point of intersection: 0.50, 2.00
```

ตัวอย่างผลลัพธ์การรันเมื่อสร้างเส้นตรงดังนี้

```
Line l1 = new Line(0.44, 3.4);
```

```
Line l2 = new Line(32, 32, 9);
```

```
Are the two lines equals?: false  
Are the two lines parallel?: false  
Do the two lines intersect?: true  
Point of intersection: 30.30, 16.73
```

ตัวอย่างผลลัพธ์การรันเมื่อสร้างเส้นตรงดังนี้

```
Line l1 = new Line(3, 5);
```

```
Line l2 = new Line(5);
```

```
Are the two lines equals?: false  
Are the two lines parallel?: false  
Do the two lines intersect?: true  
Point of intersection: 5.00, 20.00
```

ตัวอย่างผลลัพธ์การรันเมื่อสร้างเส้นตรงดังนี้

```
Line l1 = new Line(0, 4);
```

```
Line l2 = new Line(5);
```

```
Are the two lines equals?: false  
Are the two lines parallel?: false  
Do the two lines intersect?: true  
Point of intersection: 5.00, 4.00
```

ตัวอย่างผลลัพธ์การรันเมื่อสร้างเส้นตรงดังนี้

```
Line l1 = new Line(0.44, 3.4);
```

```
Line l2 = new Line(0.44, 5);
```

ในกรณีนี้เส้นตรงสองเส้นขนานกันจึงไม่พิมพ์จุดตัดออกมา

<pre>Are the two lines equals?: false Are the two lines parallel?: true Do the two lines intersect?: false</pre>
--

ตัวอย่างผลลัพธ์การรันเมื่อสร้างเส้นตรงดังนี้

```
Line l1 = new Line(1, -24);
```

```
Line l2 = new Line(29, 5, 53, 29);
```

ในกรณีนี้เส้นตรงสองเส้นนี้มีค่าเท่ากัน จึงขนานกันด้วย

<pre>Are the two lines equals?: true Are the two lines parallel?: true Do the two lines intersect?: false</pre>
