

แบบฝึกปฏิบัติการครั้งที่ 6

1. การคำนวณหาระยะทางของลูกกระสุนปืนใหญ่ที่ถูกยิงขึ้นฟ้าในแนวดิ่ง สามารถทำได้สองวิธี

1.1 ใช้สูตรคำนวณ $s(t) = -0.5 * g * t^2 + v_0 * t$ โดย $g = 9.81 \text{ m/sec}^2$, v_0 คือความเร็วตั้งต้น, t คือระยะเวลา

1.2 simulation โดยในแต่ละรอบเริ่มด้วย

1.2.1 คำนวณการเคลื่อนที่ของลูกบอลในระยะเวลาอันสั้น ๆ (Δt) จากสมการ $\Delta s = v * \Delta t$ ซึ่งเราจะกำหนด Δt

ให้มีค่าเป็น 0.01 แล้วอัปเดตค่าของระยะทางโดย $s = s + v * \Delta s$

1.2.2 อัปเดตความเร็วโดย $v = v - g * \Delta t$ เนื่องจากความเร็วจะค่อย ๆ ลดลง $g * \Delta t$ ในช่วงเวลาสั้น ๆ

ในรอบถัดไปใช้ความเร็วที่เพิ่งอัปเดตใหม่นี้ในการคำนวณระยะทาง

จงเขียนคลาส CannonBall ซึ่งมีตัวแปรดังนี้

```
private double initV; //ความเร็วตั้งต้น
```

```
private double simS; //ระยะทางที่คำนวณได้จากวิธี simulation
```

```
private double simT; //เวลาที่ใช้ในวิธี simulation
```

```
public static final double g = 9.81;
```

และเขียนเมธอด

```
public void simulatedFlight() //คำนวณหาระยะทางที่ลูกกระสุนปืนใหญ่เคลื่อนที่จนกระทั่งความเร็วเป็น 0
```

และตกกลับลงบนพื้นโลก โดยอัปเดตระยะทางและความเร็ว 100 ครั้งต่อวินาที และพิมพ์ระยะทางที่ลูกบอล

เคลื่อนไปได้ทุก ๆ 1 วินาที และระยะทางสุดท้ายก่อนตกกลับลงมา

```
public double calculusFlight(double t) //คำนวณระยะทางที่ลูกกระสุนปืนใหญ่เคลื่อนที่ได้ หากใช้
```

ระยะเวลา t

```
public double getSimulatedTime() //คืนระยะเวลาทั้งหมดที่ใช้จนลูกกระสุนปืนตกกลับลงพื้นในวิธี
```

simulation

```
public double getSimulatedDistance() //คืนระยะทางที่ลูกกระสุนปืนใหญ่เคลื่อนที่ไปก่อนตกกลับลงพื้น
```

กำหนด main ดังนี้

```
public class CannnonBallTester {  
    public static void main(String[] args) {
```

```
CannonBall ball = new CannonBall(100); //กำหนดความเร็วตั้งต้นให้ลูกกระสุนปืนใหญ่มีค่าเป็น  
100 m/sec  
    ball.simulatedFlight();  
    System.out.println(ball.calculusFlight(ball.getSimulatedTime()));  
}  
}
```

ตัวอย่างผลลัพธ์การรัน

Distance on 1 sec: 95.144

Distance on 2 sec: 180.478

Distance on 3 sec: 256.002

Distance on 4 sec: 321.716

Distance on 5 sec: 377.620

Distance on 6 sec: 423.714

Distance on 7 sec: 459.998

Distance on 8 sec: 486.472

Distance on 9 sec: 503.136

Distance on 10 sec: 509.990

Final distance: 510.184 Total time: 10.20

Distance from calculus equation: 509.684

หมายเหตุ ผลลัพธ์จากสองวิธีจะไม่เท่ากันเป๊ะเนื่องจากสมการแคลคูลัสจะคำนวณได้ค่าประมาณ

2. จงเขียนคลาส Game สำหรับเล่นเกมเป่ายิ้งฉุบ ให้ได้ผลลัพธ์ดังตัวอย่างผลลัพธ์การรัน โดยให้ผู้เล่นใส่ตัวเลข 0 หากต้องการออกตอน 1 หากต้องการออกกระดาษ 2 หากต้องการออกกรรไกร (หากผู้เล่นใส่ข้อมูลที่ไม่ใช่สามเลขนี้ให้วนรอบไปเรื่อย ๆ) และแข่งกับคอมพิวเตอร์ โดยคอมพิวเตอร์จะสุ่มค่าจากสามเลขนี้ กติกาคือหากฝ่ายใดมีคะแนนมากกว่าอีกฝ่ายหนึ่ง 2 คะแนนจะเป็นผู้ชนะ

กำหนด main ดังนี้

```
public class RockPaperScissorTester {  
  
    public static void main(String[] args) {  
  
        Game game = new Game();  
  
        game.play();  
  
    }  
  
}
```

ตัวอย่างผลลัพธ์การรัน

```
Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 0  
You enter: ROCK  
Computer: SCISSORS  
You win!  
Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 3  
Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: ฟ  
Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 1  
You enter: PAPER  
Computer: SCISSORS  
You lose!  
Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 0  
You enter: ROCK
```

Computer: SCISSORS

You win!

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 1

You enter: PAPER

Computer: PAPER

It's a tie.

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 0

You enter: ROCK

Computer: PAPER

You lose!

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 0

You enter: ROCK

Computer: PAPER

You lose!

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 2

You enter: SCISSORS

Computer: ROCK

You lose!

Too bad! You lose.

User Score: 2

Computer score: 4

ตัวอย่างผลลัพธ์การรัน

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 0

You enter: ROCK

Computer: PAPER

You lose!

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 1

You enter: PAPER

Computer: ROCK

You win!

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 2

You enter: SCISSORS

Computer: ROCK

You lose!

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 1

You enter: PAPER

Computer: PAPER

It's a tie.

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 0

You enter: ROCK

Computer: SCISSORS

You win!

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 1

You enter: PAPER

Computer: ROCK

You win!

Enter 0 for ROCK, 1 for PAPER, 2 for SCISSORS: 0

You enter: ROCK

Computer: SCISSORS

You win!

Congrats! You win.

User Score: 4

Computer score: 2

3. จงเขียนคลาส CityGrid แทนเมืองแห่งหนึ่ง

ตัวอย่างเช่นเมืองขนาด 10 x 10

0	1	2	3	4	5	6	7	8	9	10	
											0
											1
											.
											.
											.
											10

และมีตัวแปรดังนี้

```
private int xCoor; //เก็บพิกัดของชายผู้หนึ่งในแนวแกน x
```

```
private int yCoor; //เก็บพิกัดของชายผู้หนึ่งในแนวแกน y
```

```
private int gridSize; //เก็บขนาดของเมือง
```

โดยเริ่มต้นชายผู้หนึ่งจะยืนอยู่ตำแหน่งตรงกลางเมือง เช่นหากเมืองขนาด 10 x 10 ชายผู้หนึ่งจะยืนที่ตำแหน่ง (5,5)

จากนั้นเขียนเมธอดดังนี้

```
public void walk() // ชายผู้หนึ่งจะเดินเพียงหนึ่งก้าว โดยสามารถเดินได้ 4 ทิศ คือเดินขึ้น (y--) ลง (y++) ซ้าย (x--) ขวา (x++)
```

```
public boolean isInCity() // คืนค่าจริง หากชายผู้หนึ่งยังคงอยู่ในเมือง
```

```
public void reset() // reset ตำแหน่งของชายผู้หนึ่งให้กลับมาอยู่ที่กลางเมือง
```

จากนั้นใน main ให้เขียนคำสั่งเพื่อสร้างเมืองขนาด 10×10 และให้ชายผู้หนึ่งเดิน 1000 ก้าว หากชายผู้หนึ่งเดินออกนอกเมืองไปก่อนที่จะครบ 1000 ก้าว ให้หยุดการเดิน แล้วเริ่มใหม่ทำซ้ำเช่นนี้ 10000 ครั้ง และหาค่าเฉลี่ยว่าชายผู้หนึ่งเดินได้กี่ก้าวก่อนจะออกนอกเมือง และจำนวนก้าวที่เดินได้มากที่สุดก่อนออกนอกเมือง

ตัวอย่างผลลัพธ์การรัน

Average number of steps that a person can take and is still in the city: 41.67

Maximum number of steps that a person can take and is still in the city: 301