

CHAPTER 10

อาเรย์สองมิติ (Two Dimensional Arrays)

1. การประกาศและสร้างอาเรย์สองมิติ

1. การประกาศและสร้างอาเรย์สองมิติโดยใช้ Initialized List

การประกาศและสร้างอาเรย์สองมิติโดยใช้ Initialized List เป็นการสร้างอาเรย์สองมิติแบบง่าย ซึ่งจะประกาศใช้ได้ก็ต่อเมื่อทราบค่าทุกค่าที่จะเก็บลงไปในอาเรย์แล้ว โดยมีรูปแบบคำสั่งดังนี้

<ประเภทข้อมูล> <ชื่ออาเรย์>[][] = {{ สมาชิก , ..., สมาชิก }, ...};

หลังชื่ออาเรย์ต้องมี

สัญลักษณ์ [][]

เช่น `int num[][] = {{1,0},{2,1},{2,4}};` สามารถวาดเป็น โครงสร้างตาราง ได้ดังนี้

1	0
2	1
2	4

อาเรย์ num มีขนาด 3 x 2 ซึ่งประกอบไปด้วย

- แถว (แนวนอน) จำนวน 3 แถว
- หลัก (แนวตั้ง) จำนวน 2 หลัก

- ในเซตใหญ่มีเซตย่อย 3 เซต นั่นคือมี 3 แถว
- ในแต่ละเซตย่อยมีสมาชิก 2 ค่า นั่นคือมี 2 หลัก

โจทย์ข้อที่ 1 [ระดับง่าย] จงเขียนคำสั่งภาษาจาวาเพื่อประกาศและสร้างอาเรย์สองมิติโดยใช้ Initialized List ตามรายละเอียดที่กำหนดให้ต่อไปนี้ (10 คะแนน)

- 1) ประกาศและสร้างอาเรย์สองมิติชื่อ a ขนาด 2 x 3 โดยให้สมาชิกทุกตัวมีค่าเป็น 0 (2 คะแนน)

- 2) ประกาศและสร้างอาเรย์สองมิติชื่อ b ขนาด 5 x 2 โดยให้สมาชิกทุกตัวมีค่าเป็น 1 (2 คะแนน)

- 3) ประกาศและสร้างอาเรย์สองมิติชื่อ s ขนาด 3 x 1 โดยให้สมาชิกทุกตัวเป็นคำว่า "Java" (2 คะแนน)

- 4) ประกาศและสร้างอาเรย์สองมิติชื่อ t ขนาด 1 x 3 โดยให้สมาชิกทุกตัวเป็นคำว่า "Java" (2 คะแนน)

- 5) ประกาศและสร้างอาเรย์สองมิติชื่อ em ประเภทจำนวนจริง ขนาด 0 x 0 (2 คะแนน)

โจทย์ข้อที่ 2 [ระดับง่าย] จงวาดรูปโครงสร้างตารางของอาร์เรย์สองมิติที่กำหนดให้ต่อไปนี้ พร้อมทั้งกำหนดค่าให้กับสมาชิกแต่ละช่องของตารางให้ถูกต้อง (6 คะแนน)

1) `int m[][] = {{1,2,3,4},{2,3,4,5},{3,4,5,6},{4,5,6,7}};` (2 คะแนน)

2) `int n[][] = {{1},{2,3},{3,4,5},{4,5,6,7}};` (2 คะแนน)

3) `int p[][] = {{1,2,3},{2,3},{3,4,5,6},{4},{5,6}};` (2 คะแนน)

ข้อที่ 1 อาร์เรย์ m[][]	ข้อที่ 2 อาร์เรย์ n[][]	ข้อที่ 3 อาร์เรย์ p[][]

2. การประกาศและสร้างอาร์เรย์สองมิติโดยการ new

การประกาศและการสร้างอาร์เรย์สองมิติโดยการ new นี้จะเป็นวิธีแบบทั่วไปที่ใช้ในภาษาจาวา ซึ่งมีขั้นตอนดังนี้

1) การประกาศตัวแปรอาร์เรย์สองมิติ

<ประเภทข้อมูล> <ชื่ออาร์เรย์> [][];

จำนวนมิติให้นับที่จำนวนคู่ของเครื่องหมาย [] เช่น
ถ้าเป็นอาร์เรย์สองมิติจะต้องมี [] จำนวน 2 คู่

เช่น `int a[][];`



2) การสร้างอาร์เรย์สองมิติ

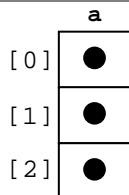
(1) สร้างแถวของอาร์เรย์

หลังชื่ออาร์เรย์ไม่ต้องมีเครื่องหมาย [][]

<ชื่ออาร์เรย์> = new <ประเภทข้อมูล> [<ขนาดแถว>] [];

ภายใน [] คู่ที่สองไม่ต้องระบุค่าใดๆ (ให้เว้นว่างเอาไว้)

เช่น `a = new int[3][];`

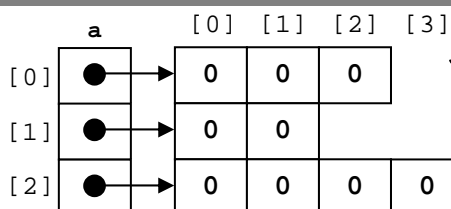


สร้างอาร์เรย์ a ให้มีขนาดเป็น 3 แถว แต่ยังไม่ระบุขนาดของหลัก (เว้นว่างเอาไว้)

(2) สร้างหลักหรือคอลัมน์ของอาร์เรย์ (เริ่มสร้างหลักของทุกแถว โดยเริ่มตั้งแต่แถวแรกจนถึงแถวสุดท้าย)

<ชื่ออาร์เรย์> [<ตำแหน่งแถว>] = new <ประเภทข้อมูล> [<ขนาดหลัก>];

เช่น `a[0] = new int[3];`
`a[1] = new int[2];`
`a[2] = new int[4];`



แถวที่ 1 (ตำแหน่ง 0) มี 3 หลัก
แถวที่ 2 (ตำแหน่ง 1) มี 2 หลัก
แถวที่ 3 (ตำแหน่ง 2) มี 4 หลัก

- 3) อาร์เรย์สองมิติที่ได้หลังจากการประกาศและการสร้างในขั้นตอนที่ 1 และ 2

a[0][0]	a[0][1]	a[0][2]	
a[1][0]	a[1][1]		
a[2][0]	a[2][1]	a[2][2]	a[2][3]

ถ้าประกาศและสร้างอาร์เรย์แบบแยกส่วนจะสามารถสร้างอาร์เรย์สองมิติที่บิดเบี้ยว โดยที่มีจำนวนช่องของแถวแต่ละแถวไม่เท่ากัน

- 4) การประกาศและการสร้างอาร์เรย์สองมิติโดยรวมขั้นตอนที่ 1 และ 2 เข้าด้วยกัน (มี 2 วิธีให้เลือกใช้)

<ประเภทข้อมูล> <ชื่ออาร์เรย์>[][] = new <ประเภทข้อมูล>[<ขนาดแถว>][<ขนาดหลัก>];

หรือ

<ประเภทข้อมูล> [][] <ชื่ออาร์เรย์> = new <ประเภทข้อมูล>[<ขนาดแถว>][<ขนาดหลัก>];

เช่น int a[][] = new int[3][4];
int [][] a = new int[3][4];

เครื่องหมาย [][] จะอยู่หน้าหรือหลังชื่ออาร์เรย์ก็ได้

หมายเหตุ ขนาดแถวและขนาดหลักต้องระบุเป็นตัวเลขจำนวนเต็มเท่านั้น

- 5) ข้อสังเกตในการประกาศและสร้างอาร์เรย์สองมิติ

ข้อสังเกต 3 คือ ขนาดหรือความยาวของแถว (ขนาดของมิติที่ 1) → แถวคือช่องในแนวนอน

4 คือ ขนาดหรือความยาวของหลัก (ขนาดของมิติที่ 2) → หลักคือช่องในแนวตั้ง

อย่าสับสน ตำแหน่งสมาชิกของอาร์เรย์ (สมาชิกตำแหน่งที่) เริ่มต้นที่ 0

ลำดับสมาชิกของอาร์เรย์ (สมาชิกลำดับที่ / สมาชิกตัวที่) เริ่มต้นที่ 1

หมายเหตุ ถ้าต้องการอาร์เรย์แบบที่บิดเบี้ยวจะไม่สามารถประกาศและสร้างอาร์เรย์แบบขั้นตอนที่ 4 นี้ได้ ซึ่งจะต้องใช้การประกาศและสร้างอาร์เรย์แบบแยกส่วนแทน

- 6) อาร์เรย์สองมิติที่ได้หลังจากการประกาศและการสร้างในขั้นตอนที่ 4

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]

ถ้าประกาศและสร้างอาร์เรย์แบบรวมจะได้อาร์เรย์สองมิติที่มีจำนวนช่องของแถวแต่ละแถวเท่ากันทุกแถว โดยไม่มีการบิดเบี้ยว

โจทย์ข้อที่ 3 [ระดับง่าย] จงเขียนคำสั่งภาษาจาวาเพื่อประกาศและสร้างอาร์เรย์ต่อไปนี้ (12 คะแนน)

- 1) ประกาศและสร้างอาร์เรย์ชื่อ matrix เป็นเมตริกซ์เก็บค่าความจริงขนาด 5 x 8 (2 คะแนน)

- 2) ประกาศและสร้างอาร์เรย์ชื่อ chess เพื่อสร้างตารางหมากรุก (2 คะแนน)

- 3) ประกาศและสร้างอาร์เรย์ชื่อ `tranMatrix` เป็นเมตริกซ์ประเภทจำนวนเต็มที่เกิดจากการทรานสโพสของเมตริกซ์ขนาด 9×4 (2 คะแนน)

- 4) ประกาศและสร้างอาร์เรย์ชื่อ `data` เก็บคะแนนรวมของรายวิชา 2110101 ของนิสิต 300 คน โดยในแต่ละคนประกอบไปด้วยคะแนนการบ้าน คะแนนกลางภาค และคะแนนปลายภาค (2 คะแนน)

- 5) สร้างจำนวนช่องของแถวลำดับที่ 8 ของอาร์เรย์ `n` ชนิดจำนวนเต็มทีประกาศไว้ก่อนหน้านี้แล้ว ให้มีตำแหน่งสูงสุดได้ถึงตำแหน่งที่ 3 (2 คะแนน)

- 6) สร้างจำนวนช่องของแถวตำแหน่งที่ 4 ของอาร์เรย์ `n` ชนิดจำนวนเต็มทีประกาศไว้ก่อนหน้านี้แล้ว ให้มีจำนวนช่องทั้งหมด $x^2 + 1$ ช่อง (2 คะแนน)

โจทย์ข้อที่ 4 [ระดับปานกลาง] จงเขียนเมทอด `getTable(...)` เพื่อคืนค่าตารางที่ใช้สำหรับเก็บปริมาณน้ำฝนตั้งแต่วันที่ 1 มกราคม ของ พ.ศ. ปัจจุบัน ถึงวันที่ 31 ธันวาคม ของ พ.ศ. ปัจจุบัน โดยแบ่งการจัดเก็บออกเป็น 12 เดือน และในแต่ละเดือนก็แบ่งตามจำนวนวันของเดือนนั้นๆ (12 คะแนน)

3. การหาขนาดและความยาวของอาร์เรย์สองมิติ

การหาขนาดของอาร์เรย์สองมิติจะใช้คำสั่ง `.length` เหมือนกับการหาขนาดของอาร์เรย์หนึ่งมิติแต่จะต้องระบุตำแหน่งของแถวหรือหลักที่ต้องการจะหาขนาดด้วย

- 1) การหาขนาดของแถว (ขนาดของมิติที่ 1)

`<ชื่ออาร์เรย์>.length`

- 2) การหาขนาดของหลัก (ขนาดของมิติที่ 2)

`<ชื่ออาร์เรย์>[<ตำแหน่งแถว>].length`

	[0]	[1]	[2]	[3]
[0]	0	0	0	0
[1]	0	0	0	0
[2]	0	0	0	0

ตัวอย่างโปรแกรม

```
1 int[][] a = new int[3][4];
2 System.out.println(a.length); → 3
3 System.out.println(a[0].length); → 4
4 System.out.println(a[1].length); → 4
5 System.out.println(a[2].length); → 4
6 System.out.println(a[3].length); → Error
```

ความยาวแถวของอาร์เรย์ a

ความยาวหลักของแถวตำแหน่งที่ 2

หรือแถวที่ 3 ของอาร์เรย์ a

ไม่มีแถวตำแหน่งที่ 3

4. การอ้างอิงค่า การกำหนดค่า และการแสดงผลของอาร์เรย์สองมิติ

- 1) การอ้างอิงค่าจากอาร์เรย์ มีรูปแบบคำสั่งดังนี้

`<ประเภทข้อมูล> <ชื่อตัวแปร> = <ชื่ออาร์เรย์>[<ตำแหน่งแถว>][<ตำแหน่งหลัก>];`

เช่น `int n = num[0][0];`

(ให้ตัวแปร n ประเภทจำนวนเต็ม เก็บค่าจากอาร์เรย์ num แถวตำแหน่งที่ 0 และหลักตำแหน่งที่ 0)

- 2) การกำหนดค่าลงไปในอาร์เรย์ มีรูปแบบคำสั่งดังนี้

`<ชื่ออาร์เรย์>[<ตำแหน่งแถว>][<ตำแหน่งหลัก>] = <ค่าข้อมูล>;`

เช่น `num[0][0] = 13;` (ให้อาร์เรย์ num แถวตำแหน่งที่ 0 และหลักตำแหน่งที่ 0 มีค่าเท่ากับ 13)

- 3) การแสดงผลของอาร์เรย์ (สองมิติ) จะใช้คำสั่ง `for` ซ้อน `for` หรือ `while` ซ้อน `while` ในการเข้าไปยังสมาชิกแต่ละตัวในอาร์เรย์ตั้งแต่สมาชิกตัวแรก (แถวตำแหน่งที่ 0 หลักตำแหน่งที่ 0) จนถึงสมาชิกตัวสุดท้าย (แถวตำแหน่งที่ `length - 1` หลักตำแหน่งที่ `length - 1`) ดังตัวอย่างต่อไปนี้

```
1 int a[][] = new int[5][8];
2 for (int i = 0; i < a.length; i++) {
3     for (int j = 0; j < a[i].length; j++) {
4         a[i][j] = 13;
5     }
6 }
```

คำสั่ง `for i` วน
ตั้งแต่แถวแรกจนถึง
แถวสุดท้าย

คำสั่ง `for j` วนตั้งแต่หลักแรกจนถึง
หลักสุดท้ายของแถวตำแหน่งที่ i ใดๆ

เริ่มพิจารณาที่สมาชิกในแถวแต่ละแถวของอาร์เรย์และตาม
ด้วยสมาชิกแต่ละตัว (แต่ละหลัก) ของอาร์เรย์ ณ แถวนั้นๆ

โจทย์ข้อที่ 5 [ระดับง่าย] จงเขียนคำสั่งภาษาจาวาเพื่ออ้างอิงและกำหนดค่าจากอาเรย์ต่อไปนี้ (16 คะแนน)

- 1) ให้ตัวแปร `x` เก็บข้อมูลประเภทจำนวนเต็มจากอาเรย์ `num` ตัวที่ 50 ของแถวตำแหน่งที่ 51 (2 คะแนน)

- 2) ให้ตัวแปร `c` เก็บข้อมูลประเภทอักขระจากอาเรย์ `code` ตำแหน่งที่ 60 ของแถวที่ 10 (2 คะแนน)

- 3) ให้ตัวแปร `var1` เก็บข้อมูลประเภทจำนวนเต็มจากอาเรย์ `bank` แถวที่ 1 หลักที่ 2 (2 คะแนน)

- 4) ให้ตัวแปร `var2` เก็บข้อมูลประเภทจำนวนจริงจากอาเรย์ `bank` ตัวสุดท้ายของแถวที่ 2 (2 คะแนน)

- 5) ให้ค่าสมาชิกหลักที่ 5 แถวที่ 7 ของอาเรย์ `code` เท่ากับค่าจากตัวแปร `x` (2 คะแนน)

- 6) ให้ค่าสมาชิกตำแหน่งที่ 9 ของแถวที่ 1 ของอาเรย์ `sName` เท่ากับค่าจากตัวแปร `s1` (2 คะแนน)

- 7) ให้ค่าสมาชิกตัวที่ 3 ของแถวที่ 2 ของอาเรย์ `sName` เท่ากับค่าจากตัวแปร `s2` (2 คะแนน)

- 8) ให้ค่าสมาชิกแถวสุดท้ายและหลักสุดท้ายของอาเรย์ `a` เท่ากับค่าจากตัวแปร `y` (2 คะแนน)

โจทย์ข้อที่ 6 [ระดับง่าย] จงเขียนและเรียกใช้เมทอด `setArray(...)` ที่สมบูรณ์เพื่อรับอาเรย์สองมิติชนิดจำนวนเต็มเข้ามาหนึ่งตัว แล้วกำหนดค่าสมาชิกแต่ละตัวของอาเรย์ให้มีค่าเท่ากับผลคูณของหมายเลขแถวและหลักของช่องนั้นๆ เช่น อาเรย์แถวที่ 3 หลักที่ 4 เก็บค่า 12 (เกิดจาก 3×4) เป็นต้น (12 คะแนน)

```
public class The2DArray {
```

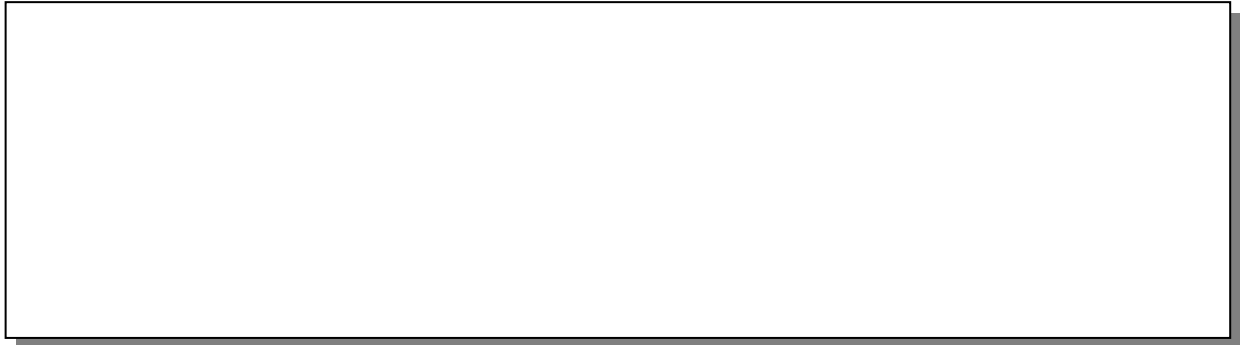
```
public static void main(String[] args) {
```

```
    } //End of main  
} //End of class
```

โจทย์ข้อที่ 7 [ระดับปานกลาง] จงเขียนและเรียกใช้เมทอด `isArrayEquals(...)` ที่สมบูรณ์เพื่อรับอาร์เรย์สองมิติชนิดจำนวนเต็ม 2 อาร์เรย์ใดๆ เข้ามาทางพารามิเตอร์ เพื่อใช้ตรวจสอบว่าอาร์เรย์ทั้งสองเท่ากันหรือไม่ โดยอาร์เรย์ที่รับเข้ามาอาจจะเป็นอาร์เรย์แถวที่มีจำนวนหลักไม่เท่ากันทุกแถว (15 คะแนน)

```
public class EqualityOf2DArray {
```

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

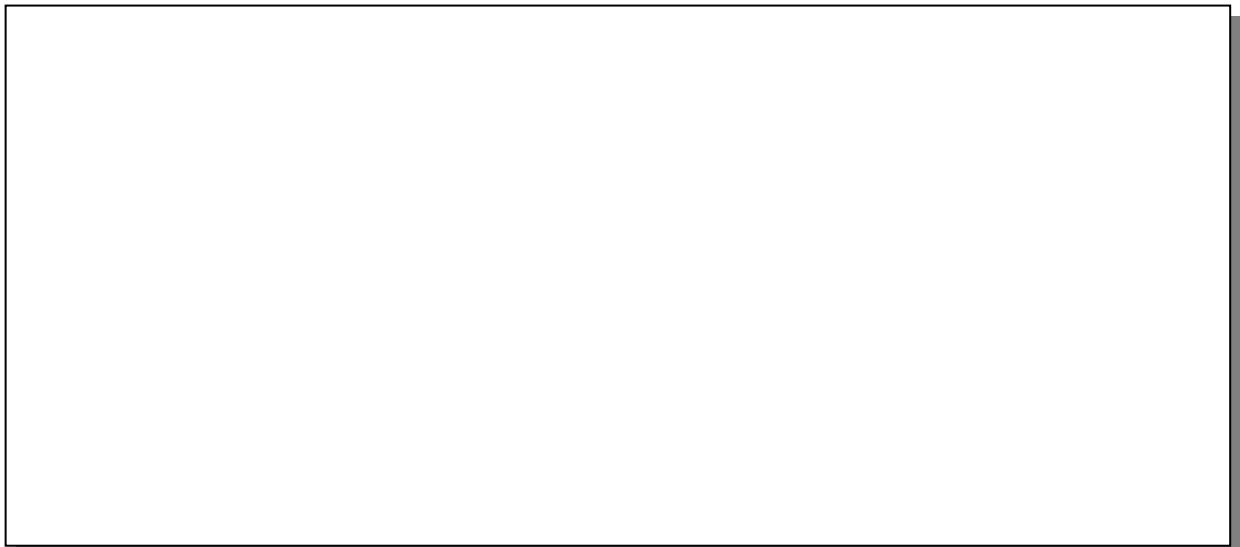


```
    } //End of main  
} //End of class
```

โจทย์ข้อที่ 8 [ระดับปานกลาง - ระดับยาก] อาเรย์ test เป็นอาเรย์สองมิติที่จัดเก็บข้อมูลประเภทจำนวนจริง 365 ค่า ซึ่งเป็นผลการทดลองทางด้านวิศวกรรมศาสตร์เป็นระยะเวลา 1 ปี ซึ่งถูกบันทึกไว้ตั้งแต่วันที่ 1 กันยายน ปีนี้ จนถึงวันที่ 31 สิงหาคม ปีหน้า จงเขียนคลาส DataExperiment ให้สมบูรณ์เพื่อเก็บบันทึกผลการทดลองดังกล่าว พร้อมทั้งคำนวณหาผลลัพธ์ต่างๆ ตามเมทอดดังต่อไปนี้ (50 คะแนน)

- 1) เมทอด max(...) คำนวณค่าสูงสุดของผลการทดลอง (8 คะแนน)
- 2) เมทอด min(...) คำนวณค่าต่ำสุดของผลการทดลอง (8 คะแนน)
- 3) เมทอด middleRange(...) คำนวณค่ากึ่งกลางพิสัยของผลการทดลอง (สูงสุด+ต่ำสุด / 2) (8 คะแนน)
- 4) เมทอด meanOfMonth(...) คำนวณค่าเฉลี่ยของผลการทดลองในแต่ละเดือนตามเดือนที่ระบุ (8 คะแนน)
- 5) เมทอด dataOfDay(...) คำนวณค่าผลการทดลองในแต่ละวันตามวันและเดือนที่ระบุ (8 คะแนน)
- 6) เมทอด main(...) ประกาศและสร้างอาเรย์ test ที่มีขนาด 12 แถวโดยแต่ละแถวยาวเท่ากับจำนวนวันของเดือนนั้นๆ โดยให้แถวแรกเป็นเดือนกันยายน แล้วรับค่าจากแป้นพิมพ์เก็บลงไปในอาเรย์ทุกช่อง พร้อมทั้งเรียกใช้งานเมทอดอื่นๆ ทุกเมทอดที่นี้ (กำหนดการเรียกใช้ได้อย่างอิสระ) (10 คะแนน)

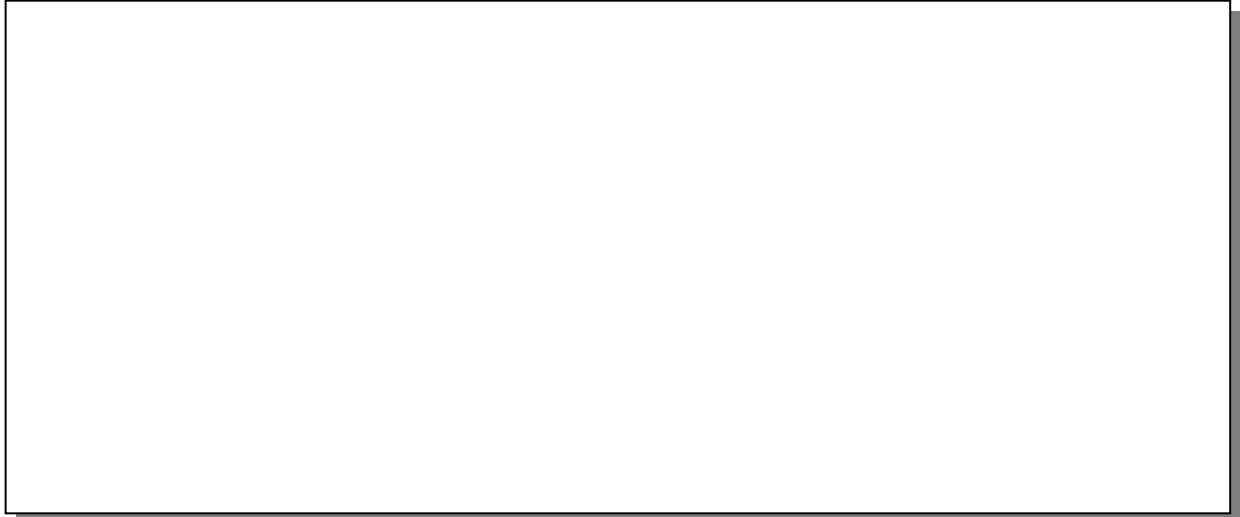
```
import java.util.Scanner;  
import jlab.graphics.DWindow;  
public class DataExperiment {  
    public static double max(double t[][]) {
```



```
    } //End of max(...)
```

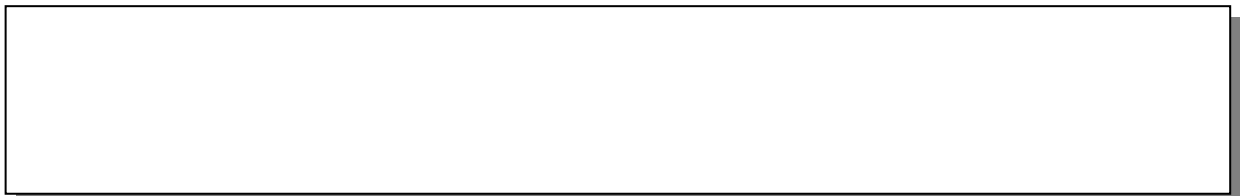


```
public static double min(double t[][]) {
```



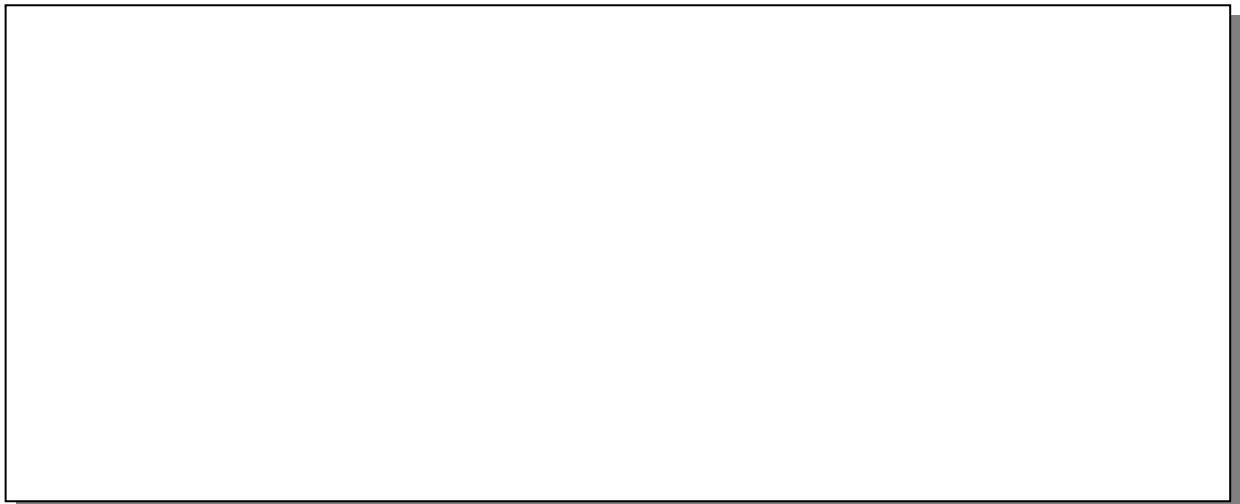
```
} //End of min(...)
```

```
public static double middleRange(double t[][]) {
```



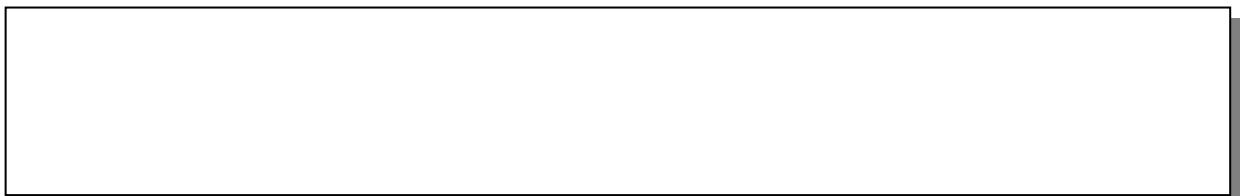
```
} //End of avgRange(...)
```

```
public static double meanOfMonth(double t[][], int m) {
```



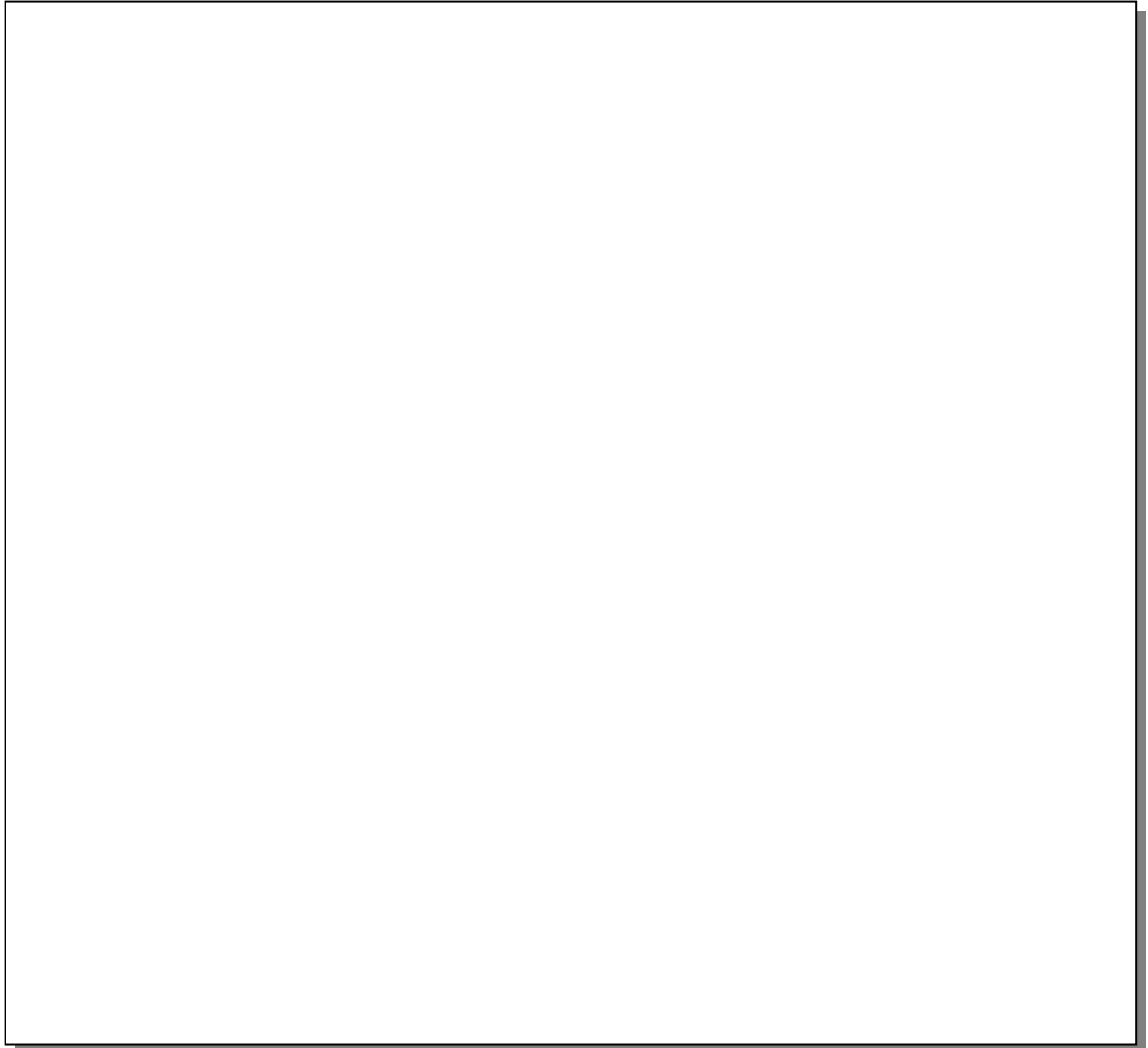
```
} //End of meanOfMonth(...)
```

```
public static double dataOfDay(double t[][], int d, int m) {
```



```
} //End of dataOfDay(...)
```

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



```
    } //End of main(...)
} //End of Class
```

โจทย์ข้อที่ 9 [ระดับง่าย – ระดับยาก] จงเขียนคลาส MatrixOperation เพื่อดำเนินการกับเมตริกซ์ซึ่งได้แก่ การบวกเมตริกซ์ การคูณเมตริกซ์ และการทรานสโพสเมตริกซ์ โดยมีรายละเอียดของเมทีอดต่างๆ ดังต่อไปนี้ (60 คะแนน)

$$\text{กำหนดให้เมตริกซ์ } a = \begin{bmatrix} 1 & 3 & 5 & 9 \\ 2 & 6 & 4 & 3 \end{bmatrix} \quad \text{เมตริกซ์ } b = \begin{bmatrix} 3 & 6 \\ 4 & 8 \\ 1 & 0 \end{bmatrix} \quad \text{เมตริกซ์ } c = \begin{bmatrix} 0 & 2 & 4 \\ 5 & 3 & 1 \\ 7 & 11 & 2 \\ 6 & 6 & 9 \end{bmatrix}$$

$$\text{กำหนดให้เมตริกซ์ } m = (2a)^T \times \left(\frac{1}{4}b\right)^T - c$$

```
import java.util.Scanner;
public class MatrixOperation {
    //[ระดับปานกลาง] เขียนเมทีอด addMatrix(...) ที่สมบูรณ์เพื่อรับเมตริกซ์ชนิดจำนวนจริงสองตัว
    ใดๆ ขนาดเท่ากันเข้ามา แล้วคำนวณหาผลบวกของสองเมตริกซ์นั้นพร้อมทั้งคืนค่ากลับ (10 คะแนน)
```

//[ระดับปานกลาง] เขียนเมทอด `mulScalarMatrix(...)` ที่สมบูรณ์เพื่อรับเมตริกซ์ชนิดจำนวนจริง
หนึ่งตัวและค่าคงที่ที่เป็นจำนวนจริงอีกหนึ่งค่า แล้วคำนวณหาผลคูณของเมตริกซ์และค่าคงที่นั้น
พร้อมทั้งคืนค่ากลับ (10 คะแนน)

//[ระดับยาก] เขียนเมทอด `transposeMatrix(...)` ที่สมบูรณ์เพื่อรับเมตริกซ์ชนิดจำนวนจริงเข้ามา
แล้วคำนวณหาผลการทรานสโพสของเมตริกซ์นั้นพร้อมทั้งคืนค่ากลับ (10 คะแนน)

//[ระดับยาก] เขียนเมทอด `mulMatrix(...)` ที่สมบูรณ์เพื่อรับเมตริกซ์ชนิดจำนวนจริงสองตัวใดๆ เข้ามาแล้วคำนวณหาผลคูณของสองเมตริกซ์นั้นพร้อมทั้งคืนค่ากลับ (10 คะแนน)

//[ระดับง่าย] เขียนเมทอด `printMatrix(...)` ที่สมบูรณ์เพื่อรับเมตริกซ์ชนิดจำนวนจริงเข้ามา แล้วแสดงค่าสมาชิกของเมตริกซ์นั้นขึ้นบนจอภาพให้สวยงาม (10 คะแนน)

//[ระดับง่าย] เขียนเมทอด `main(...)` ให้สมบูรณ์ โดยประกาศและสร้างเมตริกซ์ `a`, `b` และ `c` ที่กำหนดให้แบบ `Initialized List` พร้อมทั้งคำนวณหาผลลัพธ์ของเมตริกซ์ `x` โดยการเรียกใช้งานเมทอดต่างๆ ที่ได้สร้างไว้ก่อนหน้านี้ พร้อมทั้งแสดงค่าของเมตริกซ์ทุกตัวขึ้นบนจอภาพ (10 คะแนน)

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

```
    } //End of main  
} //End of class
```

โจทย์ข้อที่ 10 [ระดับยาก] จงเขียนเมทอด `splitRowOfMatrix(...)` ที่สมบูรณ์เพื่อดึงสมาชิกทุกตัวของเมตริกซ์ที่รับเข้ามา ตามหมายเลขแถวที่ระบุ เช่น `a[][] = {{1,2}, {2,3,4}, {1,3,5,6}}` จะได้ `splitRowOfMatrix(a,1) = {1,2}` และ `splitRowOfMatrix(a,2) = {2,3,4}` เป็นต้น แต่ถ้าระบุหมายเลขแถวผิดพลาดจะได้ `splitRowOfMatrix(a,4) = { }` (10 คะแนน)

โจทย์ข้อที่ 11 [ระดับยาก] จงเขียนเมทอด `splitColumnOfMatrix(...)` ที่สมบูรณ์เพื่อดึงสมาชิกทุกตัวของเมตริกซ์ที่รับเข้ามาตามหมายเลขหลักที่ระบุ เช่น `a[][] = {{1,2,2}, {2,3,4}, {1,3,5}}` จะได้ `splitColumnOfMatrix(a,1) = {1,2,1}` และ `splitColumnOfMatrix(a,3) = {2,4,5}` เป็นต้น แต่ถ้าระบุหมายเลขหลักผิดพลาดจะได้ `splitColumnOfMatrix(a,0) = { }` โดยกำหนดให้เมตริกซ์ทุกแถวมีจำนวนหลักที่เท่ากัน (10 คะแนน)

Take Note