อาเรย์ 2 มิติ

หัวข้อ

- การสร้างและการตั้งค่าเริ่มต้น
- อาเรย์กับวงวน for
- การประมวลผลเมทริกซ์
- การประมวลผลภาพ

อาเรย์หลายมิติ

```
int[] a = new int[4];
a[2] = 3;
```
หนึ่งมิติ

	0	1	2	3
a	0	0	3	0

```
int[][] b = new int[3][4];
b[1][3] = 4;
```
สองมิติ

b	0	1	2	3
0	0	0	0	0
1	0	0	0	4
2	0	0	0	0

```
int[][][] c = new int[2][3][4];
c[0][1][2] = 9;
```
สามมิติ

c				
	0	0	0	0
0	0	0	0	0
0	0	0	9	0
0	0	0	0	0

การตั้งค่าเริ่ม

```
int[] a = { 1, 2, 3, 4 };
```

	0	1	2	3
a	1	2	3	4

```
int[][] b = { {1,  2,  3,  4},
              {5,  6,  7,  8},
              {9, 10, 11, 12} };
```

b	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

อาเรย์ กับ วงวน for

```
public static void print(int[] d) {
  for (int i = 0; i < d.length; i++) {
    System.out.print(d[i] + " ");
  }
  System.out.println();
}
```

```
public static void print(int[][] d) {
  for (int i = 0; i < d.length; i++) {
    for (int j = 0; j < d[i].length; j++) {
      System.out.print(d[i][j] + " ");
    }
    System.out.println();
  }
}
```

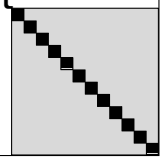
จำนวนแถวแนวนอน

จำนวนแถวแนวตั้ง

d เป็นอาเรย์ **2** มิติ
d.length แทนจำนวนแถวแนวนอนของ **d**
d[i].length แทนจำนวนช่องของอาเรย์แถวที่ **i**
คิดแบบง่าย : **d.length** จำนวนแถว, **d[0].length** จำนวนคอลัมน์
(ขนาดของอาเรย์ในแต่ละแถวไม่เท่ากันก็ได้ แต่เราจะไม่พูดถึง)

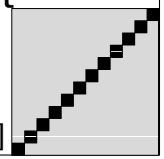
หาผลรวมของช่องในแนวทแยงมุมของอาเรย์ 2 มิติ

```
public static int sumDiag1(int[][] d) {
  int sum = 0;
  for (int i = 0; i < d.length; i++) {
    sum = sum + d[i][i];
  }
  return sum;
}
```



d[0][0] + d[1][1] + ... + d[11][11]

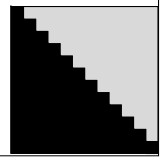
```
public static int sumDiag1(int[][] d) {
  int sum = 0;
  for (int i = 0; i < d.length; i++) {
    sum = sum + d[i][d.length-i-1];
  }
  return sum;
}
```



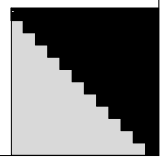
d[0][11] + d[1][10] + ... + d[11][0]

หาผลรวมของสามเหลี่ยมล่างและบนของอาเรย์ 2 มิติ

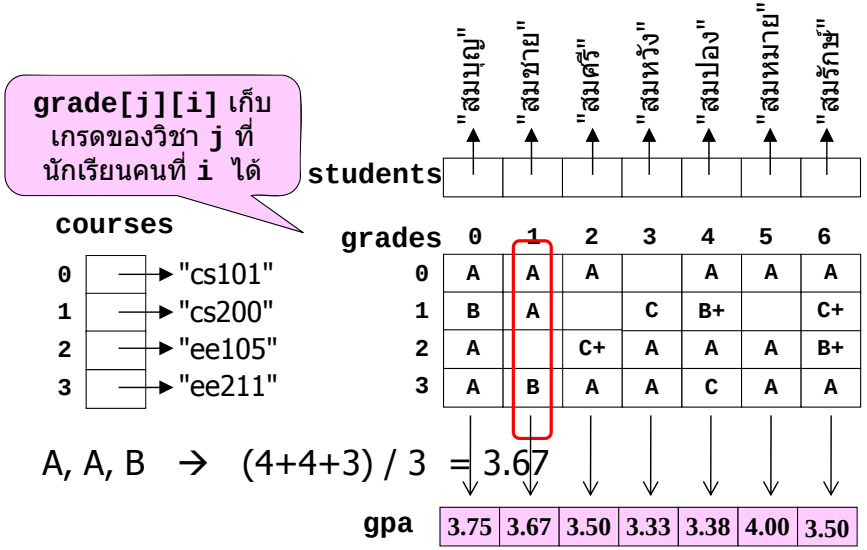
```
public static int sumLower(int[][] d) {
  int sum = 0;
  for (int i = 0; i < d.length; i++) {
    for (int j = 0; j <= i; j++) {
      sum = sum + d[i][j];
    }
  }
  return sum;
}
```



```
public static int sumUpper(int[][] d) {
  int sum = 0;
  for (int i = 0; i < d.length; i++) {
    for (int j = i; j < d.length; j++) {
      sum = sum + d[i][j];
    }
  }
  return sum;
}
```



ตัวอย่าง : หา GPA ของนักเรียน



เมท็อดคำนวณ GPA ของนักเรียนทุกคน

```
public static double[] studentGPA(String[][] grades) {
  double[] gpa = new double[grades[0].length];
  for (int i = 0; i < grades[0].length; i++) {
    gpa[i] = studentGPA(grades, i);
  }
  return gpa;
}
```

คำนวณ GPA ของนักเรียนคนที่ **i**

```
public static double studentGPA(
                        String[][] grades, int i) {
  double sum = 0;
  int n = 0;
  for (int j = 0; j < grades.length; j++) {
    if (!grades[j][i].equals("")) {
      sum = sum + grade2point(grades[j][i]);
      n++;
    }
  }
  return sum / n;
}
```

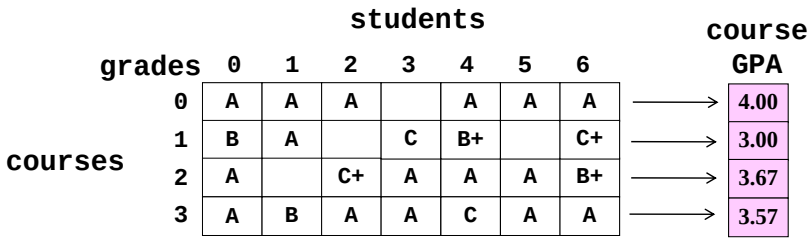
หาคะแนนของวิชา **j** นักเรียนคนที่ **i**

เมท็อดเปลี่ยนเกรดเป็นคะแนน

```
public static double grade2point(String g) {
  if (g.equals("A")) return 4.0;
  else if (g.equals("B+")) return 3.5;
  else if (g.equals("B" )) return 3.0;
  else if (g.equals("C+")) return 2.5;
  else if (g.equals("C" )) return 2.0;
  else if (g.equals("D+")) return 1.5;
  else if (g.equals("D" )) return 1.0;
  else return 0.0;
}
```

```
public static double grade2point(String g) {
  String[] grades = { "A", "B+", "B", "C+",
                      "C", "D+", "D", "F" };
  double[] points = { 4, 3.5, 3, 2.5, 2, 1.5, 1, 0 };
  for (int k = 0; k < grades.length; k++) {
    if (grades[k].equals(g)) return points[k];
  }
  return 0.0;
}
```


ลองเขียนเอง : หา GPA ของวิชา



ลองทำดู

จงเขียนเมท็อด

- **void printEqualRows(int[][] d)**
  แสดงแถวใน d ที่มีค่าเหมือนกัน

1	2	4	3
9	6	5	7
1	2	4	3

- **double saddlePoint(double[][] d)**
  คืนข้อมูลใน **d** ที่น้อยสุดในแถว แต่มากสุดในคอลัมน์, ถ้าไม่มีคืน **-1**

1	2	4	3
9	6	5	7
1	2	4	3

การบวกเมทริกซ์

$$\begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2+0 & 3+1 & 4+0 \\ 5+1 & 6+0 & 7+1 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 4 \\ 6 & 6 & 8 \end{bmatrix}$$

a **b** **c**

c[i][j] = a[i][j] + b[i][j]

```
public static double[][] add(double[][] a, double[][] b) {
  int row = a.length, col = a[0].length;
  double[][] c = new double[row][col];
  for (int i = 0; i < row; i++) {
    for (int j = 0; j < col; j++) {
      c[i][j] = a[i][j] + b[i][j];
    }
  }
  return c;
}
```


การคูณเมทริกซ์กับจำนวนจริง

$$0.5\begin{bmatrix} 2 & 4 & 6 \\ 8 & 2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 2 \end{bmatrix}$$

s **a** **c**

c[i][j] = s*a[i][j]

```
public static double[][] add(double[][] a, double s) {
  int row = a.length, col = a[0].length;
  double[][] c = new double[row][col];
  for (int i = 0; i < row; i++) {
    for (int j = 0; j < col; j++) {
      c[i][j] = s * a[i][j];
    }
  }
  return c;
}
```


การคูณเวกเตอร์กับเมทริกซ์

$$\begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} \\ b_{1,0} & b_{1,1} & b_{1,2} \end{bmatrix}$$

$$\begin{bmatrix} a_0 b_{0,0} + a_1 b_{1,0} , & a_0 b_{0,1} + a_1 b_{1,1} , & a_0 b_{0,2} + a_1 b_{1,2} \end{bmatrix}$$

$$c_i = \sum_{k=0}^{n-1} a_k b_{k,i}$$

```
public static double[] multiply(double[] a, double[][] b) {
  int row = b.length, col = b[0].length;
  double[] c = new double[col];
  for (int i = 0; i < col; i++) {
    for (int k = 0; k < row; k++) {
      c[i] = c[i] + a[k] * b[k][i];
    }
  }
  return c;
}
```


การคูณเมทริกซ์กับเมทริกซ์

$$\begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{bmatrix} \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} \\ b_{1,0} & b_{1,1} & b_{1,2} \end{bmatrix}$$

$$\begin{bmatrix} a_{0,0}b_{0,0} + a_{0,1}b_{1,0} & a_{0,0}b_{0,1} + a_{0,1}b_{1,1} & a_{0,0}b_{0,2} + a_{0,1}b_{1,2} \\ a_{1,0}b_{0,0} + a_{1,1}b_{1,0} & a_{1,0}b_{0,1} + a_{1,1}b_{1,1} & a_{1,0}b_{0,2} + a_{1,1}b_{1,2} \end{bmatrix}$$

$$c_{ij} = \sum_{k=0}^{n-1} a_{i,k} b_{k,j}$$

```
for (int k = 0; k < n; k++) {
  c[i][j] = c[i][j] + a[i][k] * b[k][j];
}
```

การคูณเมทริกซ์กับเมทริกซ์

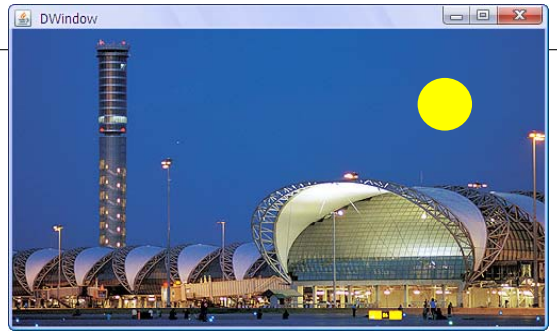
$$\begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{bmatrix}\begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} \\ b_{1,0} & b_{1,1} & b_{1,2} \end{bmatrix} = \begin{bmatrix} a_{0,0}b_{0,0}+a_{0,1}b_{1,0} & a_{0,0}b_{0,1}+a_{0,1}b_{1,1} & a_{0,0}b_{0,2}+a_{0,1}b_{1,2} \\ a_{1,0}b_{0,0}+a_{1,1}b_{1,0} & a_{1,0}b_{0,1}+a_{1,1}b_{1,1} & a_{1,0}b_{0,2}+a_{1,1}b_{1,2} \end{bmatrix}$$

$$= \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} \\ c_{1,0} & c_{1,1} & c_{1,2} \end{bmatrix} \qquad c_{i,j} = \sum_{k=0}^{n-1} a_{i,k}b_{k,j}$$

```
public static double[][] multiply(double[][] a, double[][] b) {
  int row = a.length, col = b[0].length;
  int[][] c = new int[row][col];
  int n = b.length;
  for (int i = 0; i < row; i++) {
    for (int j = 0; j < col; j++) {
      for (int k = 0; k < n; k++) {
        c[i][j] = c[i][j] + a[i][k] * b[k][j];
      }
    }
  }
  return c;
}
```

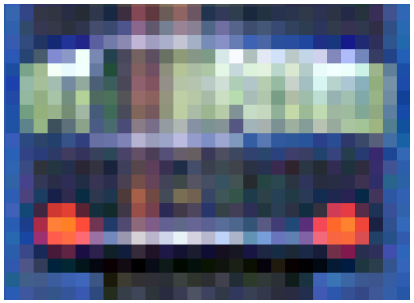

การอ่านแฟ้มภาพ

```
import jlab.graphics.DWindow;

public class ImageDisplay {
  public static void main(String[] args) {
    DWindow w = new DWindow();
    w.loadImage("c:/java101/airport.jpg");
    w.fillEllipse(DWindow.YELLOW, 400, 70, 40, 40);
  }
}
```



แผนที่จุดภาพ (pixmap)

`p[x][y]`



`int[500][279]`

หนึ่งจุด หนึ่งสี
หนึ่งสี หนึ่ง **int**

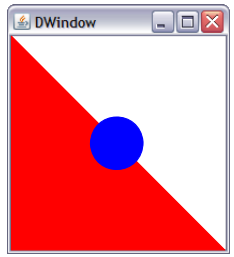
การอ่านและการตั้งแผนที่จุดภาพ

```
DWindow w = new DWindow();
w.loadImage("c:/java101/airport.jpg");
w.fillEllipse(DWindow.YELLOW, 400, 70, 40, 40);

int[][] p = w.getPixmap(); // คืนแผนที่จุดภาพ

...

w.setPixmap(p);  // ตั้งแผนที่จุดภาพ
```

การพลิกภาพ (ซ้าย ↔ ขวา)

```
DWindow w = new DWindow();
w.loadImage("c:/java101/airport.jpg");
int[][] p = w.getPixmap();
int width = p.length, height = p[0].length;
for (int x = 0; x < width/2; x++) {
  for (int y = 0; y < height; y++) {
    int t = p[x][y];
    p[x][y] = p[width - x - 1][y];
    p[width - x - 1][y] = t;
  }
}
w.setPixmap(p);
```

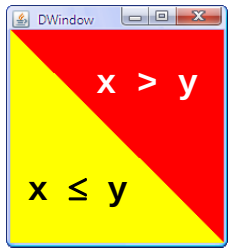

สีมาตรฐานใน DWindow



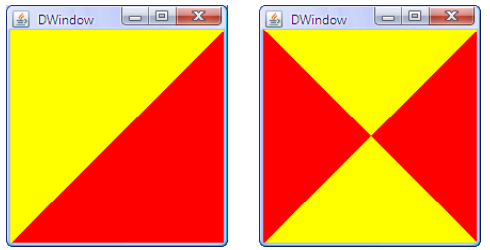

```
DWindow w = new DWindow(200, 200);
int[][] p = w.getPixmap();
for (int x = 0; x < p.length; x++) {
  for (int y = x; y < p[0].length; y++) {
    p[x][y] = DWindow.RED;
  }
}
w.setPixmap(p);
w.fillEllipse(DWindow.BLUE, 100, 100, 50, 50);
```


getPixmap, setPixmap

```
import jlab.graphics.DWindow;

public class FillColor {
  public static void main(String[] args) {
    DWindow w = new DWindow(200, 200);
    int[][] p = w.getPixmap();
    for (int x = 0; x < p.length; x++) {
      for (int y = 0; y < p[0].length; y++) {
        if (x > y)
          p[x][y] = DWindow.RED;
        else
          p[x][y] = DWindow.YELLOW;
      }
    }
    w.setPixmap(p);
  }
}
```



ลองเขียนดู



การผสมและการแยกสี

```
DWindow w = new DWindow(200, 200);
w.fillEllipse(DWindow.ORANGE, 100, 100, 20, 20);
                              R   G   B
int color = DWindow.mixRGB(255, 128, 0);
w.fillEllipse(color, 50, 50, 20, 20);
int r = DWindow.getR(color);
int g = DWindow.getG(color);
int b = DWindow.getB(color);
```

แม่สีมีค่าตั้งแต่ 0 -255

การทำภาพเนกาทิฟ

```
int[][] p = w.getPixmap();
. . .
int r = DWindow.getR(p[x][y]);
int g = DWindow.getG(p[x][y]);
int b = DWindow.getB(p[x][y]);
r = 255 - r;
g = 255 - g;
b = 255 - b;
p[x][y] = DWindow.mixRGB(r, g, b);
```


เมท็อดทำภาพเนกาทิฟ

```
public static void main(String[] args) {
  DWindow w = new DWindow(200, 200);
  w.loadImage("c:/java101/airport.jpg");
  int[][] p = w.getPixmap();
  w.setPixmap(negative(p));
}
public static int[][] negative(int[][] p) {
  for (int x = 0; x < p.length; x++) {
    for (int y = 0; y < p[0].length; y++) {
      int r = DWindow.getR(p[x][y]);
      int g = DWindow.getG(p[x][y]);
      int b = DWindow.getB(p[x][y]);
      r = 255 – r;  g = 255 – g;  b = 255 - b;
      p[x][y] = DWindow.mixRGB(r, g, b);
    }
  }
  return p;
}
```


การทำภาพขาวดำ

```
int[][] p = w.getPixmap();
. . .
int r = DWindow.getR(p[x][y]);
int g = DWindow.getG(p[x][y]);
int b = DWindow.getB(p[x][y]);
int gray = (r + g + b) / 3;
p[x][y] = DWindow.mixRGB(gray, gray, gray);
```

เมท็อดทำภาพขาวดำ

```
public static void main(String[] args) {
  DWindow w = new DWindow(200, 200);
  w.loadImage("c:/java101/airport.jpg");
  int[][] p = w.getPixmap();
  w.setPixmap(gray(p));
}
public static int[][] gray(int[][] p) {
  for (int x = 0; x < p.length; x++) {
    for (int y = 0; y < p[0].length; y++) {
      int r = DWindow.getR(p[x][y];);
      int g = DWindow.getG(p[x][y];);
      int b = DWindow.getB(p[x][y];);
      int gray = (r + g + b) / 3;
      p[x][y] = DWindow.mixRGB(gray, gray, gray);
    }
  }
  return p;
}
```