

บทที่ 8

Arrays

อาเรย์

อาร์เรย์(Arrays)

- **array** คือ ประเภทของข้อมูลที่สามารถเก็บข้อมูลประเภทเดียวกันแบบเป็นลำดับได้หลายค่า โดยข้อมูลนั้นจะอยู่ในตัวแปรตัวเดียวกันที่เรียกว่า ตัวแปร อาร์เรย์ และใช้ index ในการเข้าถึงข้อมูล
- เช่น การประกาศตัวแปร 100 ตัวแปร เพื่อเก็บข้อมูล 100 ค่าคงเป็นเรื่องที่ลำบาก ดังนั้นการใช้ตัวแปรชนิด array ในการจัดเก็บข้อมูลจึงสะดวกกว่า

ประเภทของอาร์เรย์(Arrays) ใน Java

array ในภาษา Java แบ่งตามประเภทได้ 2 ประเภท คือ

1. อาร์เรย์ของข้อมูลชนิดพื้นฐาน(Primitive type array)
2. อาร์เรย์ชนิดอ้างอิง(Reference type array) หรือ อาร์เรย์ของคลาส

อาร์เรย์ของชนิดข้อมูลพื้นฐาน

- รูปแบบการประกาศ

แบบที่ 1

```
data_type[] array_name;  
array_name = new data_type[size];
```

แบบที่ 2

```
data_type[] array_name = new data_type[size];
```

แบบที่ 3

```
data_type[] array_name = {value,...,...}; //กำหนดค่าเริ่มต้น
```

data_type : ชนิดข้อมูลของตัวแปรอาร์เรย์ เช่น int, double เป็นต้น

array_name : ชื่อตัวแปรอาร์เรย์

size : ขนาดของตัวแปร

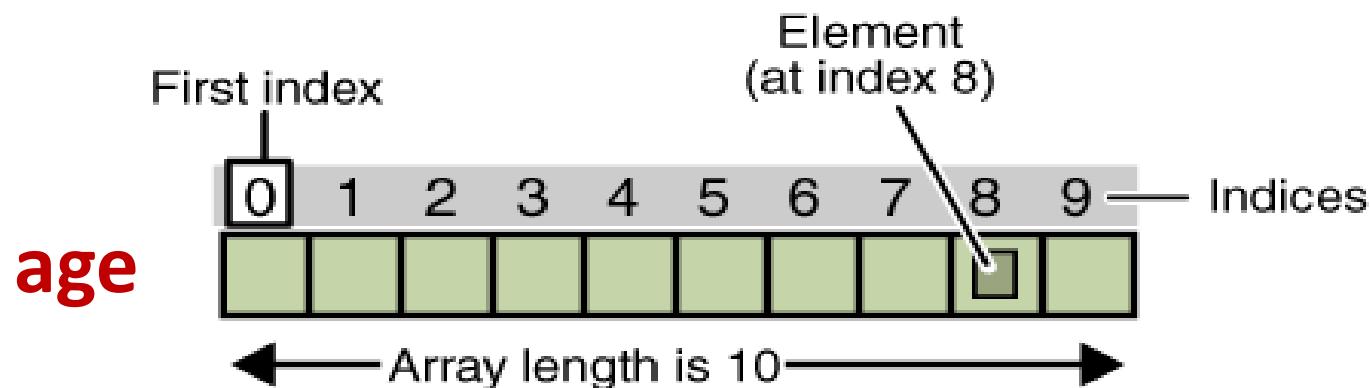
ตัวอย่างการประกาศอาร์เรย์

- ตัวอย่างการประกาศอาร์เรย์ :

แบบที่ 1: int [] age = new int[10]; หรือ

แบบที่ 2: int age [] = new int[10];

เป็นการประกาศตัวแปรอเรย์ชื่อ age ใช้เก็บเลขจำนวนเต็ม 10 ค่า โดยตัวแปร age ตำแหน่งที่ 1 มีตัวซึ่งเป็น 0



ตัวอย่างการประกาศอาร์เรย์

ตัวอย่างการประกาศอาร์เรย์:

แบบที่ 1: double [] data; data = new double[10] หรือ

แบบที่ 2: double [] data = new double[10]; หรือ

แบบที่ 3: double data[] = new double[10];

data[0]	data[1]	data[2]	data[3]	data[4]	data[5]	data[6]	data[7]	data[8]	data[9]

การเพิ่มข้อมูลเข้าไปในอะเรย์ data ตำแหน่งที่ 3 ให้มีค่าเท่ากับ 29.95

ตำแหน่งที่ 5 ให้มีค่า 2.19

เขียนได้โดย

data[2] = 29.95;

data[4] = 2.19;

		29.95		2.19					
data[0]	data[1]	data[2]	data[3]	data[4]	data[5]	data[6]	data[7]	data[8]	data[9]

ตัวอย่างการประกาศอาร์เรย์

12.11	44.00	29.95	55.65	2.19					
data[0]	data[1]	data[2]	data[3]	data[4]	data[5]	data[6]	data[7]	data[8]	data[9]

- การอ่านค่าข้อมูลที่จัดเก็บอยู่ในอะเรย์

```
double a = data[3]; // อ่านค่าอะเรย์ data ในตำแหน่งที่ 4 (55.65) มาเก็บลงตัวแปร a
```

```
System.out.println("Value " + data[4]); // พิมพ์ค่าอะเรย์ data ตำแหน่งที่ 5 (2.19)
```

- การอ่านค่าความยาวของอะเรย์

```
int len = data.length; // เป็นการหาความยาวของอะเรย์ data (ในที่นี้มีค่าเป็น  
10) และเก็บลงตัวแปร len
```

➤ ตัวชี้อะเรย์ เริ่มจาก 0 ถึง length - 1

➤ อะเรย์มีความยาวจำกัด การเข้าถึงสมาชิกที่ไม่มีอยู่ในอะเรย์ (เช่น data[10]) จะทำให้เกิด **error out of bound**

ตัวอย่างการประกาศอาร์เรย์

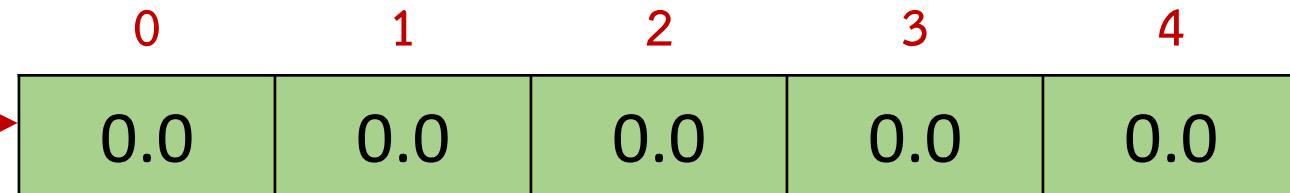
double[] rainfall; หรือ double rainfall[];
rainfall



rainfall = new double[5];

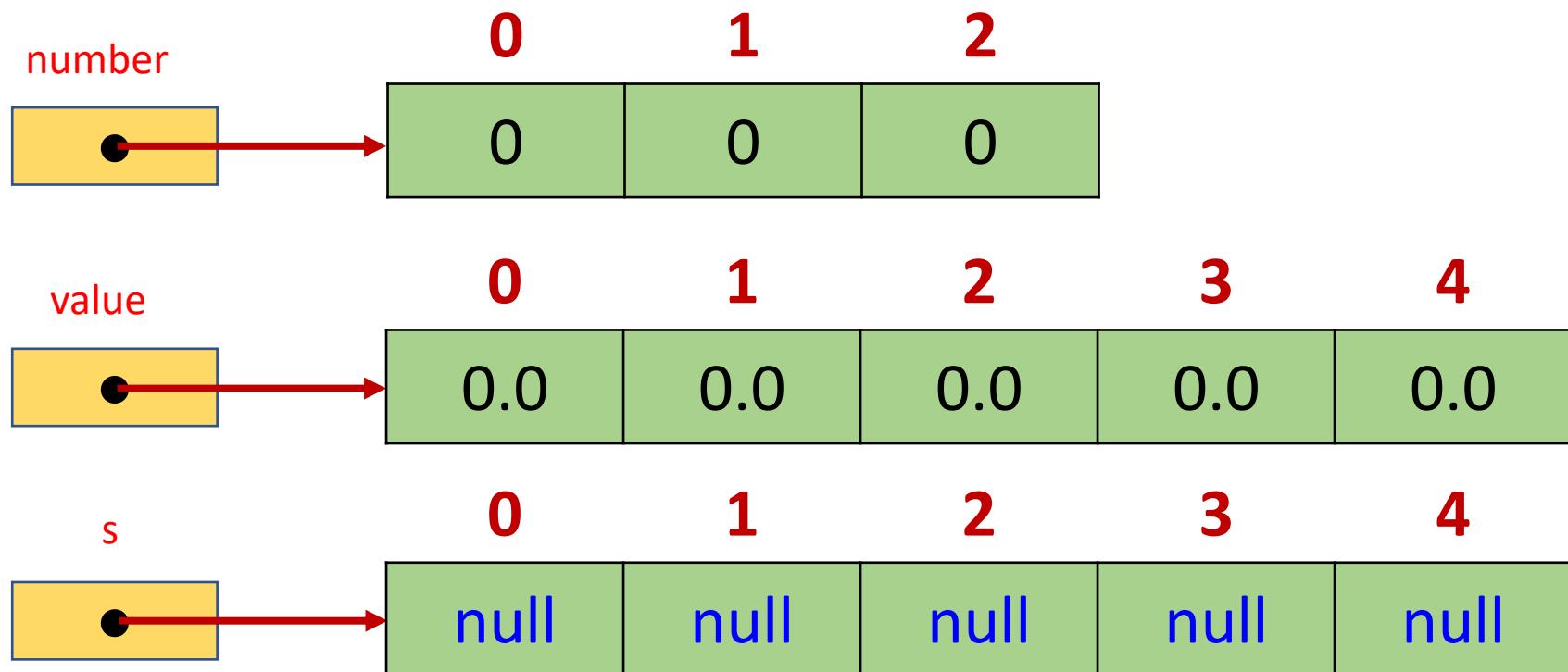
หรือ

double[] rainfall = new double[5];
rainfall



ตัวอย่างการประกาศอาร์เรย์

```
int number[] = new int[3];
double value[] = new double[5];
String s[] = new String[5];
```



การเข้าถึงตัวแปรอาร์เรย์

- ตัวอย่างการกำหนดค่าให้อาร์เรย์

```
double value[] = new double[5];
```

```
value[3] = 50.25;
```

```
value[1] = 30.35;
```

```
value[4] = 15.00;
```

```
value[0] = value[4] * 2;
```

```
value[2] = value[3] - value[1];
```

0	1	2	3	4
0.0	0.0	0.0	0.0	0.0
0	1	2	3	4
0.0	0.0	0.0	50.25	0.0
0	1	2	3	4
0.0	30.35	0.0	50.25	0.0
0	1	2	3	4
30.00	30.35	0.0	50.25	15.00
0	1	2	3	4
30.00	30.35	19.9	50.25	15.00

การกำหนดค่าให้ตัวแปรอาร์เรย์

- ตัวอย่าง (**การเขียนแบบทั่วๆไป**)

```
class ArrayDemo
{
    public static void main(String[] args)
    {
        int [ ] anArray;
        anArray = new int[5];
        anArray[0] = 10;
        anArray[1] = 20;
        anArray[2] = 30;
        anArray[3] = 40;
        anArray[4] = 50;
        System.out.println("index 0: " + anArray[0]);
        System.out.println("index 1: " + anArray[1]);
        System.out.println("index 2: " + anArray[2]);
        System.out.println("index 3: " + anArray[3]);
        System.out.println("index 4: " + anArray[4]);
    }
}
```

ผลลัพธ์ :

```
index 0: 10
index 1: 20
index 2: 30
index 3: 40
index 4: 50
```

การกำหนดค่าให้ตัวแปรอาร์เรย์

- การเลือกอื่นๆ ในการสร้างอะเรย์คือการกำหนดค่าเริ่มต้น

```
int [ ] anArray = { 10, 20, 30, 40, 50};
```

```
class ArrayDemo
{
    public static void main(String[] args)
    {
        int[ ] anArray = { 10, 20, 30, 40, 50 };
        System.out.println("index 0: " + anArray[0]);
        System.out.println("index 1: " + anArray[1]);
        System.out.println("index 2: " + anArray[2]);
        System.out.println("index 3: " + anArray[3]);
        System.out.println("index 4: " + anArray[4]);
    }
}
```

ผลลัพธ์ :

```
index 0: 10
index 1: 20
index 2: 30
index 3: 40
index 4: 50
```

การกำหนดค่าให้ตัวแปรอาร์เรย์

- ตัวอย่าง (การเขียนแบบใช้ loop)

```
class ArrayDemo
{
    public static void main(String[] args)
    {
        int [] anArray;
        anArray = new int[5];

        for (int i=0; i<5; i++)
            anArray[i] = (i+1) * 10;

        for (int i=0; i<5; i++)
            System.out.println("index " + i + ": " + anArray[i]);
    }
}
```

ผลลัพธ์ :

```
index 0: 10
index 1: 20
index 2: 30
index 3: 40
index 4: 50
```

การเข้าถึงตัวแปรอาร์เรย์

- ตัวอย่างการแสดงค่าอาร์เรย์ในกรณียังไม่ได้กำหนดข้อมูล

```
public class Array_Ex1 {  
    public static void main(String args[]) {  
  
        double value[] = new double[5]; //declare and create array  
  
        System.out.printf("%s %10s\n" , "Index" , "Value"); //column heading  
  
        //loop for output each array elements  
        for(int i=0; i< value.length; i++)  
        {  
            System.out.printf("%5d %10.2f\n" , i , value[i]);  
        }  
    }  
}
```

ผลลัพธ์	Index	Value
	0	0.00
	1	0.00
	2	0.00
	3	0.00
	4	0.00

การเข้าถึงตัวแปรอาร์เรย์

- ตัวอย่างการแสดงค่าอาร์เรย์และค่าในอาร์เรย์

```
public class Array_Ex2 {  
    public static void main(String args[]) {  
  
        //declare and initiate value  
        int number[] = {32, 25, 64, 95, 14, 90, 55, 60, 42, 37};  
  
        System.out.printf("%s %10s\n", "Index", "Value"); //column heading  
  
        //loop for output each array elements  
        for(int i=0; i< number.length; i++)  
        {  
            System.out.printf("%5d %10d\n", i, number[i]);  
        }  
    }  
}
```

ผลลัพธ์	
Index	Value
0	32
1	25
2	64
3	95
4	14
5	90
6	55
7	60
8	42
9	37

ตัวอย่างโปรแกรมรับค่าตัวเลขผู้ใช้จำนวน 5 ตัวเลขใส่ อาร์เรย์ ให้แสดงผลรวมและค่าเฉลี่ย

```
import java.util.Scanner;

public class Array_Ex3 {
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);

        int number[] = new int[5];
        int sum = 0;
        double average;

        for(int i = 0; i < number.length; i++) {
            System.out.print("Enter integer number ==> ");
            number[i] = input.nextInt();
            sum += number[i];
        }

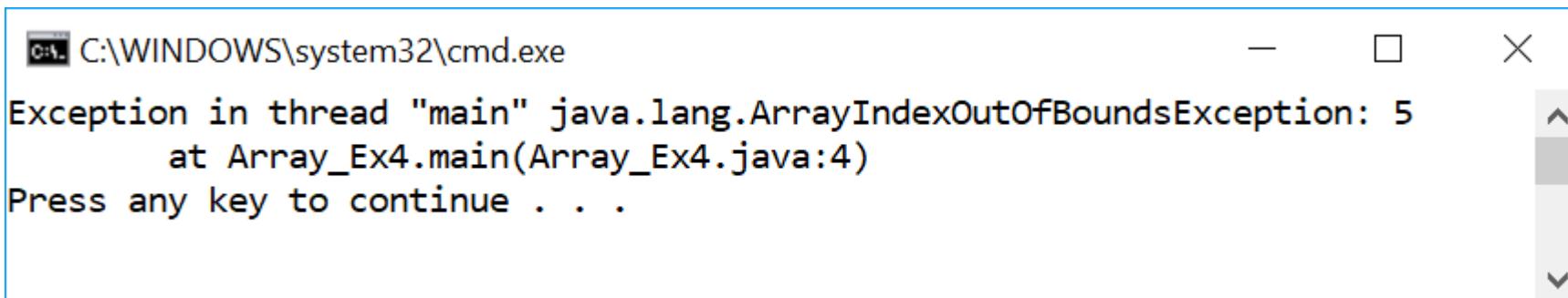
        average = (double)sum / number.length;

        System.out.println("-----");
        System.out.println("Sum of all number = " + sum);
        System.out.println("Average value = " + average);
    }
}
```

การอ้างอิงอาร์เรย์เกินขนาด

```
class Array_Ex4 {  
    public static void main(String[] args) {  
        int number[] = new int[5];  
        number[5] = 10;  
        System.out.println("number[5] : " + number[5]);  
    }  
}
```

ผลลัพธ์ขณะรันโปรแกรม



The screenshot shows a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The window contains the following text:
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
at Array_Ex4.main(Array_Ex4.java:4)
Press any key to continue . . .

อาร์เรย์ของคลาส(Array of Class)

- อาร์เรย์ของคลาส คือ การเก็บวัตถุ(**object**) ของคลาสในอาร์เรย์
- รูปแบบการประกาศ

```
Class_name [] array_name = new Data_type[size];
```

หรือ

```
Class_name array_name [] = new Data_type[size];
```

Class_name : ชนิดข้อมูลแบบคลาสของตัวแปรอาร์เรย์ เช่น Book, Student, Employee เป็นต้น

array_name : ชื่อตัวแปรอาร์เรย์

size : ขนาดของตัวแปร

อาร์เรย์ของคลาส(Array of Class)

- ตัวอย่าง

```
Employee [] emp = new Employee[5];
```

หรือ

```
Employee emp[] = new Employee[5];
```

ไฟล์ Employee.java สร้างคลาส Employee

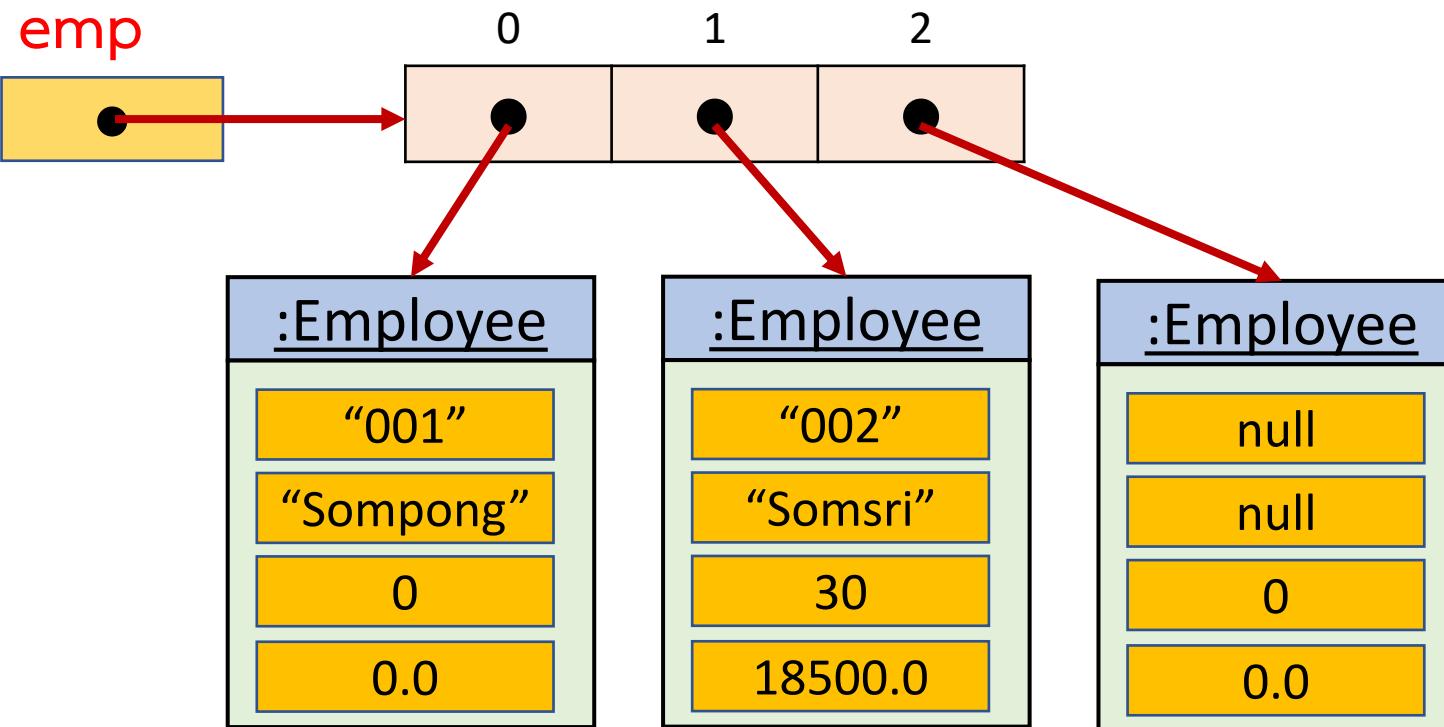
```
public class Employee {  
    //member  
    private String id;  
    private String name;  
    private int age;  
    private double salary;  
  
    //constructor  
    public Employee (){ }  
  
    public Employee (String id, String name) {  
        this.id = id;  
        this.name = name;  
    }  
  
    public Employee (String id, String name , int age , double salary) {  
        this.id = id;  
        this.name = name;  
        this.age = age;  
        this.salary = salary;  
    }  
}
```

```
Employee emp [ ] = new Employee [ 3 ] ;  
emp
```

0 1 2



```
emp [ 0 ] = new Employee ( "001" , "Sompong" ) ;  
emp [ 1 ] = new Employee ( "002" , "Somsri" , 30 , 18500 ) ;  
emp [ 2 ] = new Employee ( ) ;
```



ไฟล์ Employee.java เพิ่มเมธอดset/getในการกำหนดและอ่านค่าให้ตัวแปร

```
//method set
public void setID(String id) { this.id = id; }

public void setName(String name) { this.name = name; }

public void setAge(int age) { this.age = age; }

public void setSalary(double salary) { this.salary = salary; }

//method get
public String getID() { return id; }

public String getName() { return name; }

public int getAge() { return age; }

public double getSalary() { return salary; }
```

ไฟล์ Employee.java เพิ่มメธอดสำหรับแสดงข้อมูลตัวแปร

```
//other methods
public void getEmployeeInfo() {
    System.out.printf("%-10s : %s\n" , "id" , id);
    System.out.printf("%-10s : %s\n" , "name" , name);
    System.out.printf("%-10s : %d\n" , "age" , age);
    System.out.printf("%-10s : %.2f\n" , "salary" , salary);
    System.out.println("-----");
}
```

ไฟล์ ArrayEmployee.java ทดสอบสร้างอาร์เรย์เก็บวัตถุEmployee

```
public class Array_Ex5 {
    public static void main(String[] args) {
        Employee emp[] = new Employee[3];

        emp[0] = new Employee("001", "Sompong");
        emp[1] = new Employee("002", "Somsri", 30, 18500);
        emp[2] = new Employee();

        emp[2].setID("003");
        emp[2].setName("Vanida");
        emp[2].setAge(25);
        emp[2].setSalary(15000);

        emp[0].getEmployeeInfo();
        emp[1].getEmployeeInfo();
        emp[2].getEmployeeInfo();
    }
}
```

คลาส Arrays

- คลาส Array อยู่ในแพกเกจ `java.util.Arrays`
- เมธอดที่น่าสนใจในคลาส Arrays ได้แก่
 - `sort()` – จัดเรียงข้อมูลในอาร์เรย์
 - `binarySearch()` –ค้นหาข้อมูลในอาร์เรย์ ถ้าพบจะส่งค่า `index` กลับ ถ้าไม่พบจะส่งค่า `null` กลับ
 - `toString()` --แปลงข้อมูลในอาร์เรย์ให้เป็นข้อความแสดงใน `[]` คันແຕ່ລະຄ່າດ້ວຍເຄື່ອງໝາຍ , (comma)
 - `equal()` –เปรียบเทียบค่าของอาร์เรย์ว่าเท่ากันหรือไม่ ถ้าเท่ากันจะส่งค่า `true` กลับ ถ้าไม่เท่ากันจะส่งค่า `false` กลับ

sort()

- sort() ใช้เพื่อเรียงลำดับข้อมูลในอาร์เรย์



```
int [ ] scores = {51, 3, 81, 20, 14};  
Arrays.sort(scores);
```

// ผลที่ได้ จะเรียกมีการเรียงข้อมูล
ในอาร์เรย์ score จากน้อยไปมาก

```
System.out.println("Minimum no.: " + scores[0]); // แสดงค่าต่ำสุดคือ 3  
System.out.println("Maximum no.: " + scores[scores.length - 1]); // แสดงค่าสูงสุดคือ 81
```

ไฟล์ TestArraySort.java ทดสอบการใช้งานเมธอด sort()

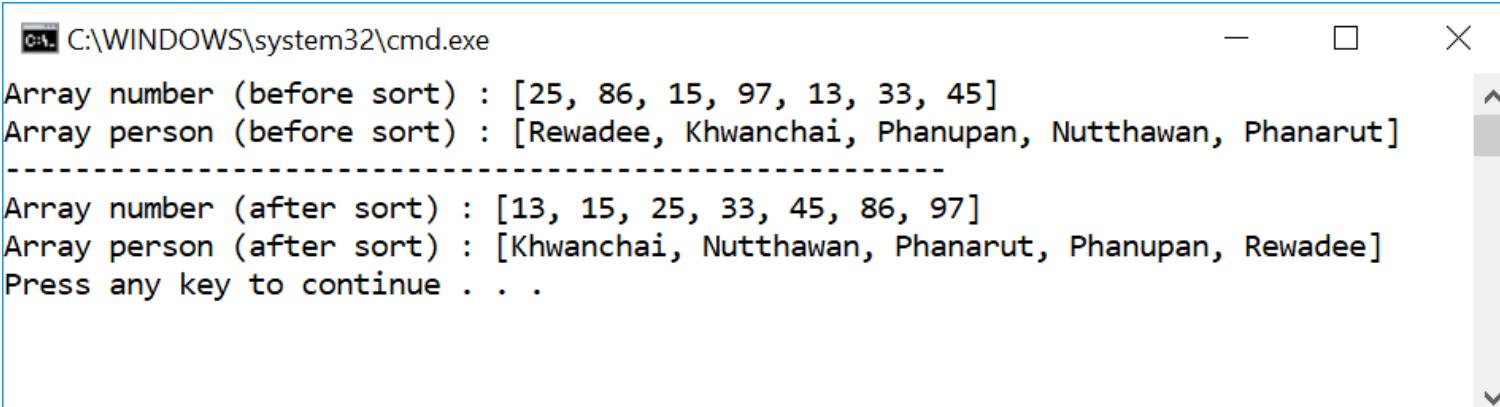
```
import java.util.Arrays;

public class TestArraySort {
    public static void main(String[] args) {
        int number[] = {25, 86, 15, 97, 13, 33, 45};
        String person[] = {"Rewadee", "Khwanchai", "Phanupan", "Nutthawan", "Phanarut"};
        System.out.println("Array number (before sort) : " + Arrays.toString(number));
        System.out.println("Array person (before sort) : " + Arrays.toString(person));

        Arrays.sort(number);
        Arrays.sort(person);

        System.out.println("-----");
        System.out.println("Array number (after sort) : " + Arrays.toString(number));
        System.out.println("Array person (after sort) : " + Arrays.toString(person));
    }
}
```

ผลลัพธ์



```
C:\WINDOWS\system32\cmd.exe
Array number (before sort) : [25, 86, 15, 97, 13, 33, 45]
Array person (before sort) : [Rewadee, Khwanchai, Phanupan, Nutthawan, Phanarut]
-----
Array number (after sort) : [13, 15, 25, 33, 45, 86, 97]
Array person (after sort) : [Khwanchai, Nutthawan, Phanarut, Phanupan, Rewadee]
Press any key to continue . . .
```

binarySearch()

- binarySearch() ใช้ค้นหาข้อมูลในอาร์เรย์
- ก่อนที่จะค้นหาเราต้องเรียงลำดับข้อมูลเสียก่อน
- ผลของการค้นหาคืออินเด็กซ์(ตำแหน่ง)ในอาร์เรย์
 - ✓ ถ้าพบ จะส่งตำแหน่งที่พบกลับมาให้
 - ✓ ถ้าไม่พบ จะส่งค่าติดลบมาให้

```
int [ ] id = {51, 3, 81, 20, 14};  
Arrays.sort(id);  
int index = Arrays.binarySearch(id, 51);  
System.out.println(index);
```

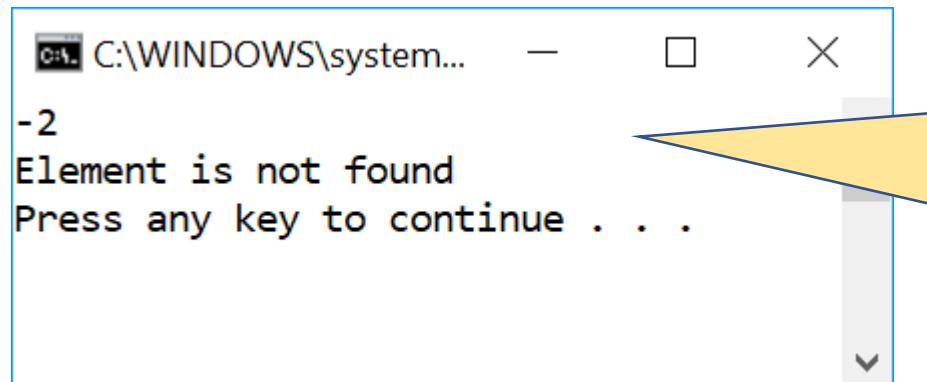
ไฟล์ TestArrayBinarySearch.java

ทดสอบการใช้งานเมธอด binarySearch()

```
import java.util.Arrays;

public class TestArrayBinarySearch {
    public static void main(String[] args) {
        int number[] = {25, 86, 15, 97, 13, 33, 45};
        int index = Arrays.binarySearch(number, 33);
        System.out.println(index);
        if(index >= 0)
            System.out.println("Element is found at the index : " + index);
        else
            System.out.println("Element is not found");
    }
}
```

ผลลัพธ์



ผลการทำงาน
ผิดพลาดหั้งที่มีตัวเลข
33 ในอาร์เรย์ แต่หา
ไม่พบ
**ต้องเรียงข้อมูลใน
อาร์เรย์ก่อน**

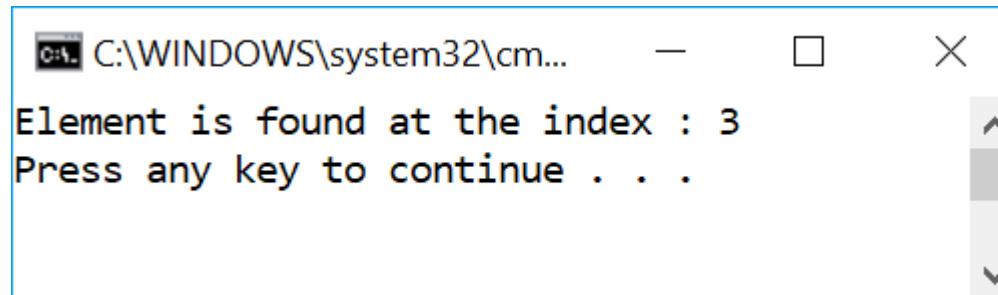
ไฟล์ TestArrayBinarySearch.java

ทดสอบการใช้งานเมธอด sort()

```
import java.util.Arrays;

public class TestArrayBinarySearch {
    public static void main(String[] args) {
        int number[] = {25, 86, 15, 97, 13, 33, 45};
        -----
        | Arrays.sort(number);
        | int index = Arrays.binarySearch(number , 33);
        -----
        if(index >= 0 )
            System.out.println("Element is found at the index : " + index);
        else
            System.out.println("Element is not found");
    }
}
```

ผลลัพธ์



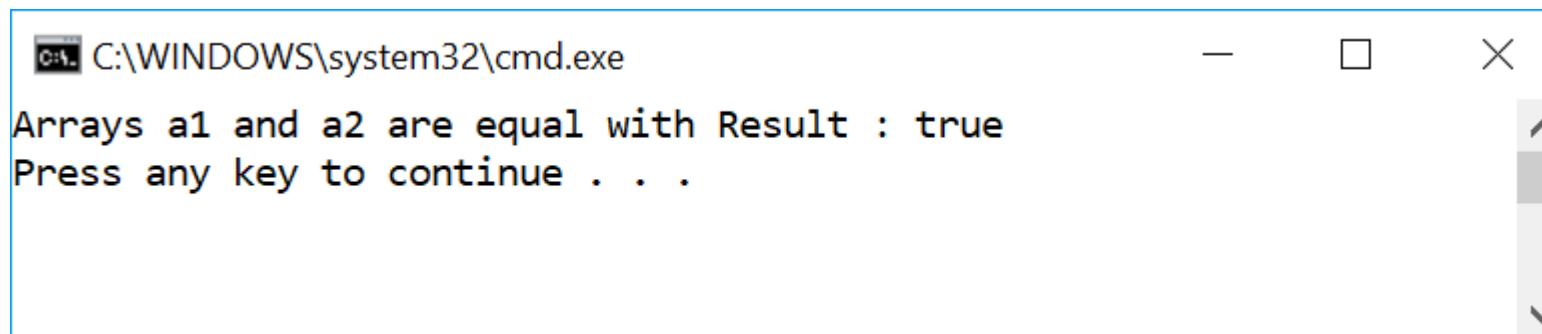
ไฟล์ TestArrayEqual.java ทดสอบการใช้งานเมธอด equals()

```
import java.util.Arrays;

public class TestArrayEqual {
    public static void main(String[] args) {
        int[] a1 = { 1, 2, 3 };
        int[] a2 = { 1, 2, 3 };

        boolean equal = Arrays.equals(a1, a2);
        if (equal)
            System.out.println("Arrays a1 and a2 are equal with Result : " + equal);
        else
            System.out.println("Arrays a1 and a2 are not equal with Result : " + equal);
    }
}
```

ผลลัพธ์



The screenshot shows a Windows Command Prompt window titled 'cmd C:\WINDOWS\system32\cmd.exe'. The window contains the following text:
Arrays a1 and a2 are equal with Result : true
Press any key to continue . . .

อาร์เรย์ 2 มิติ (Two-dimensional array)

- อาร์เรย์ 2 มิติ คือ อาร์เรย์ที่มีการจัดเก็บค่าแบบແລວและຄອລິມນໍ ກາພຄລ້າຍຕາຮາງ
- ມີຕັວໜີລຳດັບ(index) 2 ດຳ ຄື່ອ index ຂອງແລວ ແລະ index ຂອງຄອລິມນໍ

	0	1	2	3
0				
1				
2				

อาร์เรย์ 2 มิติ (Two-dimensional array)

- รูปแบบการประกาศ

```
Data_type [][] array_name = new Data_type[size][size];
```

หรือ

```
Data_type array_name [][] = new Data_type[size][size];
```

อาร์เรย์ 2 มิติ (Two-dimensional array)

- ตัวอย่างการประกาศ

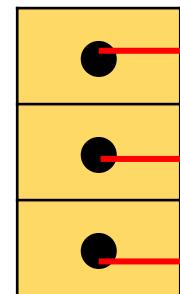
```
int score[][] = new int[3][4];
int number[][] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}};
String person[][] = {
    {"Rewadee", "Kunawut"}, ,
    {"Phanupan", "Khwanchai"}, ,
    {"Nutthawan", "Treerat"} ;
};
```

อาร์เรย์ 2 มิติ (Two-dimensional array)

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

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

number



(4)

1	2	3	4
---	---	---	---

(4)

5	6	7	8
---	---	---	---

(4)

9	10	11	12
---	----	----	----

อาร์เรย์ 2 มิติ (Two-dimensional array)

ตัวอย่างการกำหนดค่าให้อาร์เรย์ 2 มิติ

```
import java.util.Arrays;

public class Array2D {
    public static void main(String[] args) {

        int score[][] = new int[3][4];

        score[0][0] = 25;
        score[0][1] = 45;
        score[0][2] = 35;
        score[0][3] = 66;
        score[1][0] = 34;
        score[1][1] = 77;
        score[1][2] = 15;
        score[1][3] = 23;
        score[2][0] = 87;
        score[2][1] = 56;
        score[2][2] = 99;
        score[2][3] = 65;

        System.out.println(Arrays.deepToString(score));
    }
}
```

อาร์เรย์ 2 มิติ (Two-dimensional array)

- การหาขนาดของอาร์เรย์ 2 มิติ

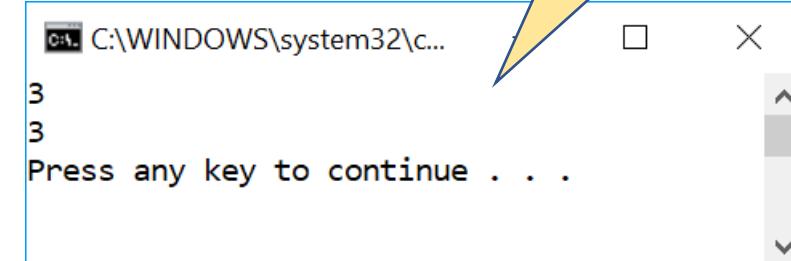
number r	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

```
public class Array2DLength {
    public static void main(String[] args) {

        int number[][] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}};
        String person[][] = {
            {"Rewadee", "Kunawut"}, 
            {"Phanupan", "Khwanchai"}, 
            {"Nutthawan", "Treerat"}
        };

        System.out.println(number.length);
        System.out.println(person.length);
    }
}
```

ผลลัพธ์



แสดงเฉพาะขนาด
ของແກວ

ไฟล์ Array2DLength.java วนซ้ำแสดงค่าในอาร์เรย์ 2 มิติ

```
public class Array2DLength {
    public static void main(String[] args) {

        int number[][] = {{1,2,3,4} , {5,6,7,8} , {9,10,11,12}};
        String person[][] = {
            {"Rewadee" , "Kunawut" } ,
            {"Phanupan" , "Khwanchai" },
            {"Nutthawan" , "Treerat" }
        };

        for(int i = 0; i < person.length; i++) {
            for(int j = 0; j < person[i].length; j++) {
                System.out.printf("%s \t" , person[i][j]);
            }
            System.out.println("\n-----");
        }
    }
}
```

ผลลัพธ์

Rewadee	Kunawut

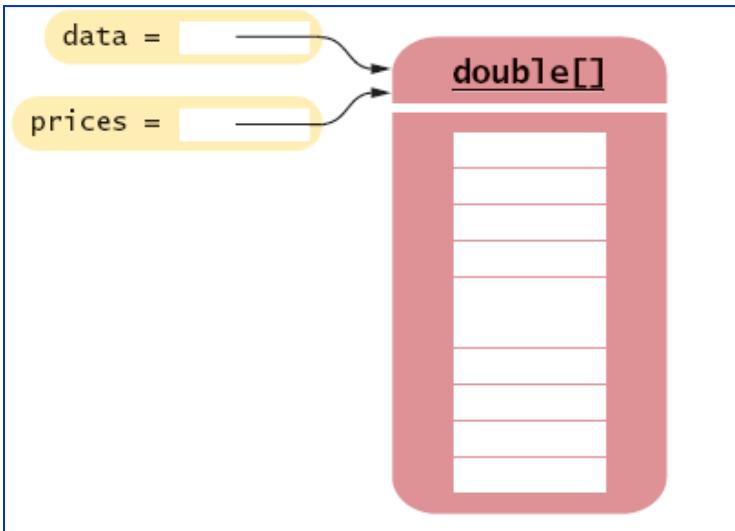
Phanupan	Khwanchai

Nutthawan	Treerat

การคัดลอกอาร์เรย์

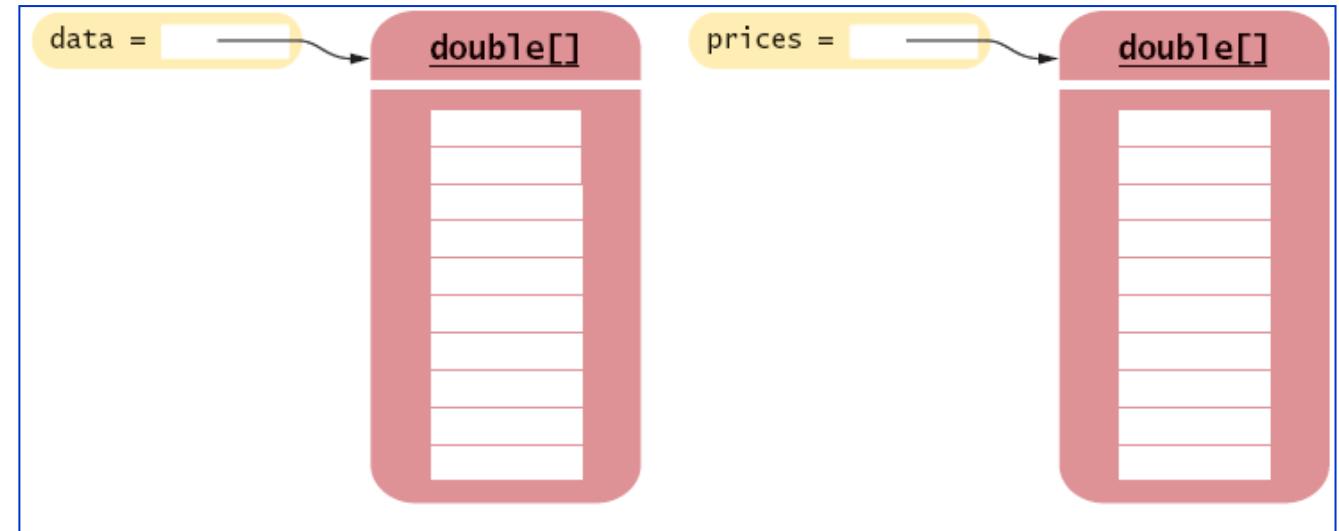
1. วิธีคัดลอกตัวชี้ที่ซึ่งไปยังอาร์เรย์เดียวกัน

```
double [ ] data = new double[10];  
// .ใส่ค่าในอะเรย์...  
  
double [ ] prices = data;
```



2. วิธีคัดลอกตัวชี้ที่ซึ่งไปยังอาร์เรย์ต่างกัน

```
double [ ] prices=(double[ ]) data.clone();
```



ไฟล์ ArrayCopy1.java การคัดลอกตัวชี้ไปยังอาร์เรย์เดียวกัน

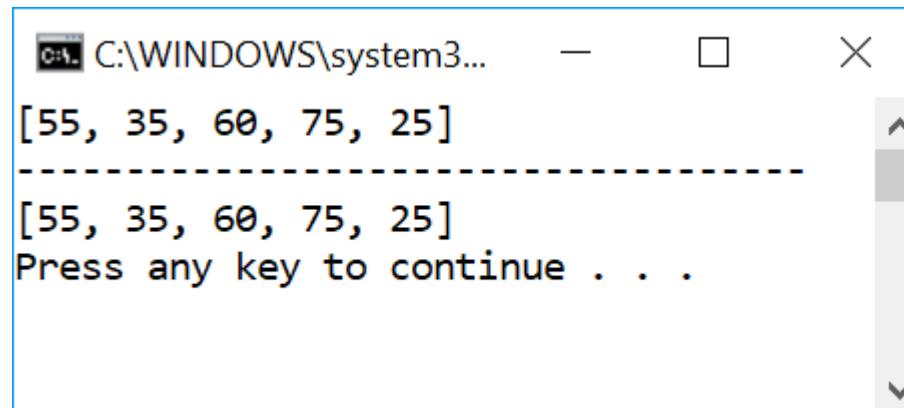
```
import java.util.Arrays;

public class ArrayCopy1 {
    public static void main(String[] args) {
        int number[] = {55, 35, 40, 75, 25};
        int n[] = number;

        number[2] = 60;

        System.out.println(Arrays.toString(number));
        System.out.println("-----");
        System.out.println(Arrays.toString(n));
    }
}
```

ผลลัพธ์



ไฟล์ ArrayCopy2.java การคัดลอกอาร์เรย์ไปยังอาร์เรย์ที่ต่างกัน

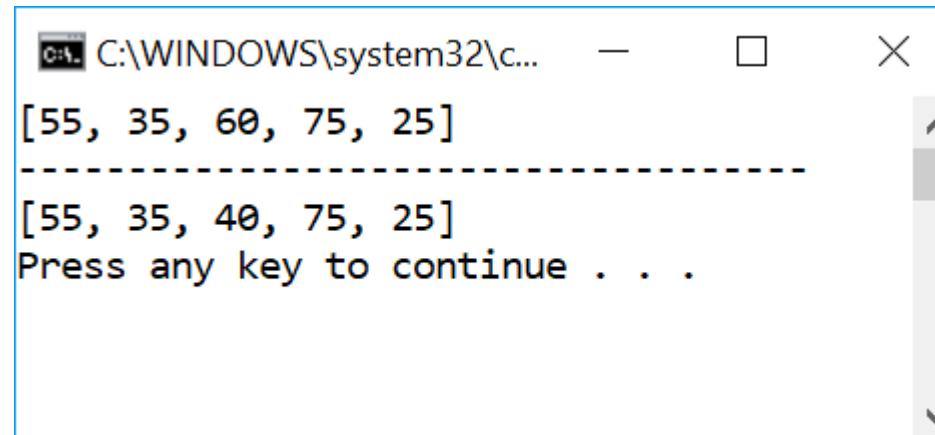
```
import java.util.Arrays;

public class ArrayCopy2 {
    public static void main(String[] args) {
        int number[] = {55, 35, 40, 75, 25};
        int n[] = (int[])number.clone();

        number[2] = 60;

        System.out.println(Arrays.toString(number));
        System.out.println("-----");
        System.out.println(Arrays.toString(n));
    }
}
```

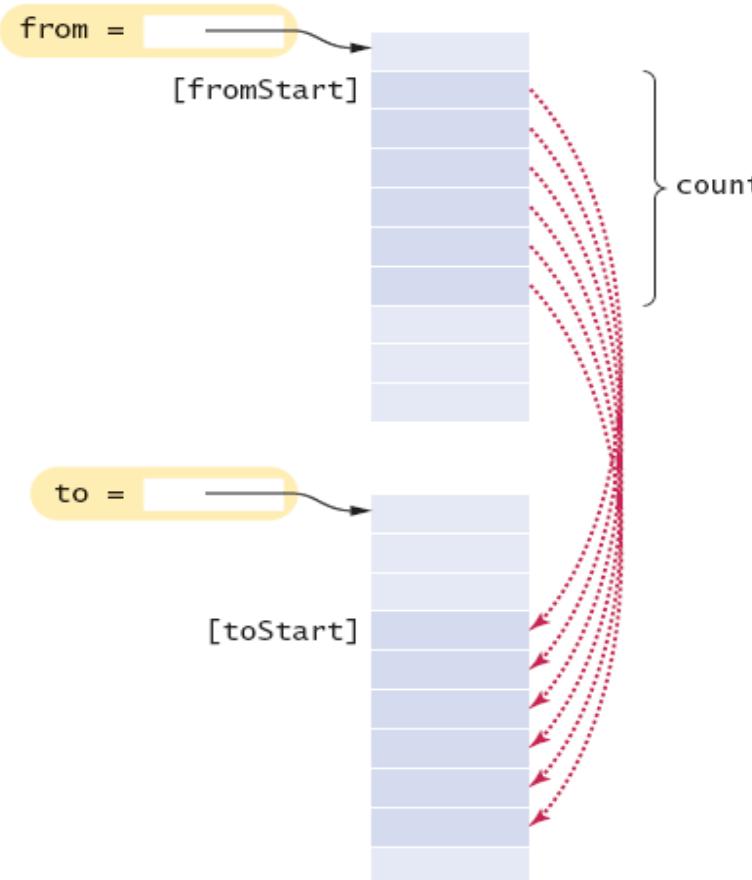
ผลลัพธ์



การคัดลอกอาร์เรย์

วิธีคัดลอกสมาชิกของอาร์เรย์

`System.arraycopy(from, fromStart, to, toStart, count);`



ไฟล์ CopyDemo.java การคัดลอกอาร์เรย์ตัวหนึ่งที่ต้องการไปยังอาร์เรย์อื่น

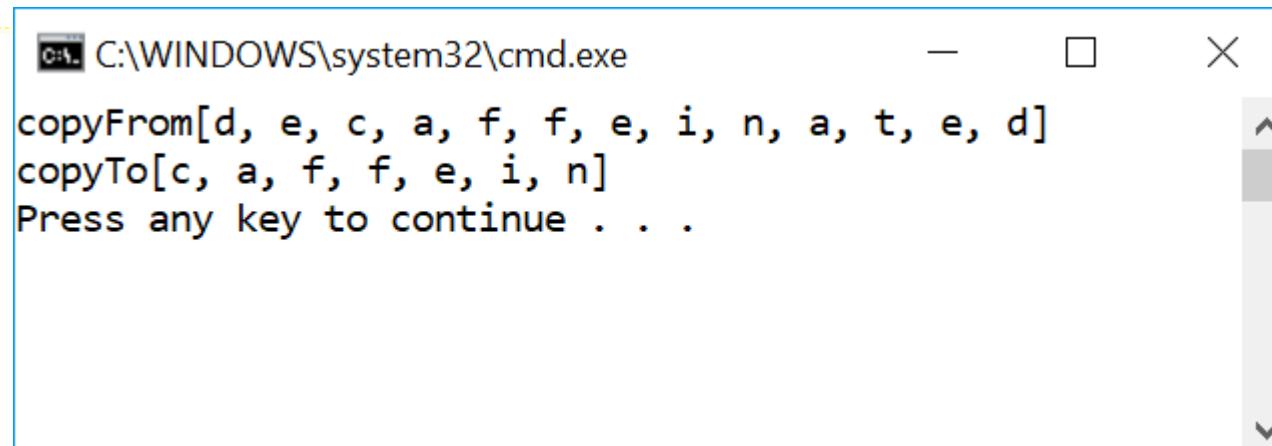
```
import java.util.Arrays;

class ArrayCopyDemo
{
    public static void main(String[] args)
    {
        char [] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e', 'i', 'n', 'a', 't', 'e', 'd' };
        char [] copyTo = new char[7];

        System.arraycopy(copyFrom, 2, copyTo, 0, 7);

        System.out.println("copyFrom" + Arrays.toString(copyFrom));
        System.out.println("copyTo" + Arrays.toString(copyTo));
    }
}
```

ผลลัพธ์



อาร์เรย์ลิสต์ (ArrayList)

- อาร์เรย์ลิสต์ (ArrayList)
 - เป็นคลาสในแพคเกจ `java.util` ที่ใช้สำหรับจัดเก็บข้อมูล/วัตถุชนิดต่างๆ
 - `ArrayList` มีการทำงานคล้ายกับ `Array` แต่สามารถขยายและลดขนาดในการจัดเก็บได้ตามต้องการ (มี `method` ต่างๆ ที่สนับสนุนการเพิ่ม/ลบ สมาชิกในอาร์เรย์ลิสต์)
 - การใช้งานใน `ArrayList` จะใช้ `index` โดยเริ่มนับจาก 0 เช่นเดียวกับ `Array` ทั่วไป

อาร์เรย์ลิสต์ (ArrayList)

การเรียกใช้ package

```
import java.util.ArrayList;
```

การประกาศใช้งานอะเรย์ลิสต์

```
ArrayList<ชนิดข้อมูล> ชื่อตัวแปร;
```

```
ชื่อตัวแปร = new ArrayList<ชนิดข้อมูล>();
```

หรือ

```
ArrayList<ชนิดข้อมูล> ชื่อตัวแปร = new ArrayList<ชนิดข้อมูล>();
```

ตัวอย่าง

```
ArrayList<Integer> number;
```

```
ArrayList<Integer> number = new ArrayList<Integer>();
```

```
ArrayList<String> names = new ArrayList<String>();
```

อาร์เรย์ลิสต์ (ArrayList)

```
ArrayList<String> names = new ArrayList<String>();
```

การเพิ่ม / การอ่าน / การกำหนดค่า

- การเพิ่ม/แทรกข้อมูลลงในอาร์เรย์ลิสต์
- การอ่านข้อมูลจากอาร์เรย์ลิสต์
- การกำหนดค่าในอาร์เรย์ลิสต์

```
names.add("Rewadee");  
names.add("Khwanchai");
```

```
String n = names.get(1);
```

อ่านค่าจากตัวแปรอาร์เรย์ลิสต์ names ตำแหน่งที่ 1 ในที่นี่ก็คือ Khwanchai

```
names.set(1, "Pisan");
```

กำหนดค่าตัวแปรอาร์เรย์ลิสต์ names ตำแหน่งที่ 1 จาก Khwanchai กลายเป็น Pisan แทน

อาร์เรย์ลิสต์ (ArrayList)

การดูจำนวนข้อมูล / การลบ

- การดูจำนวนข้อมูลในอาร์เรย์ลิสต์
- การลบข้อมูลออกจากอาร์เรย์ลิสต์
- การลบข้อมูลทั้งหมดออกจากอาร์เรย์ลิสต์

```
int size = names.size();
```

อ่านค่าขนาดของตัวแปรอาร์เรย์ลิสต์ names ได้เป็น 2

```
names.remove(1);  
names.remove("Rewadee");
```

```
names.clear();
```

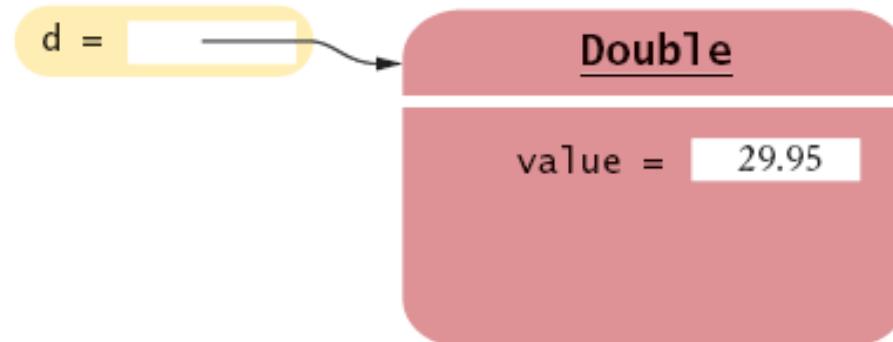
อาร์เรย์ลิสต์ (ArrayList)

- เราไม่สามารถเพิ่มข้อมูลที่มีชนิดข้อมูลแบบ Primitive Types เข้าไปในอะเรย์ลิสต์ได้โดยตรง แต่จะต้องทำให้มีลักษณะเป็น objects แล้วเรียกใช้ Wrapper classes แทน
- Wrapper Class: ชนิดของคลาสที่สามารถใช้ ArrayList ได้
 - ถ้าต้องการเก็บ char ให้ใช้ชนิดข้อมูลเป็น Character
 - ถ้าต้องการเก็บ int ให้ใช้ชนิดข้อมูลเป็น Integer
 - ถ้าต้องการเก็บ byte ให้ใช้ชนิดข้อมูลเป็น Byte
 - ถ้าต้องการเก็บ short ให้ใช้ชนิดข้อมูลเป็น Short
 - ถ้าต้องการเก็บ long ให้ใช้ชนิดข้อมูลเป็น Long
 - ถ้าต้องการเก็บ float ให้ใช้ชนิดข้อมูลเป็น Float
 - ถ้าต้องการเก็บ double ให้ใช้ชนิดข้อมูลเป็น Double
 - ถ้าต้องการเก็บ boolean ให้ใช้ชนิดข้อมูลเป็น Boolean
 - ถ้าต้องการเก็บ String ให้ใช้ชนิดข้อมูลเป็น String

Primitive Type	Wrapper Class
byte	Byte
boolean	Boolean
char	Character
double	Double
float	Float
int	Integer
long	Long
short	Short

อาร์เรย์ลิสต์ (ArrayList)

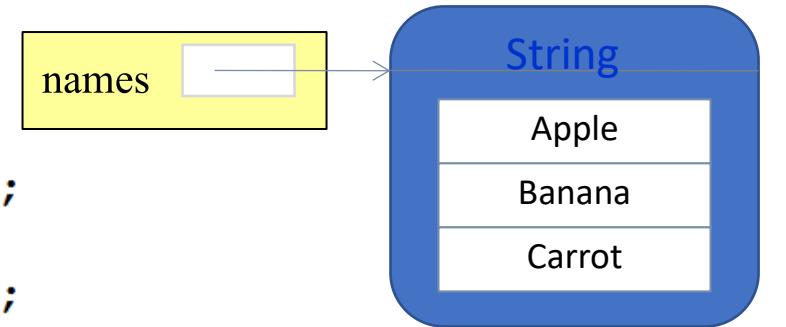
```
ArrayList<Double> d = new ArrayList<Double>();  
d.add(29.95);  
double x = d.get(0);
```



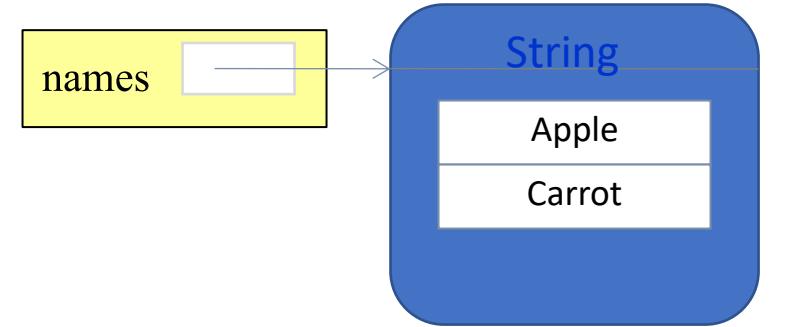
อาร์เรย์ลิสต์ (ArrayList)

```
import java.util.ArrayList;

public class TestArrayList {
    public static void main(String args[]) {
        ArrayList<String> names = new ArrayList<String>();
        names.add("Apple");
        names.add("Banana");
        names.add("Carrot");
        System.out.println("Name = " +names);
        names.remove("Banana");
        System.out.println("Name = " +names);
    }
}
```



Name = [Apple, Banana, Carrot]



Name = [Apple, Carrot]

```
import java.util.*;
class ArrayListDemo
{
    public static void main(String args[])
    {
        ArrayList <Character> al = new ArrayList<Character>();
        System.out.println("Initial size of al: " + al.size());
        al.add("C");
        al.add("A");
        al.add("E");
        al.add("B");
        al.add(1, "A2");
        System.out.println("Size of al after additions: " + al.size());
        System.out.println("Contents of al: " + al);
        al.remove("E");
        al.remove(2);
        System.out.println("Size of al after deletions: " + al.size());
        System.out.println("Contents of al: " + al);
    }
}
```

Initial size of a1: 0

Size of a1 after additions: 5

Contents of a1: [C, A2, A, E, B]

Size of a1 after deletions: 3

Contents of a1: [C, A2, B]

```
import java.util.*;
class ArrayListDemo
{
    public static void main(String args[])
    {
        ArrayList<Character> al = new ArrayList<Character>();
        System.out.println("Initial size of al: " + al.size());
        al.add("C");
        al.add("A");
        al.add("E");
        al.add("B");
        al.add(1, "A2");
        al.remove("E");
        al.remove(2);
        for (int i=0 ; i<al.size(); i++)
            System.out.println(al.get(i));
        System.out.println("Initial size of al: " + al.size());
    }
}
```

Initial size of a1: 0
C
A2
B
Initial size of a1: 3

อาร์เรย์ลิสต์ของคลาส

- อาร์เรย์ลิสต์ของคลาส: คือตัวแปรซึ่งเก็บค่าวัตถุของคลาสได้หลายวัตถุ เริ่มวัตถุที่ลำดับที่ 0 เช่นเดียวกับอาร์เรย์ลิสต์แบบทั่วไป

รูปแบบการประกาศของเรย์ลิสต์ของ

```
ArrayList<ClassName> arrlistName = new ArrayList<ClassName>();
```

โดย `ClassName` = ชนิดข้อมูลแบบคลาสรึคือชื่อคลาสนั่นเอง เช่น

Book, Dog, Employee เป็นต้น

`arrlistName` = ชื่อของตัวแปรอาร์เรย์ลิสต์

ไฟล์ TestArrayList.java แสดงการประกาศและกำหนดค่า ให้อาร์เรย์ลิสต์ของคลาส

```
import java.util.ArrayList;

public class TestArrayList {
    public static void main(String args[]) {
        ArrayList<Employee> emp = new ArrayList<Employee>();
        emp.add(new Employee("001", "Sompong"));
        emp.add(new Employee("002", "Somsri", 30, 18500));

        Employee e1 = new Employee();
        e1.setID("003");
        e1.setName("Vanida");
        e1.setAge(25);
        e1.setSalary(15000);

        emp.add(e1);
    }
}
```

อาร์เรย์ลิสต์ของคลาส

- การอ่านข้อมูลจากอาร์เรย์ลิสต์ ใช้เมธอด `get(index)`

```
System.out.println("id : " + emp.get(0).getID()) ;  
System.out.println("-----");  
emp.get(1).getEmployeeInfo();
```

```
import java.util.ArrayList;

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

        ArrayList<Employee> emp = new ArrayList<Employee>();

        emp.add(new Employee("001" , "Sompong"));
        emp.add(new Employee("002" , "Somsri" , 30 , 18500));

        Employee e1 = new Employee();
        e1.setID("003");
        e1.setName("Vanida");
        e1.setAge(25);
        e1.setSalary(15000);
        emp.add(e1);

        System.out.println("id : " + emp.get(0).getID());
        System.out.println("-----");
        emp.get(1).getEmployeeInfo();
    }
}
```

อาร์เรย์ลิสต์ของคลาส

- การหาขนาดของอาร์เรย์ลิสต์ ใช้เมธอด size()

```
for(int i=0; i< emp.size(); i++)
{
    emp.get(i).getEmployeeInfo();
}
```

The Generalized for Loop

for (ชนิดข้อมูล ชื่อตัวแปรที่ใช้วนลูป: ชื่อตัว
แปรที่เก็บข้อมูล)
คำสั่งต่างๆ

- การวนลูปแบบ Generalized for loop:

```
double[] data = . . .;
```

```
double sum = 0;
```

```
for (double e : data) // อ่านว่า “สำหรับแต่ละ e ใน data”
```

```
    sum = sum + e;
```

- การวนลูปโดยทั่วไป:

```
double[] data = . . .;
```

```
double sum = 0;
```

```
for (int i = 0; i < data.length; i++)
```

```
{ double e = data[i];
```

```
    sum = sum + e;
```

```
}
```

ตัวอย่าง:

```
for (double e : data)
    sum = sum + e;
```

The Generalized for Loop

- การนำ Generalized for loop มาใช้กับ ArrayLists

```
ArrayList<BankAccount> accounts = . . . ;
```

```
double sum = 0;  
for (BankAccount a : accounts)  
{  
    sum = sum + a.getBalance();  
}
```

- การวนลูปโดยทั่วไป :

```
double sum = 0;  
for (int i = 0; i < accounts.size(); i++)  
{  
    BankAccount a = accounts.get(i);  
    sum = sum + a.getBalance();  
}
```

จบร้านนำเสนอ