

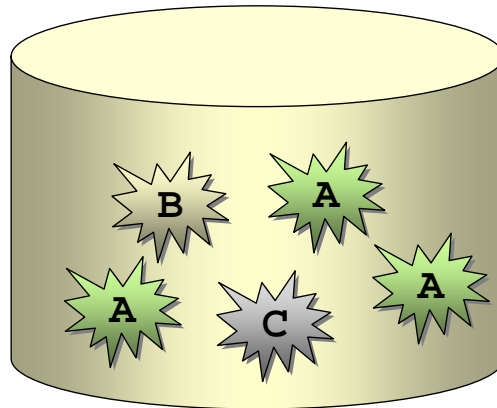
# การเก็บข้อมูลด้วยอาเรย์ (ArrayCollection)

## หัวข้อ

- คอลเล็กชัน
  - นิยาม
  - บริการ
  - การใช้งาน
- การสร้างคอลเล็กชันด้วยอาเรย์
- เรื่องจุกจิกของจาวา

# คอลเล็กชัน (Collection)

- ที่เก็บกลุ่มของข้อมูลประเภทเดียวกัน
- ไม่มีอันดับ
- มีข้อมูลซ้ำได้
- บริการ : add, remove, contains, size, isEmpty



## Collection Interface

```
public interface Collection {  
    public void add(Object element);  
    public void remove(Object element);  
    public boolean isEmpty();  
    public boolean contains(Object element);  
    public int size();  
}
```

```
public class ArrayCollection implements Collection {  
    public ArrayCollection(int cap) { ... }  
    public void add(Object element) { ... }  
    public void remove(Object element) { ... }  
    public boolean isEmpty() { ... }  
    public boolean contains(Object element) { ... }  
    public int size() { ... }  
}
```

# ตัวอย่างการใช้คอลเล็กชัน

```
public class TestCollection {  
    public static void main(String[] args) {  
        Collection c = new ArrayCollection(100);  
        System.out.println(c.isEmpty()); true  
        c.add("BANGKOK");  
        c.add("PHUKET");  
        c.add("BANGKOK");  
        c.add("SONGKLA");  
        System.out.println(c.size()); 4  
        c.remove("PHUKET");  
        System.out.println(c.contains("PHUKET")); false  
        System.out.println(c.contains("BANGKOK")); true  
    }  
}
```



# โปรแกรมตรวจสอบข้อความซ้ำซ้อน

- แฟ้มข้อความที่ได้รับ  
มีบรรทัดที่มีข้อความซ้ำกันหรือไม่

Hello  
สวัสดี  
ลาก่อน  
good bye  
สวัสดี

Hello  
สวัสดี  
ลาก่อน  
good bye  
สวัสดีครับ

# โปรแกรมตรวจสอบข้อความซ้ำซ้อน

```
import java.io.*;

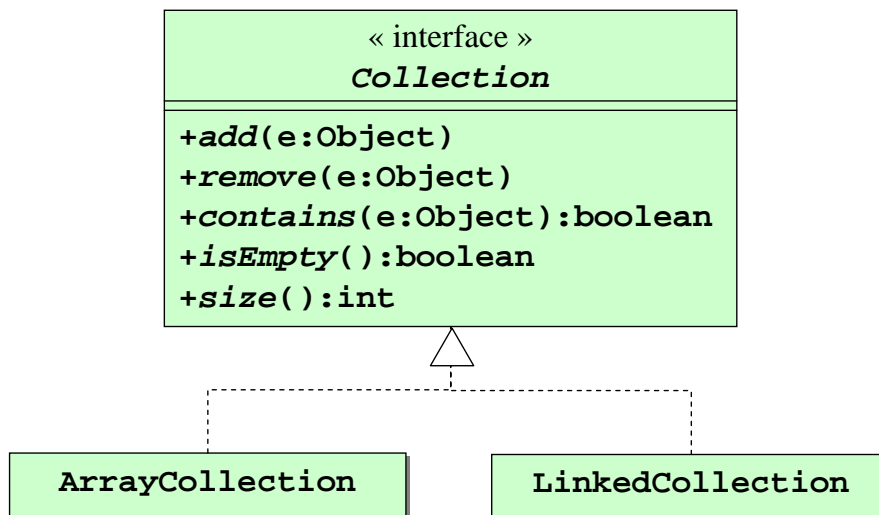
public class Test {
    public static void main(String[] args)
        throws IOException {
        FileReader fr = new FileReader("data.txt");
        BufferedReader br = new BufferedReader(fr);
        Collection c = new ArrayCollection(100);
        String line;
        while ((line = br.readLine()) != null) {
            if (c.contains(line))
                System.out.println("ข้อความซ้ำ : " + line);
            else
                c.add(line);
        }
        System.out.println(c.size());
    }
}
```

© S. Prasitjutrakul 2006

04/10/49 7

## แล้วจะสร้างคอลเล็กชันได้อย่างไร ?

- ใช้อาร์เรย์ (ArrayCollection)
- ใช้การโยง (LinkedList)

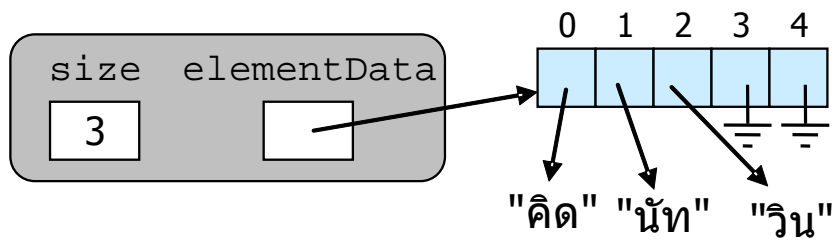


© S. Prasitjutrakul 2006

04/10/49 8

# ArrayCollection : ใช้อาร์เรย์เก็บข้อมูล

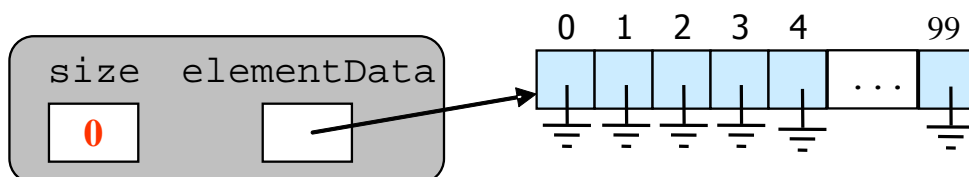
```
public class ArrayCollection implements Collection {  
    private Object[] elementData;  
    private int      size;  
    ...  
}
```



ข้อมูลในคอลเล็กชันเก็บอยู่ใน  
elementData ช่องที่ 0 ถึง size - 1

# ArrayCollection : constructor

```
public class ArrayCollection implements Collection {  
    private Object[] elementData;  
    private int      size;  
  
    public ArrayCollection(int c) {  
        elementData = new Object[c];  
        size = 0;  
    }  
}
```

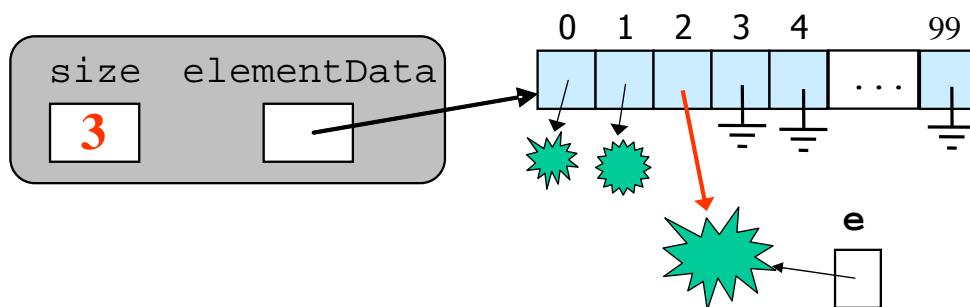


```
Collection c = new ArrayCollection(100);
```

# ArrayCollection : add

```
public class ArrayCollection implements Collection {  
    private Object[] elementData;  
    private int      size;  
    ...  
    public void add(Object e) {  
        if(e == null) throw new IllegalArgumentException();  
        elementData[size++] = e;  
    }  
}
```

นำตัวใหม่ต่อท้ายตัวสุดท้าย  
ใส่ใน elementData[size]

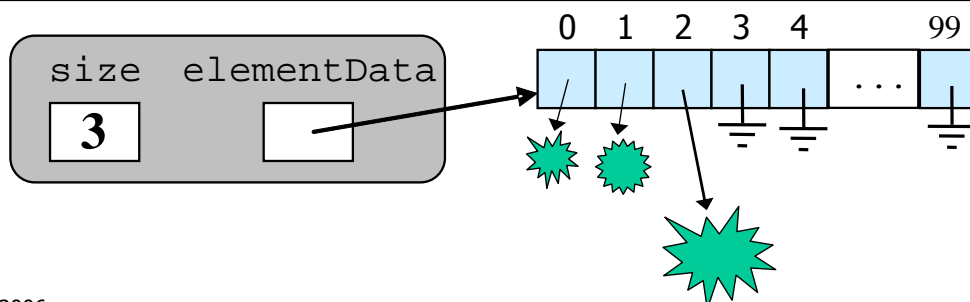


# ArrayCollection : size กับ isEmpty

```
public class ArrayCollection implements Collection {  
    private Object[] elementData;  
    private int      size;  
    ...  
    public int size() {  
        return size;  
    }  
    public boolean isEmpty() {  
        return size == 0;  
    }  
}
```

ใช้ size ให้เป็นประโยชน์

คืน true ถ้า size เป็น 0  
คืน false ถ้ามีข้อมูล  
(size ≠ 0)



# ArrayCollection : บริการต่าง ๆ

```
public class ArrayCollection implements Collection {  
    private Object[] elementData;  
    private int size;  
  
    public ArrayCollection(int c) {  
        elementData = new Object[c];  
        size = 0;  
    }  
    public void add(Object e) {  
        if(e == null) throw new IllegalArgumentException();  
        elementData[size++] = e;  
    }  
    public int size() {  
        return size;  
    }  
    public boolean isEmpty() {  
        return size == 0;  
    }  
    ...  
}
```

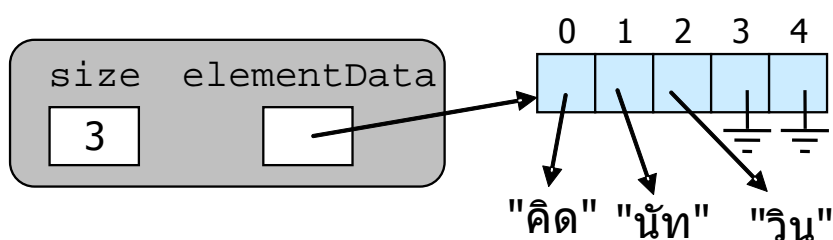
องค์ประกอบของ  
โครงสร้างภายใน

บริการสาธารณะ  
ต่าง ๆ ให้กับผู้ใช้

# ArrayCollection : contains

```
public class ArrayCollection implements Collection {  
    ...  
    private int indexOf(Object e) {  
        for (int i=0; i<size; i++)  
            if (elementData[i].equals(e)) return i;  
        return -1;  
    }  
    public boolean contains(Object e) {  
        return indexOf(e) != -1;  
    }  
}
```

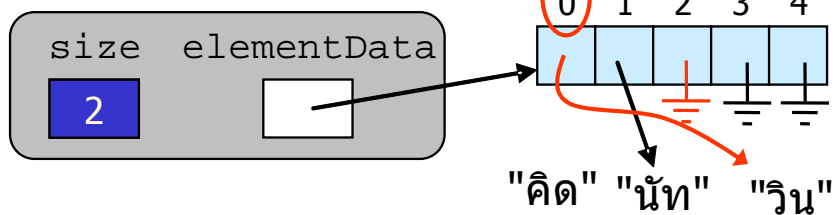
ไม่พบ e, คืน -1  
พบ e, คืน index ที่พบ



# ArrayCollection : remove

```
public class ArrayCollection implements Collection {  
    ...  
    public void remove(Object e) {  
        int i = indexOf(e);  
        if (i != -1) {  
            elementData[i] = elementData[--size];  
            elementData[size] = null;  
        }  
    }  
}
```

ย้ายตัวท้ายมาแทน  
ตัวที่ต้องการลบออก



## เรื่องลึกลับของจาวา

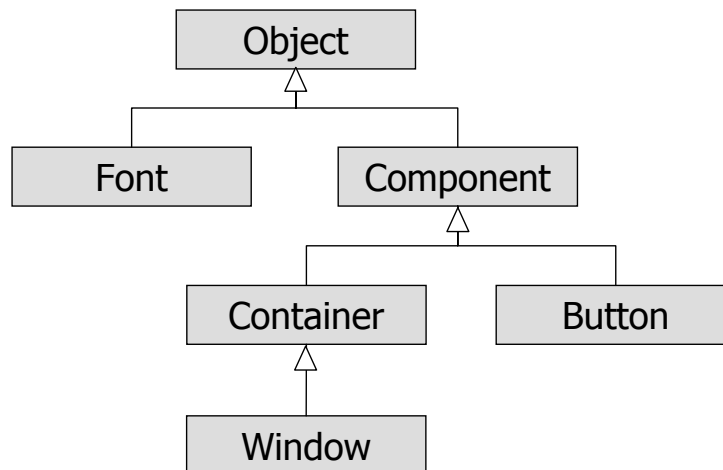
```
public class ArrayCollection implements Collection {  
    private Object[] elementData;  
    private int size;  
    ...  
    private int indexOf(Object e) {  
        for (int i=0; i<size; i++)  
            if (elementData[i].equals(e)) return i;  
        return -1;  
    }  
    public void remove(Object e) {  
        int i = indexOf(e);  
        if (i != -1) {  
            elementData[i] = elementData[--size];  
            elementData[size] = null;  
        }  
    }  
    ...  
}
```



# จาวา : คลาส Object เป็น root class

```
public class ArrayCollection implements Collection {  
    private Object[] elementData;  
    private int size;  
    ...  
}
```

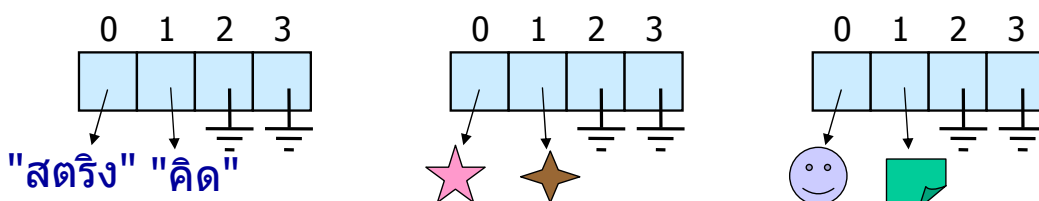
- ทุก ๆ คลาสในจาวาเป็นลูกหลานของคลาส Object



# จาวา : ทุกอ็อบเจกต์เป็น Object

```
public class ArrayCollection implements Collection {  
    private Object[] elementData;  
    private int size;  
    ...  
}
```

- ทุก ๆ คลาสในจาวาเป็นลูกหลานของคลาส Object
- ทุกอ็อบเจกต์ "เป็น" อ็อบเจกต์ของคลาส Object
- Object[] จึงเป็นอาร์เรย์เก็บอ็อบเจกต์ได้ทุกแบบ



# Wrapper Class

- Object[] เก็บข้อมูลพื้นฐานไม่ได้
  - int, short, byte, long, char, float, double, boolean
- ให้ใช้ Wrapper classes
  - Integer, Short, Byte, Long, Character, Float, Double, Boolean

```
Collection c = new ArrayCollection(10);  
c.add(new Integer(12));  
c.add(new Float(17.2f));  
c.add(new Double(13.2));
```

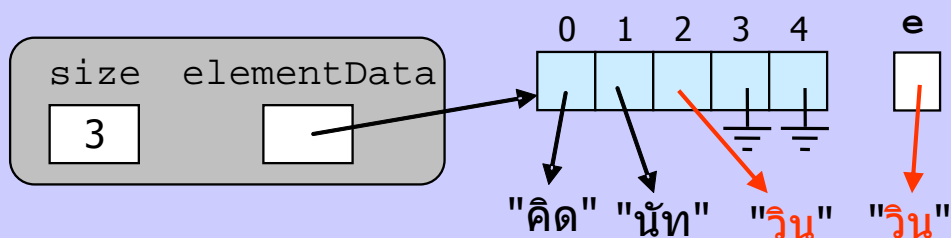
```
Collection c = new ArrayCollection(10);  
c.add(12);  
c.add(17.2f);      Java 5 ทำ autoboxing ให้  
c.add(13.2);
```

# จาวา : equals

```
public class ArrayCollection implements Collection {  
    ...  
    private int indexOf(Object e) {  
        for (int i=0; i<size; i++)  
            if (elementData[i].equals(e)) return i;  
        return -1;  
    }  
}
```

```
if (elementData[i].equals(e)) return i;
```

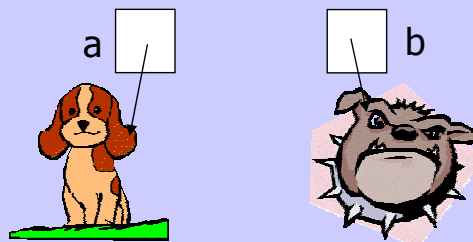
อ็อบเจกต์ elementData[i] มีค่าเท่ากับอ็อบเจกต์ e หรือไม่



# จาวา : dereference

ในจาวา มีแต่ขอสร้างอ็อบเจกต์ "ไม่มีการคืนอ็อบเจกต์  
อ็อบเจกต์ใดที่ไม่มีตัวแปรอ้างอิงถือว่าเป็น "ขยะ"

```
Dog a = new Dog("ต่าง");  
Dog b = new Dog("บุญ");  
a = b;  
a = null;  
b = null;
```



```
public void remove(Object e) {  
    int i = indexOf(e);  
    if (i != -1) {  
        elementData[i] = elementData[--size];  
        elementData[size] = null;  
    }  
}
```

# อาเรย์อาจล้นได้

```
public class ArrayCollection implements Collection {  
    private Object[] elementData;  
    private int      size;  
    ...  
    public void add(Object e) {  
        if(e == null) throw new IllegalArgumentException();  
        elementData[size++] = e;  
    }  
}
```

```
Collection c = new ArrayCollection(3);  
c.add("nok");  
c.add("kai");  
c.add("bird");  
c.add("poo");
```

3 ช่องเก็บ 3 ตัว

# ArrayCollection : add แบบขยายอาเรย์ได้

```
public class ArrayCollection implements Collection {
    private Object[] elementData;
    private int      size;
    ...
    public void add(Object e) {
        if(e == null) throw new IllegalArgumentException();
        ensureCapacity(size + 1);
        elementData[size++] = e;
    }
    private void ensureCapacity(int capacity) {
        if (capacity > elementData.length) {
            int s = Math.max(capacity, 2*elementData.length);
            Object[] arr = new Object[s];
            for(int i = 0; i < size; i++)
                arr[i] = elementData[i];
            elementData = arr;
        }
    }
}
```

ขยายอาเรย์ถ้ามีขนาด  
ไม่พอ capacity

ขยายให้ใหญ่ขึ้น  
สองเท่า

## บริการเสริม : toArray()

- มีคอลเล็กชันแล้วต้องการดึงข้อมูลออกมาใช้งาน
- toArray คือนอาเรย์
  - มีขนาดเท่ากับจำนวนข้อมูลของคอลเล็กชัน
  - เก็บข้อมูลทุกตัวในคอลเล็กชัน

```
ArrayCollection c = new ArrayCollection(4);
c.add("A");
c.add("B");
c.add("C");
c.add("D");
c.remove("B");
Object[] a = c.toArray();
System.out.println(a.length);
System.out.println(Arrays.toString(a));
```

[A,D,C]

# ArrayCollection : toArray()

```
public class ArrayCollection implements Collection {
    private Object[] elementData;
    private int      size;
    ...
    public Object[] toArray() {
        Object[] a = new Object[size];
        for (int i=0; i<size; i++)
            a[i] = elementData[i];
        return a;
    }
}
```

## สรุป

- คอลเล็กชันคือที่เก็บข้อมูลประเภทหนึ่ง
  - ไม่สนใจอันดับของข้อมูล
  - เก็บข้อมูลซ้ำได้
- ArrayCollection : สร้างคอลเล็กชันด้วยอาเรย์
  - เก็บอ็อบเจกต์ได้ทุกประเภท (แต่ไม่เก็บ primitive)
  - มีอาเรย์และตัวแปรอีกตัวเก็บจำนวนข้อมูล
  - ทำให้อาเรย์ขยายขนาดได้ เมื่อเก็บข้อมูลจนเต็ม