

2110211 โครงสร้างข้อมูลเบื้องต้น

ArrayCollection

สมชาย ประสิทธิ์จตุระกุล

Collection

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

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



จาวามี java.util.Collection ที่มีบริการมากกว่า

class ArrayCollection implements Collection

```
import java.io.*;

public class TestArray {
    public static void main(String[] args)
        throws Exception {
        FileReader fr = new FileReader("TestArray.java");
        BufferedReader br = new BufferedReader(fr);
        String line;

        Collection c = new ArrayCollection();

        while ((line = br.readLine()) != null) {
            c.add(line.trim());
        }
        System.out.println(c.contains("String line;"));
        System.out.println(c.size());
    }
}
```

ArrayCollection : สร้างด้วยอาเรย์

- ภายใน collection มีอาเรย์ไว้เก็บข้อมูล
- มีข้อมูล n ตัว ก็เก็บในช่องที่ 0, 1, 2, ..., n-1
- ให้เป็นอาเรย์ของ Object (ซึ่งเป็น root class)
 - เก็บออบเจกต์ทุกชนิดได้ แต่เก็บ primitive type ไม่ได้
- เพื่อให้รู้จำนวนข้อมูล มีอีก field เก็บจำนวนข้อมูล

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

    public ArrayCollection() {
        elementData = new Object[10];
        size = 0;
    }
    ...
}
```

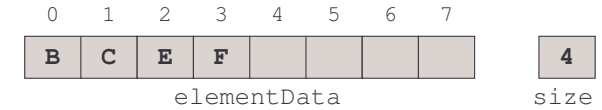
ArrayCollection : isEmpty, size

- size เป็น field ที่คอยเก็บจำนวนข้อมูลใน collection
- ใช้ size สำหรับบริการ isEmpty() และ size() ได้เลย

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

ArrayCollection : contains

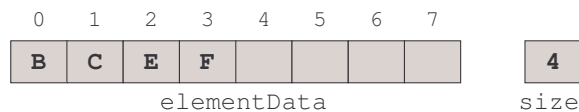
- ค่อย ๆ ริงหาจากช่องที่ 0 ถึง size-1
- การเปรียบเทียบออกเจดต์ต้องใช้ equals
 - ใช้ == เพื่อเปรียบเทียบว่าเป็นออกเจดต์เดียวกันหรือไม่



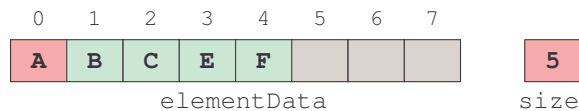
```
public class ArrayCollection implements Collection {
    Object[] elementData;
    int size;
    ...
    public boolean contains(Object element) {
        for(int i=0; i<size; i++) {
            if (elementData[i].equals(element)) return true;
        }
        return false;
    }
}
```

ArrayCollection : add

- เพิ่มข้อมูลใหม่ไว้ที่ช่อง 0 ต้องเลื่อนข้อมูล ซ้าย



add("A")



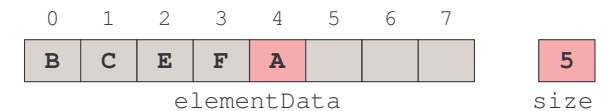
```
public void add(Object element) {
    for(int i=size; i>0; i--) {
        elementData[i] = elementData[i-1];
    }
    elementData[0] = element;
    size++;
}
```

ArrayCollection : add

- เพิ่มได้รวดเร็วมาก ถ้าเรานำข้อมูลใหม่ไปต่อท้าย



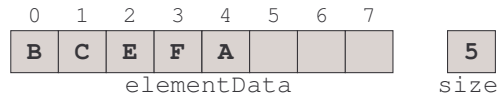
add("A")



```
public void add(Object element) {
    elementData[size] = element;
    size++;
}
```

ArrayCollection : remove

- ลบช้า ถ้าต้องเลื่อนขุดทางขวามาทางซ้าย 1 ช่อง

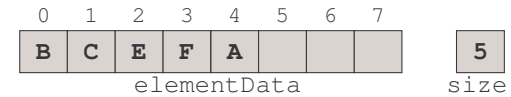


remove("C")

```
public void remove(Object element) {
    for(int i=0; i<size; i++) {
        if (elementData[i].equals(element)) {
            for(int j=i+1; j<size; j++) {
                elementData[j-1] = elementData[j];
            }
            size--;
            break;
        }
    }
}
```

ArrayCollection : remove

- ลบเร็วมาก ถ้านำตัวท้ายสุดมาทับตัวที่ถูกลบ



remove("C")

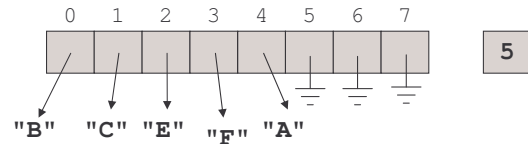
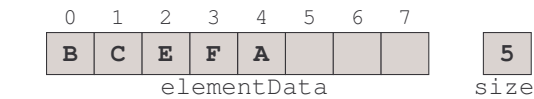
```
public void remove(Object element) {
    for(int i=0; i<size; i++) {
        if (elementData[i].equals(element)) {
            elementData[i] = elementData[size-1];
            size--;
            break;
        }
    }
}
```

```
public class ArrayCollection implements Collection {
    Object[] elementData;
    int size;
    public ArrayCollection() {
        elementData = new Object[100];
    }
    public boolean isEmpty() { return size == 0; }
    public int size() { return size; }
    public boolean contains(Object e) {
        for(int i=0; i<size; i++)
            if (elementData[i].equals(e)) return true;
        return false;
    }
    public void add(Object e) { elementData[size++] = e; }
    public void remove(Object e) {
        for(int i=0; i<size; i++) {
            if (elementData[i].equals(e)) {
                elementData[i] = elementData[size-1];
                size--;
                break;
            }
        }
    }
}
```

```
public class ArrayCollection implements Collection {
    Object[] elementData;
    int size;
    public ArrayCollection() {
        elementData = new Object[100];
    }
    public boolean isEmpty() { return size == 0; }
    public int size() { return size; }
    public boolean contains(Object e) {return indexOf(e) != -1;}
    public void add(Object e) { elementData[size++] = e; }
    public void remove(Object e) {
        int i = indexOf(e);
        if (i != -1) {
            elementData[i] = elementData[--size];
            elementData[size] = null; // next slide
        }
    }
    private int indexOf(Object e) {
        for(int i=0; i<size; i++)
            if (elementData[i].equals(e)) return i;
        return -1;
    }
}
```

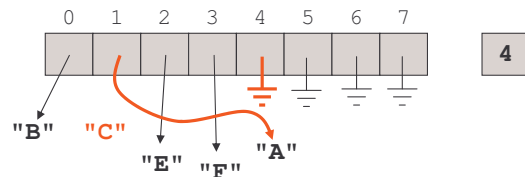
ปรับให้ทรู

จาวา : ข้อมูลที่ไม่มีใครอ้างอิงถือว่าเป็นขยะ



remove("C")

```
elementData[i] = elementData[--size];
elementData[size] = null;
```



Array มีขนาดจำกัด

- size ต้องไม่เกิน elementData.length
- จะจองอาเรย์สักกี่ช่อง ถึงจะพอ
- ถ้าไม่พอ ก็จะทำให้เกิด
ArrayIndexOutOfBoundsException

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

    public ArrayCollection() {
        elementData = new Object[100];
    }
    ...
    public void add(Object e) { elementData[size++] = e; }
    ...
}
```

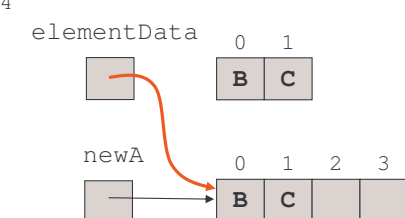
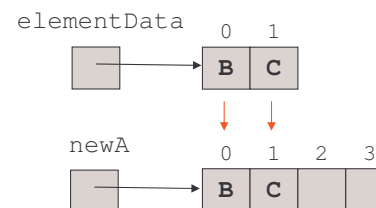
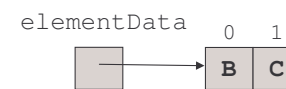
อาเรย์ขยายขนาดได้

- เมื่อข้อมูลเต็มอาเรย์
- จองอาเรย์ใหม่ให้ใหญ่กว่าเดิม
- ย้ายข้อมูลจากอาเรย์เก่า ไปยังอาเรย์ใหม่
- ให้ elementData อ้างอิงอาเรย์ใหม่

```
public void add(Object e) {
    ensureCapacity(size+1);
    elementData[size++] = e;
}

void ensureCapacity(int capacity) {
    if (capacity > elementData.length) {
        Object[] newA = new Object[2*elementData.length];
        for(int i=0; i<elementData.length; i++) {
            newA[i] = elementData[i];
        }
        elementData = newA;
    }
}
```

ขั้นตอนการขยายขนาดอาเรย์



newA เป็น local variable เมื่อ add ทำงานเสร็จ newA ก็ถูกทำลาย

การเก็บ null ใน Collection

- บางตัวใน elementData[0] ถึง elementData[size-1] เป็น null
- จะเกิดปัญหา (ตอนใช้ member ของ null)

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

NullPointerException
เมื่อ elementData[i] เป็น null

วิธีแก้ปัญหาเรื่องการเก็บ null

- ตรวจสอบมากขึ้น

```
private int indexOf(Object e) {
    for(int i=0; i<size; i++) {
        if (e == null && elementData[i] == null ||
            e != null && elementData[i].equals(e))
            return i;
    }
    return -1;
}
```

วิธีแก้ปัญหาเรื่องการเก็บ null

- ไม่อนุญาตให้เก็บ null
 - ต้องดัก ตรวจสอบตอนเพิ่มข้อมูล
 - ต้องเขียนเตือนไว้ในคู่มือการใช้งานด้วย

```
/**
 * add a new non-null element e in this collection
 *
 * @param e the new element
 * @throws IllegalArgumentException if e is null
 */
public void add(Object e) {
    if (e == null) throw new IllegalArgumentException();
    ...
    elementData[size++] = e;
}
```

การบ้าน : toArray()

- ผู้ใช้เก็บข้อมูลใน ArrayCollection ไปสักพัก แล้วอยากจะทำอะไรกับข้อมูลที่เก็บไว้มาประมวลผลทีละตัวจะทำอย่างไร ?
- ให้บริการใหม่ toArray()
 - คืนอาร์เรย์ของออบเจกต์ที่เก็บข้อมูลทุกตัวใน collection

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

การบ้าน : toString()

- toString เป็นเมทอดมาตรฐานของคลาส Object
- เพื่อให้การ debug โปรแกรมที่ใช้ ArrayCollection กระทำได้ง่าย เราควรมี toString()

```
public class ArrayCollection implements Collection {
    Object[] elementData;
    int size;
    ...
    public String toString() {
        ...
    }
}
```

```
Collection c = new ArrayCollection();
c.add("A");
c.add("B");
c.add("xyz");
System.out.println(c);
```

[A, B, xyz]

การบ้าน : equals()

- equals เป็นเมทอดมาตรฐานของคลาส Object
- override equals ใน ArrayCollection เพื่อใช้เปรียบเทียบ collection

```
public class ArrayCollection implements Collection {
    Object[] elementData;
    int size;
    ...
    public boolean equals(Object o) {
        if (! o instanceof ArrayCollection) return false;
        ArrayCollection that = (ArrayCollection) o;
        ... // equality test between this and that objects
    }
}
```

ข้อสังเกต : [x, y, z] เท่ากับ [y, x, z] แต่ [x, x] ไม่เท่ากับ [x]