

Collection

โครงสร้างข้อมูลเบื้องต้น

2301263

Collections

โครงสร้างแบบ Collection



เก็บข้อมูลแบบกลุ่ม



สามารถเอาค่าใส่ / เอาค่าออก



สามารถหาค่าที่กำหนดใน collection



สามารถบอกจำนวนค่าที่เก็บใน collection

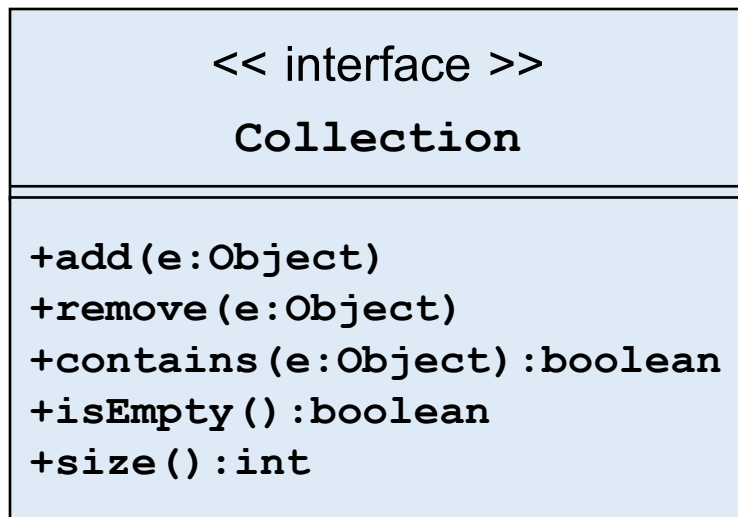


ไม่มีลำดับ



เก็บค่าซ้ำได้

interface



`void add(e:Object)`

ใส่ e ในคอลเล็กชัน

`void remove(e:Object)`

ลบ e ออกจากคอลเล็กชัน

`boolean contains(e:Object)`

หาว่า e อยู่ในคอลเล็กชันหรือไม่

`boolean isEmpty()`

มีค่าอยู่ในคอลเล็กชันหรือไม่

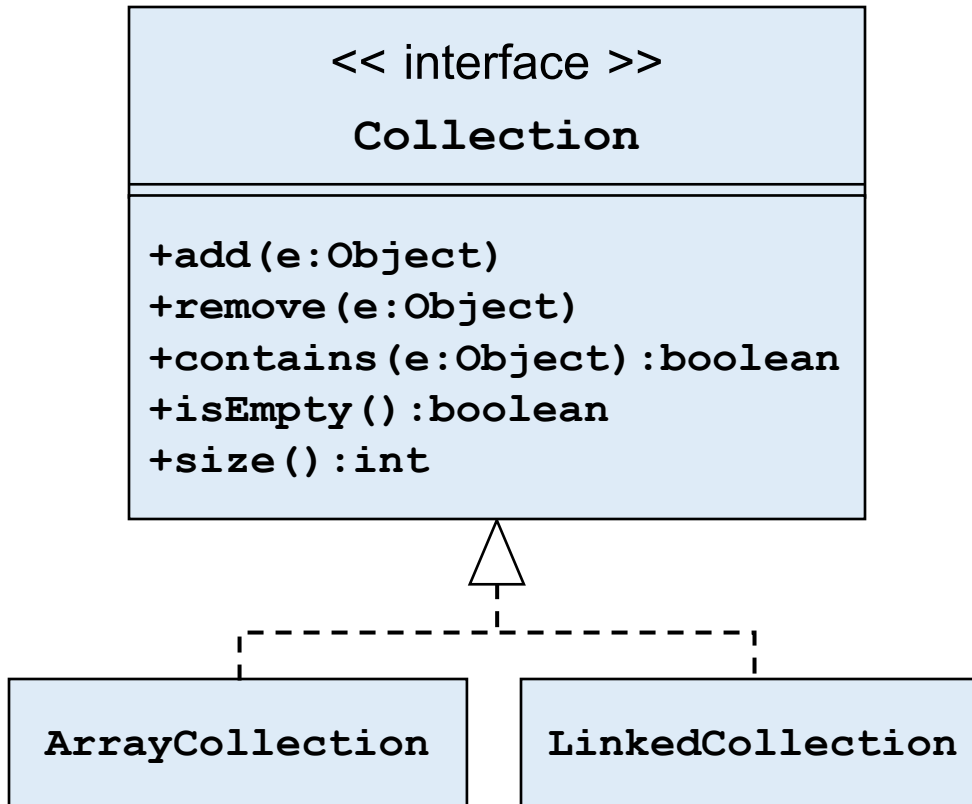
`int size()`

หาจำนวนของในคอลเล็กชัน

Interface Collection

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

interface



อินเทอร์เฟส Collection ใช้สำหรับ

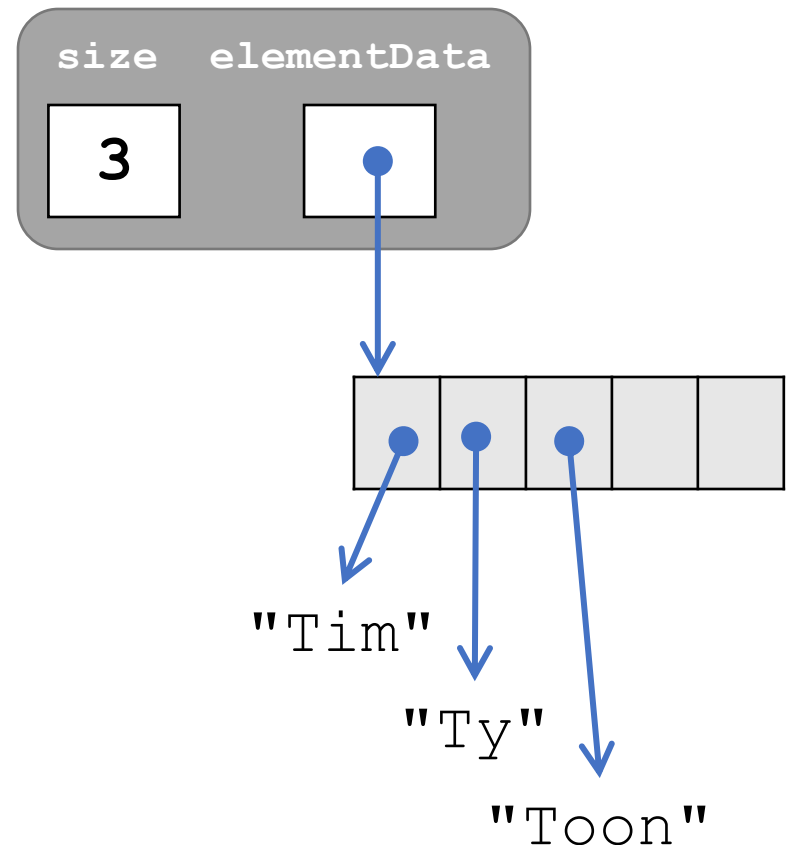
- คลาส ArrayCollection และ
- คลาส LinkedList

โดยเก็บค่าในแต่ละคลาสต่างกัน

ขั้นตอนการทำงานของเมทอดเดียวกันใน
แต่ละคลาสต่างกัน

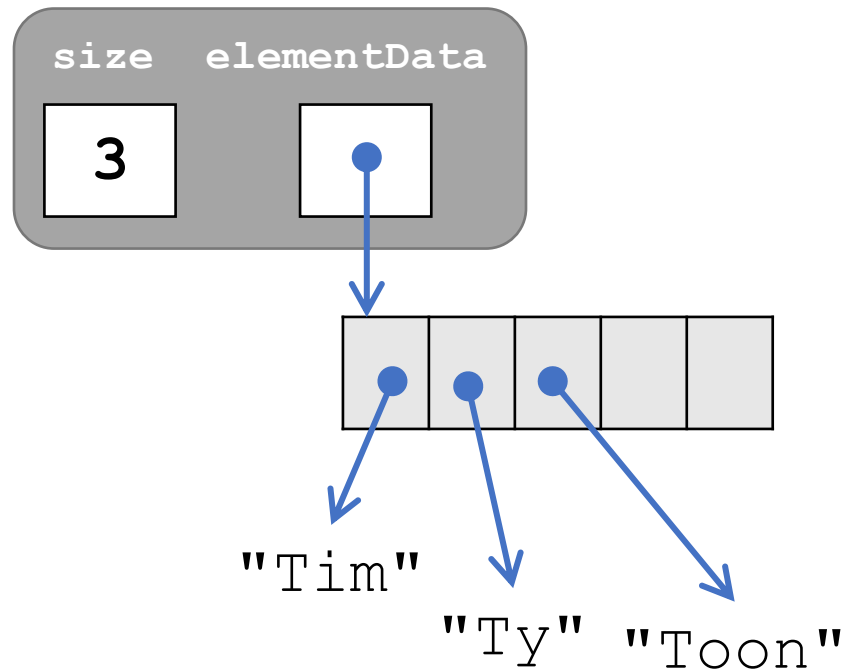
Class ArrayCollection

- Attribute ที่เก็บข้อมูลแต่ละตัว (**elementData**)
 - เป็น array ของ Object
- Attribute ที่เก็บจำนวนข้อมูล (**size**)
 - เป็น integer
 - ถ้าไม่มี attribute นี้ สามารถหาจำนวนโดยนับจำนวนข้อมูลใน **elementData**



Class ArrayCollection

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



Class ArrayCollection

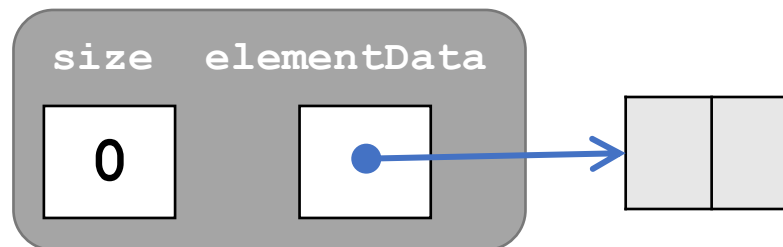
```
public class ArrayCollection implements Collection{
    private Object[] elementData;    // attribute
    private int size;                // attribute

    public ArrayCollection() {...}    // create new object
    public void add(Object e) {...}   // add e in object
    public int size() {...}           // get the size of object
    public boolean isEmpty() {...}    // is the object empty ?
    private void ensureCapacity(int capacity) {...}
    private int indexOf(Object e) {...}
    public void remove(Object e) {...} // remove e from object
    public boolean contains(Object e) {...} // is e in object?
    ...}
```


new instance: Class ArrayCollection

```
public ArrayCollection() {           // create an instance
    elementData = new Object[2];    // reserve some space
    size = 0;                       // nothing in the collection
}
```

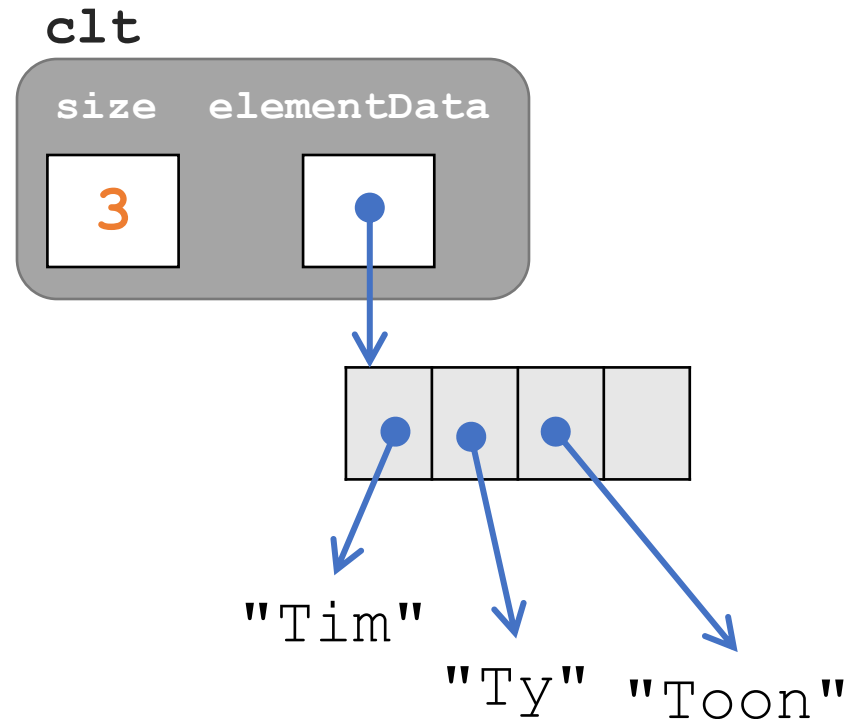
```
clt = new ArrayCollection();
```



size & isEmpty: Class ArrayCollection

```
public int size() {  
    return size;  
}
```

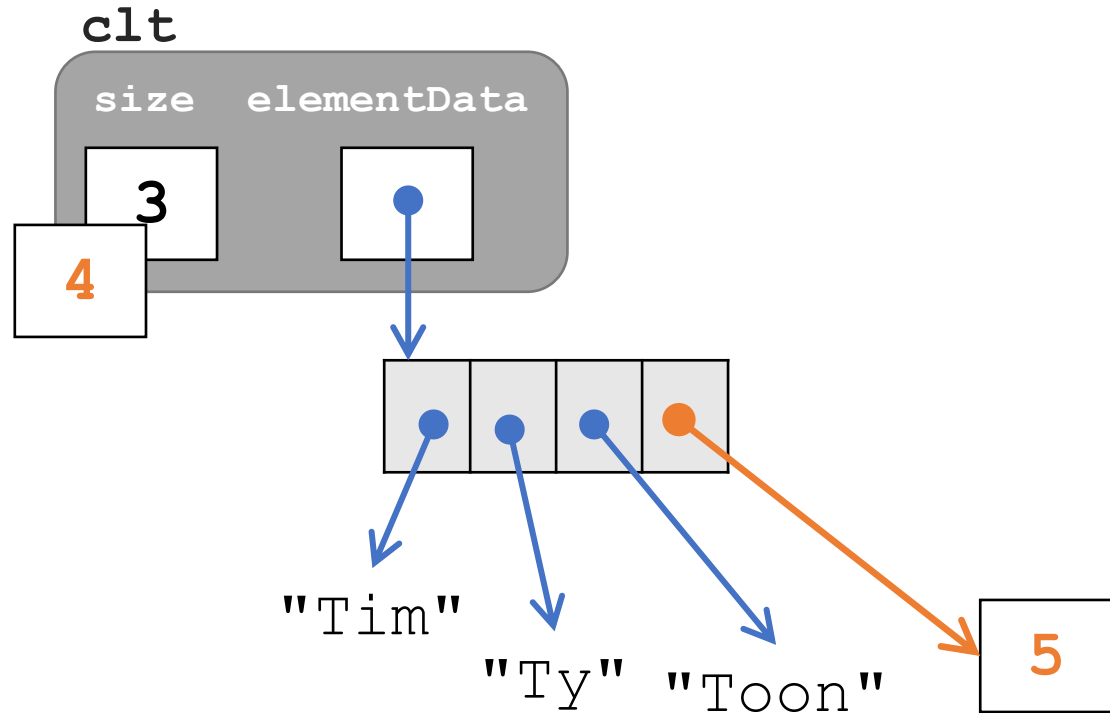
```
public boolean isEmpty() {  
    return size==0;  
}
```



add: Class ArrayCollection

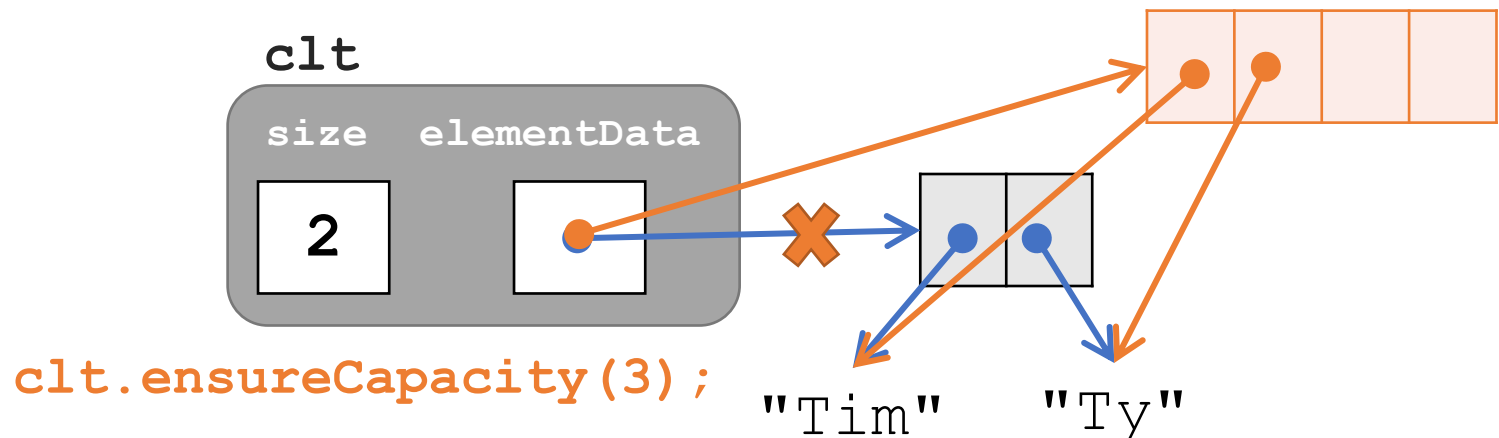
```
public void add(Object e) {  
    if (e==null) throw new IllegalArgumentException();  
    ensureCapacity(size+1);    // allocate space if needed  
    elementData[size++] = e;  
}
```

clt.add(5);



ขยาย elementData: Class ArrayCollection

```
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;  
    }  
}
```



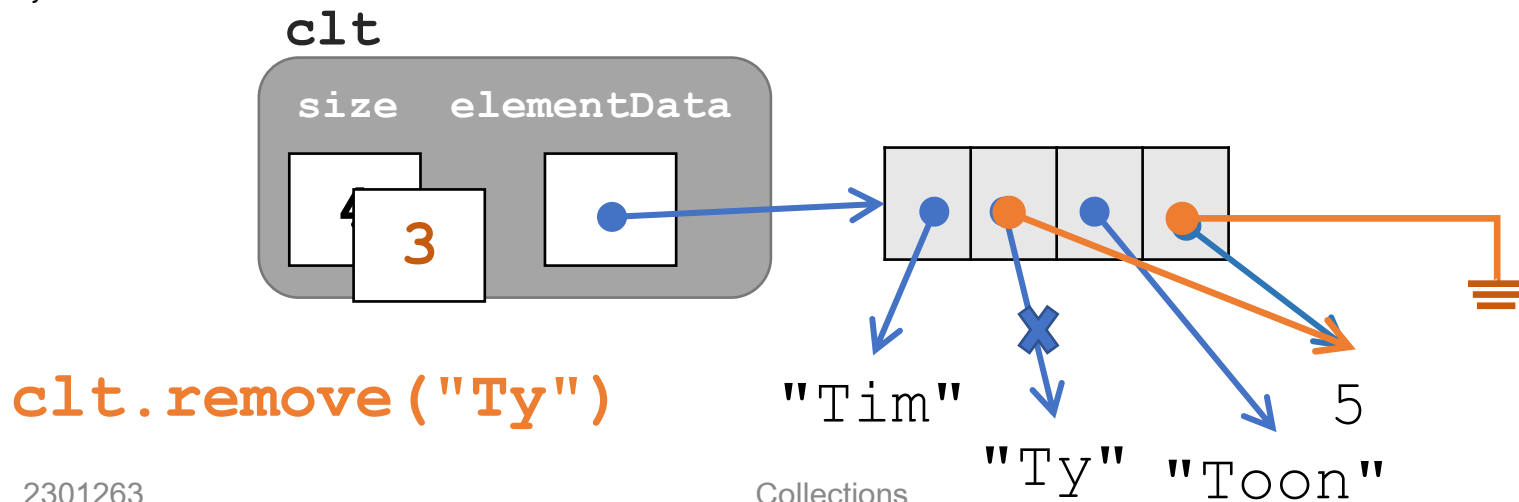
indexOf, contains: Class ArrayCollection

```
private int indexOf(Object e) { // return position of e
    for (int i=0; i<size; i++) {
        if (elementData[i].equals(e)) { // found e
            return i; // return the position of e
        }
    }
    return -1; // when e is not found, return -1
}
```

```
public boolean contains(Object e) {
    return indexOf(e) != -1;
}
```

remove: Class ArrayCollection

```
public void remove(Object e) {  
    int i = indexOf(e); //find the position of e  
    if (i != -1) {      // e is in the object  
        elementData[i] = elementData[--size]; // replace e  
        elementData[size] = null;  
    }  
}
```

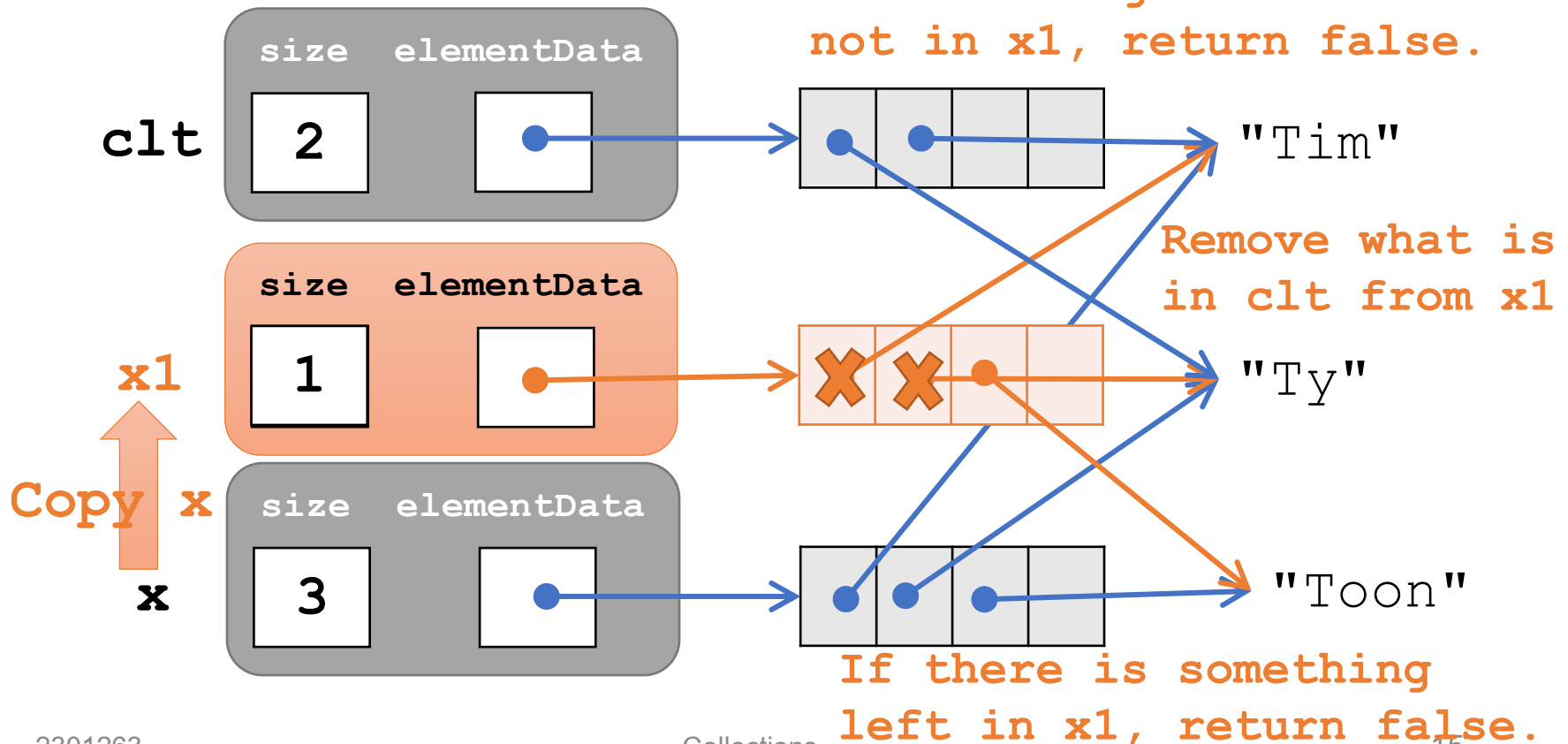


equals: Class ArrayCollection

```
public boolean equals(Object x) {...}
```

clt.equals(x)

If something in clt is not in x1, return false.



equals: Class ArrayCollection

```
public boolean equals(Object x) {
    Object e;
    if (x instanceof ArrayCollection) {//x is ArrayCollection
        // copy x to x1
        ArrayCollection x1 = new ArrayCollection();
        for (int i=0; i<((ArrayCollection) x).size; i++)
            x1.add(((ArrayCollection) x).elementData[i]);
        for (int i=0; i<size; i++) {
            e = elementData[i];
            if (!(x1.indexOf(e)==-1)) x1.remove(e);
            else return false;
        }
        return x1.isEmpty();
    }
    else return false;
}
```

toArray: Class ArrayCollection

```
public Object[] toArray() {  
    Object[] a = new Object[size];  
    for (int i=0; i<size; i++) {  
        a[i]= elementData[i];  
    }  
    return a;  
}
```

clt.toArray() **clt**

