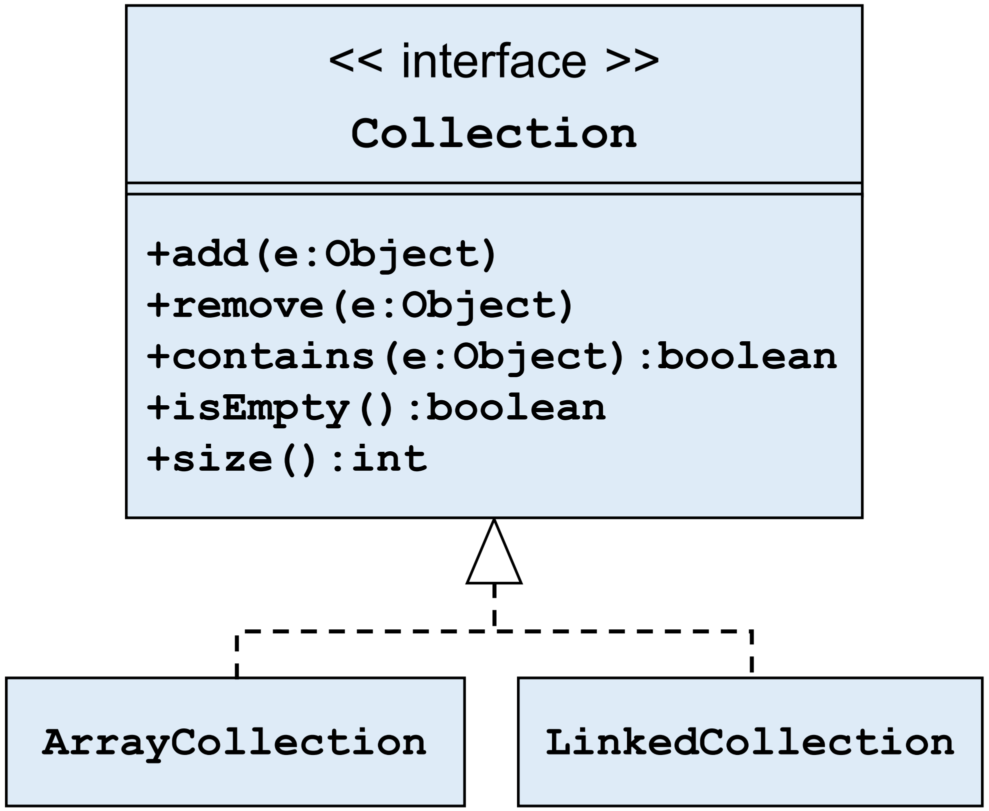
Linked Collection

โครงสร้างแบบโยง

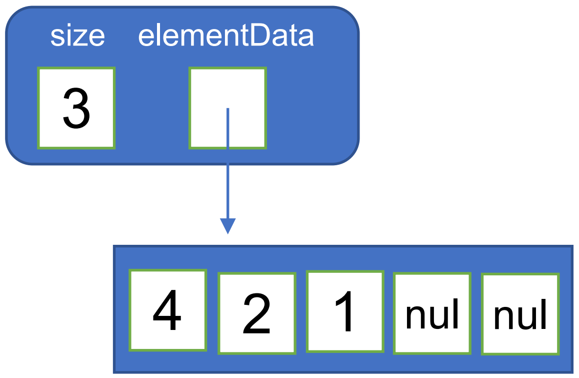
interface

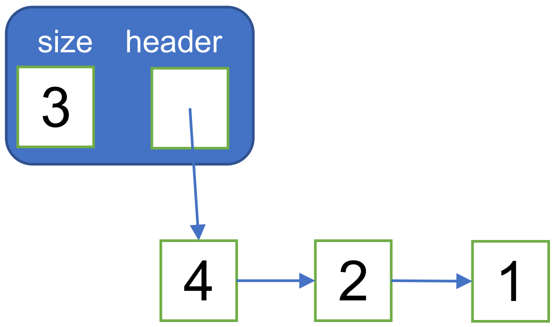


คลาส LinkedCollection ใช้อินเทอร์เฟสเดียวกับ ArrayCollection และโยงโหนด(node) เข้าด้วยกัน

มี node เป็น private class

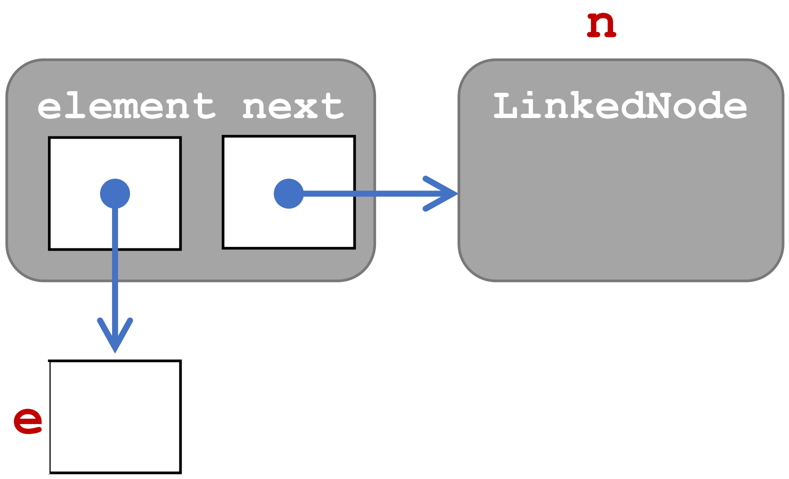
ArrayCollection vs. LinkedCollection





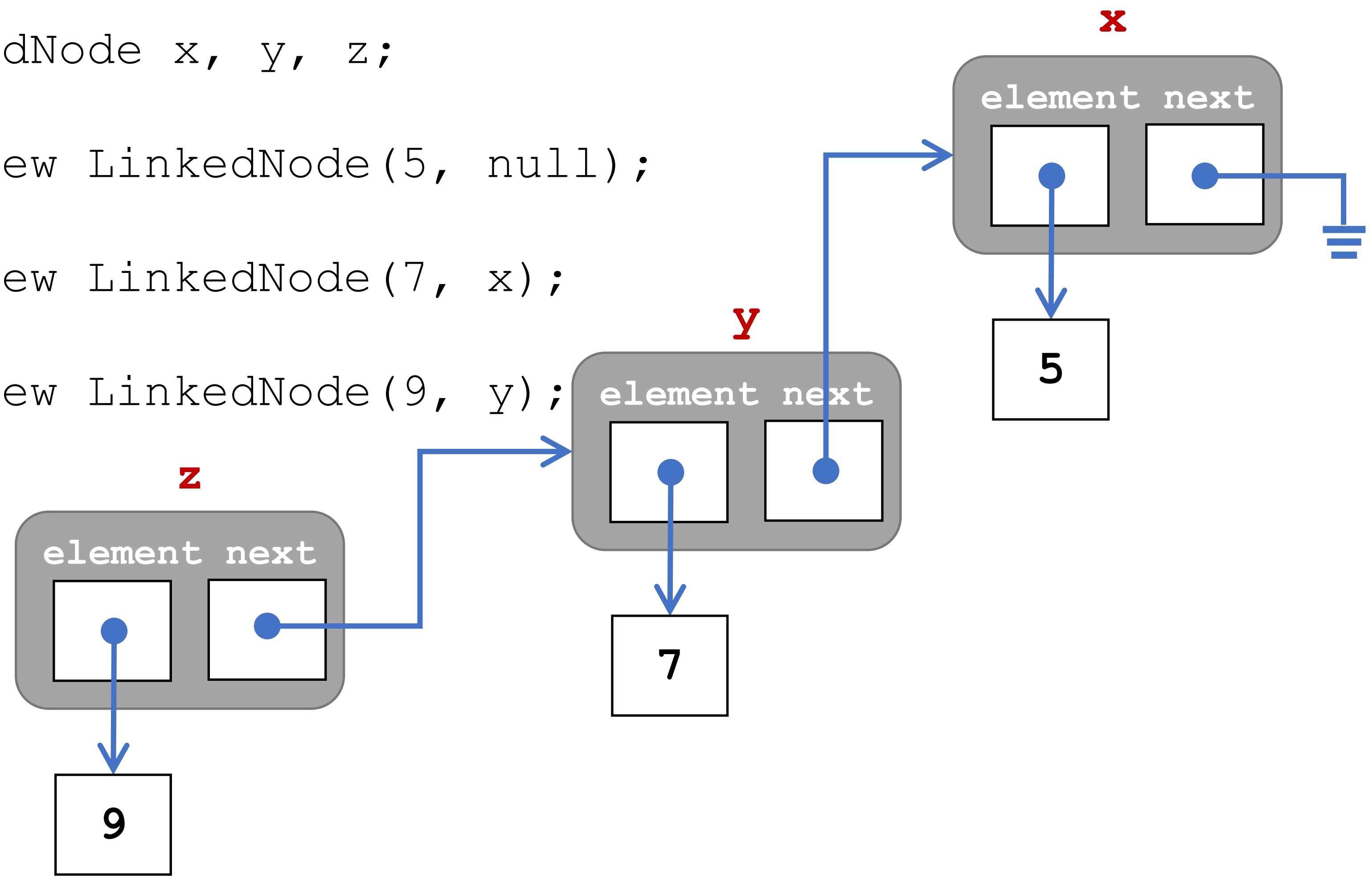
Linked Node

```
private static class LinkedNode {
   private Object element;
   private LinkedNode next;

   LinkedNode(Object e, LinkedNode n) {
      element = e;
      next = n;
   }
}
```



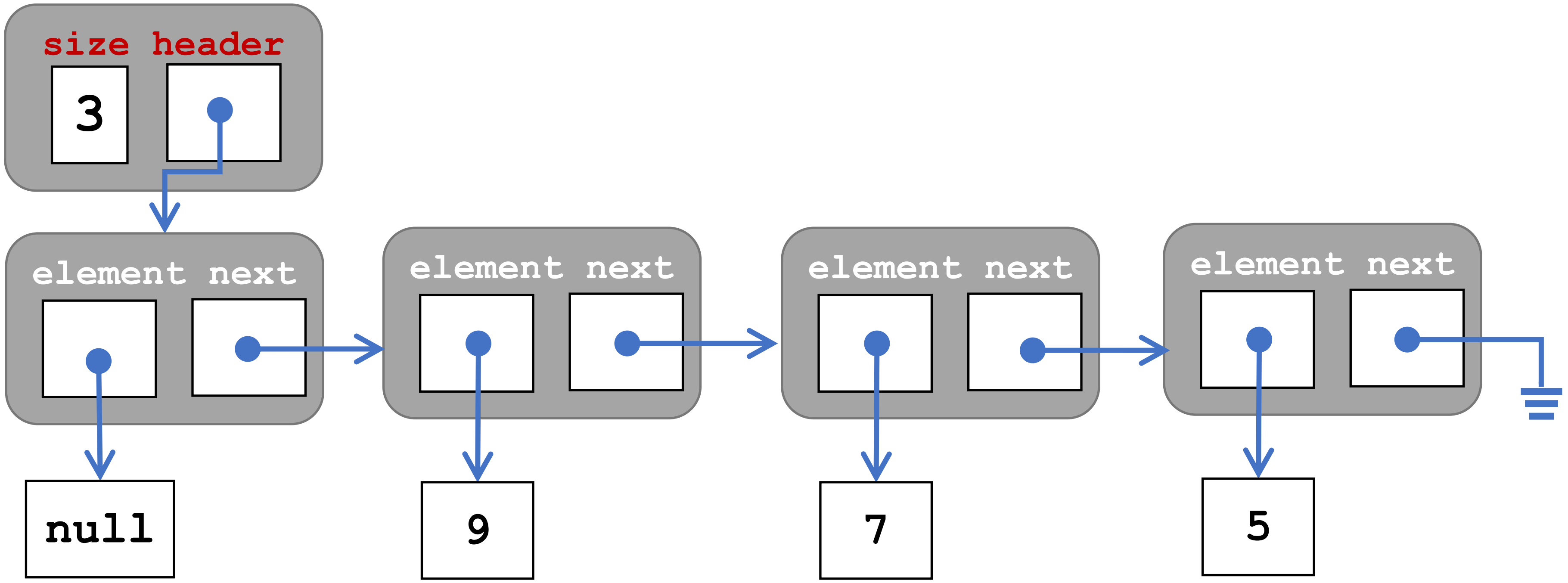
Linked Node

```
LinkedNode x, y, z;

x = new LinkedNode(5, null);

y = new LinkedNode(7, x);

z = new LinkedNode(9, y);
```



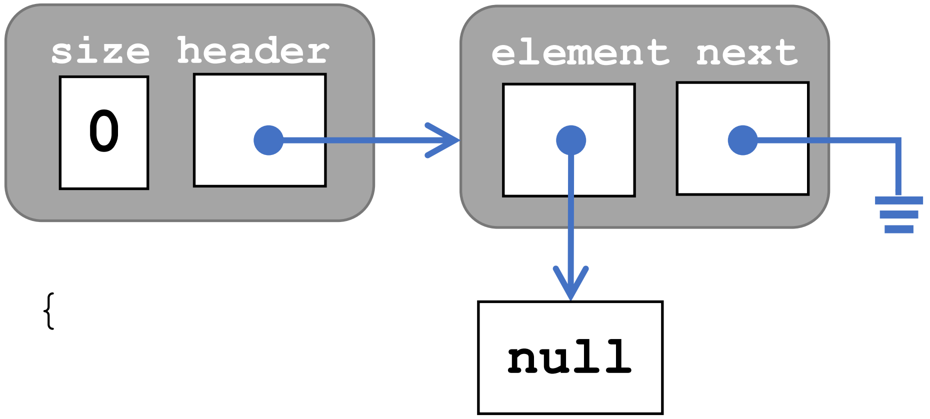
Class LinkedCollection

- Attribute ที่เก็บข้อมูลในชุด (`header`) ที่เป็นโหนดแรกในชุด
- Attribute ที่เก็บจำนวนข้อมูล (`size`) เป็น integer



Create new object: Class LinkedCollection

```
public class LinkedCollection implements Collection {
  private static class LinkedNode { ... }
  private int size;
  private LinkedNode header;

  public int size() {
    return size;
  }
  public LinkedCollection() {
    size = 0;
    header = new LinkedNode(null, null); }
   …
}
```



Methods in Class LinkedCollection

```
public class LinkedCollection implements Collection{
  private static class LinkedNode { ... }

  private int size;
  private LinkedNode header;

  public LinkedCollection() {…}     // 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 ?
  public void remove(Object e) {…} // remove e from object
  public boolean contains(Object e) {…} // is e in  object?
...}
```

Exercises

เขียน method `isEmpty`

Method add

add: Class LinkedCollection

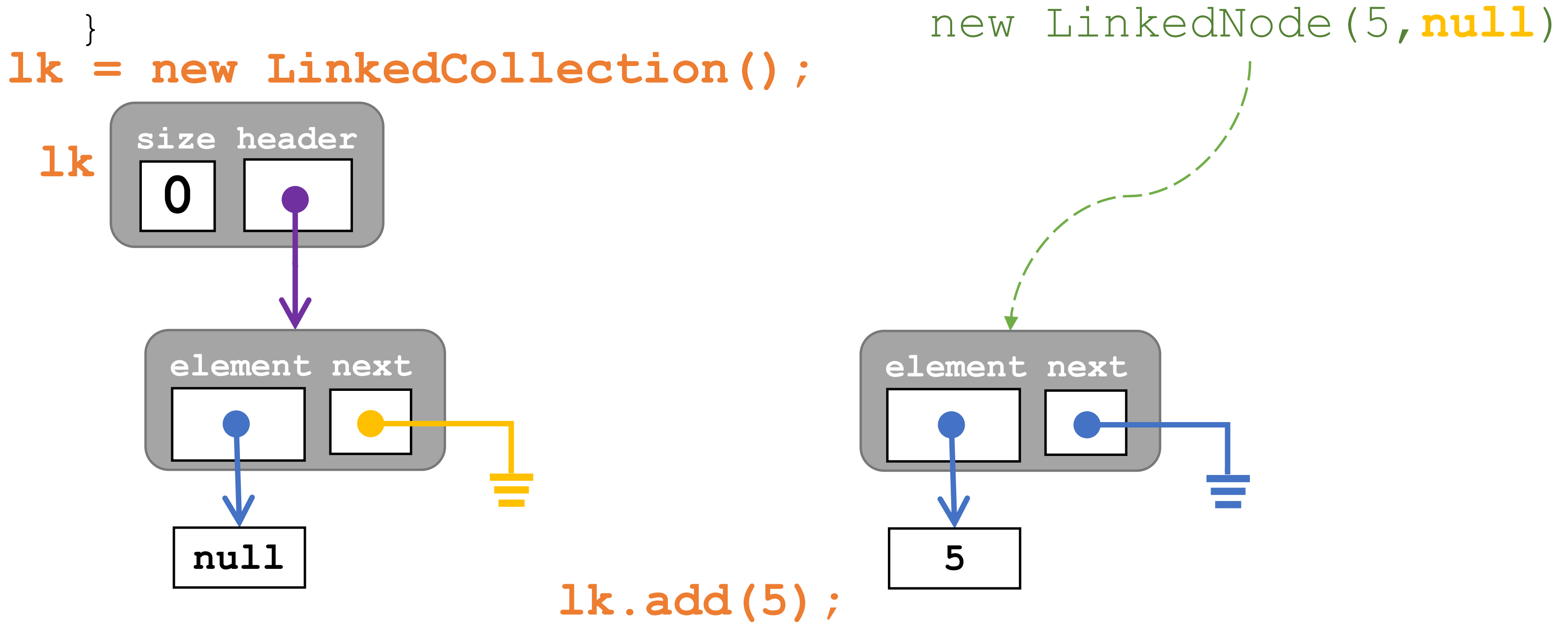
```
public void add(Object e) {
   if (e==null) throw new IllegalArgumentException();
   // create new node which links to the old first node.
   n = new LinkedNode(e, header.next);
   // set the new node as the first node
   header.next = n;
   size++;
}
public void add(Object e) {
   if (e==null) throw new IllegalArgumentException();
   header.next = new LinkedNode(e, header.next);
   size++;
}
```

Add a value in an empty collection

Example 1

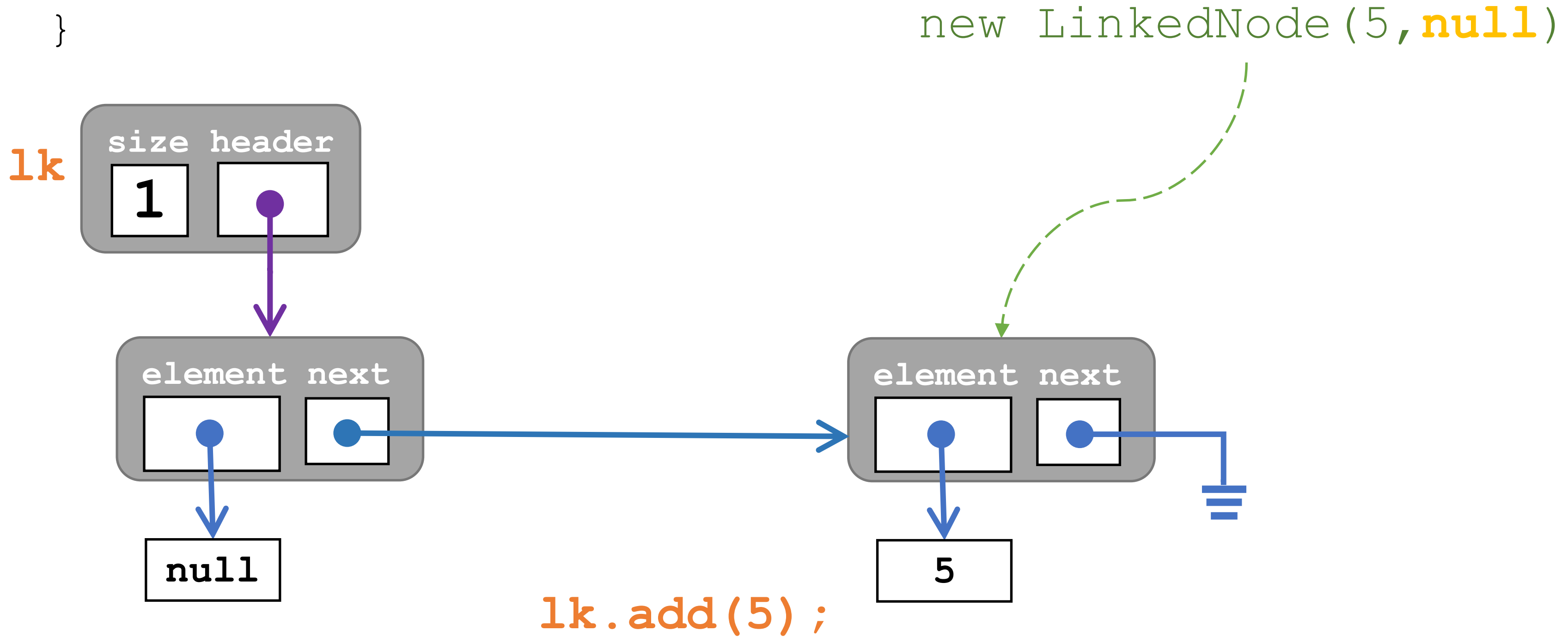
Example 1: add in an empty collection

```
public void add(Object e) {
    if (e==null) throw new IllegalArgumentException();
    header.next = new LinkedNode(e, header.next);
    size++;
}
```



Example 1: add in an empty collection

```
public void add(Object e) {
   if (e==null) throw new IllegalArgumentException();
   header.next = new LinkedNode(e, header.next);
   size++;
}
```

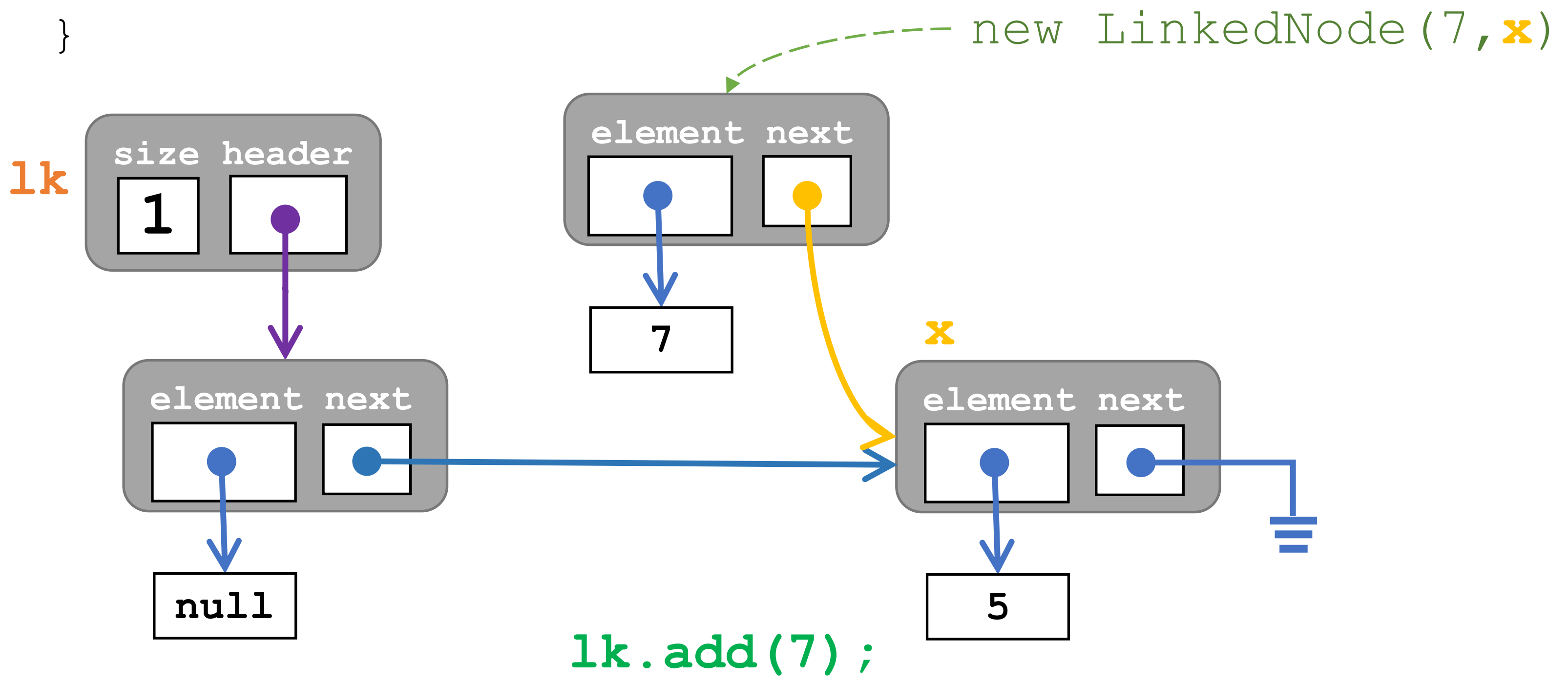


Add a value in a non-empty collection

Example 2

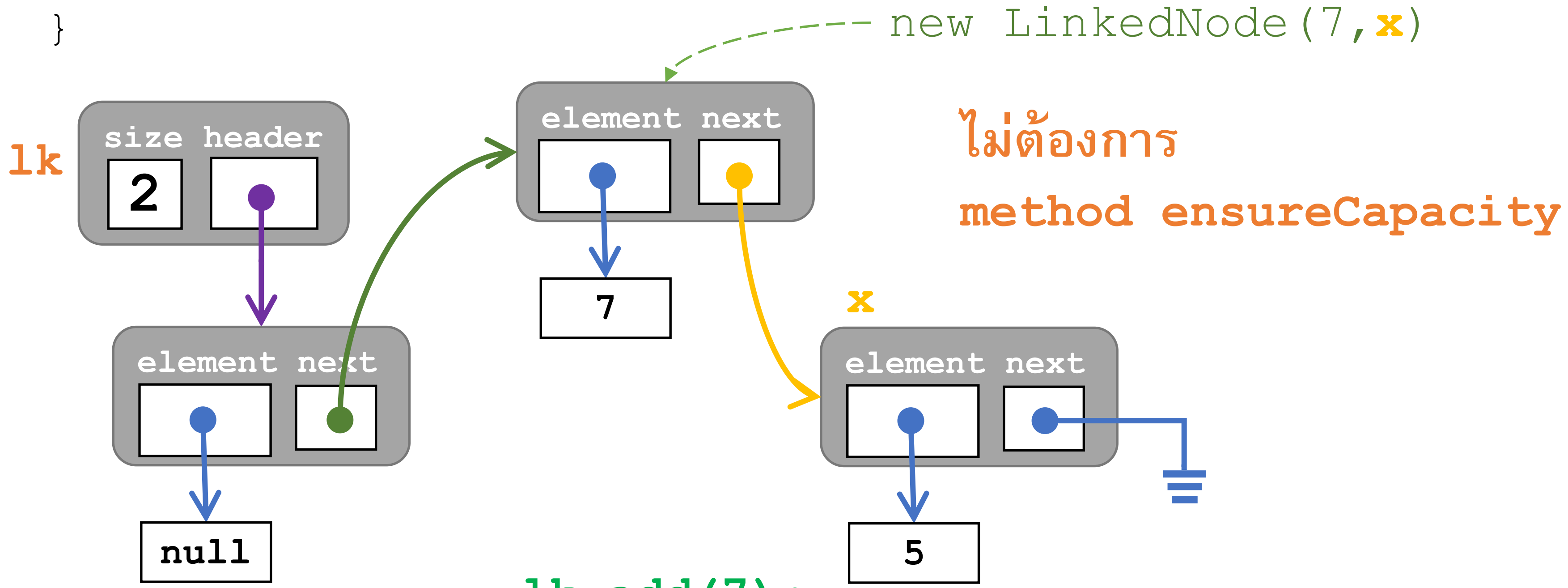
Example 2: add in a non-empty collection

```
public void add(Object e) {
    if (e==null) throw new IllegalArgumentException();
    header.next = new LinkedNode(e, header.next);
    size++;
}
```



Example 2: add in a non-empty collection

```
public void add(Object e) {
    if (e==null) throw new IllegalArgumentException();
    header.next = new LinkedNode(e, header.next);
    size++;
}
```



ไม่ต้องการ
method ensureCapacity

lk.add(7);

Method contains

contains: Class LinkedCollection

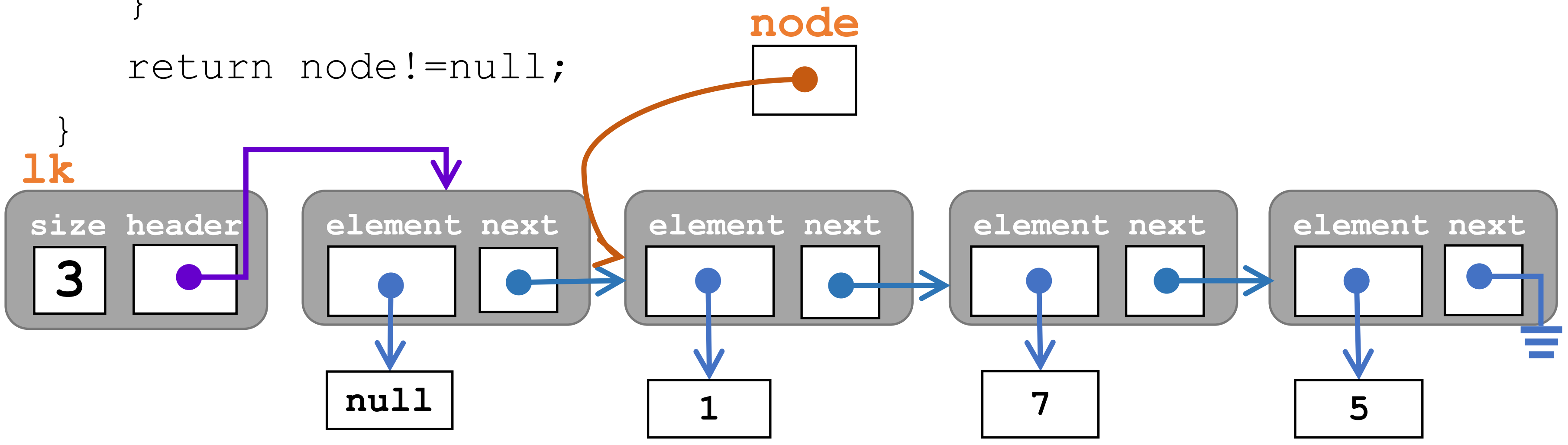
```
public boolean contains(Object e) {
    // the variable node points to the node to be checked
    LinkedNode node = header.next;
    while (node!=null && !node.element.equals(e)) {
        node = node.next;
    }
    // exit the loop when node is null or node.element is e
    // node is null means e is not in the collection
    return node!=null;
}
```

How contains works when the value is not in the collection

Example 1

Example 1 : `lk.contains(0);`

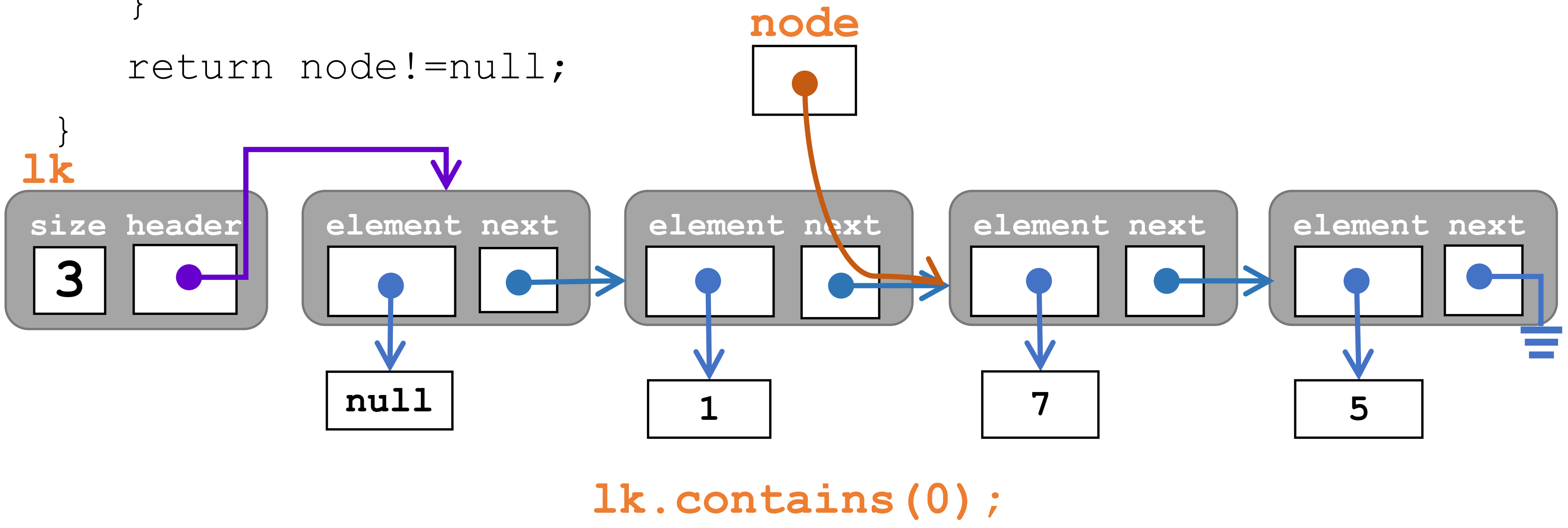
```
public boolean contains(Object e) {
    LinkedNode node = header.next;
    while (node!=null && !node.element.equals(e)) {
        node = node.next;
    }
    return node!=null;
}
```



lk.contains(0);

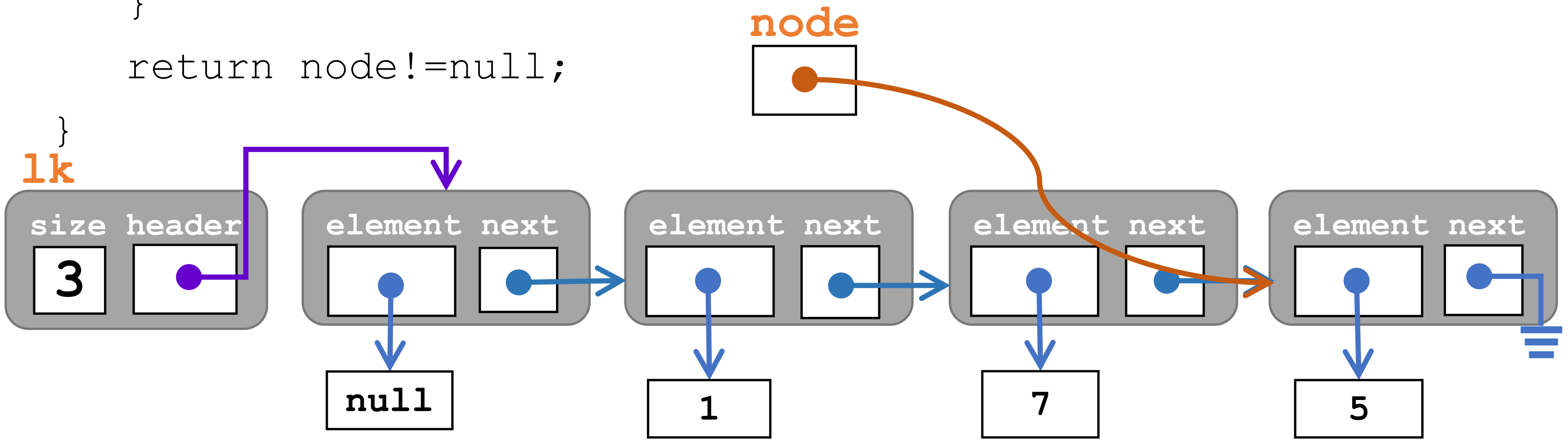
Example 1 : **`lk.contains(0);`**

```
public boolean contains(Object e) {
    LinkedNode node = header.next;
    while (node!=null && !node.element.equals(e)) {
        node = node.next;
    }
    return node!=null;
}
```



Example 1 : `lk.contains(0);`

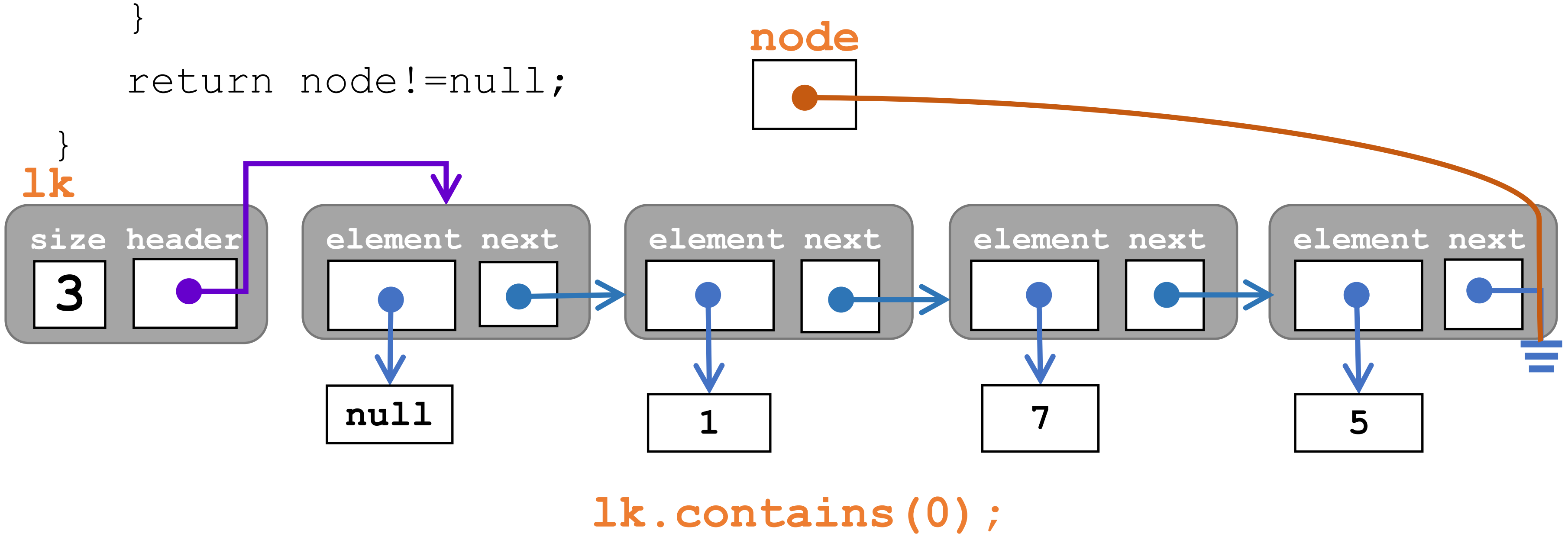
```
public boolean contains(Object e) {
    LinkedNode node = header.next;
    while (node!=null && !node.element.equals(e)) {
        node = node.next;
    }
    return node!=null;
}
```



`lk.contains(0);`

Example 1 : **`lk.contains(0);`**

```
public boolean contains(Object e) {
    LinkedNode node = header.next;
    while (node!=null && !node.element.equals(e)) {
        node = node.next;
    }
    return node!=null;
}
```

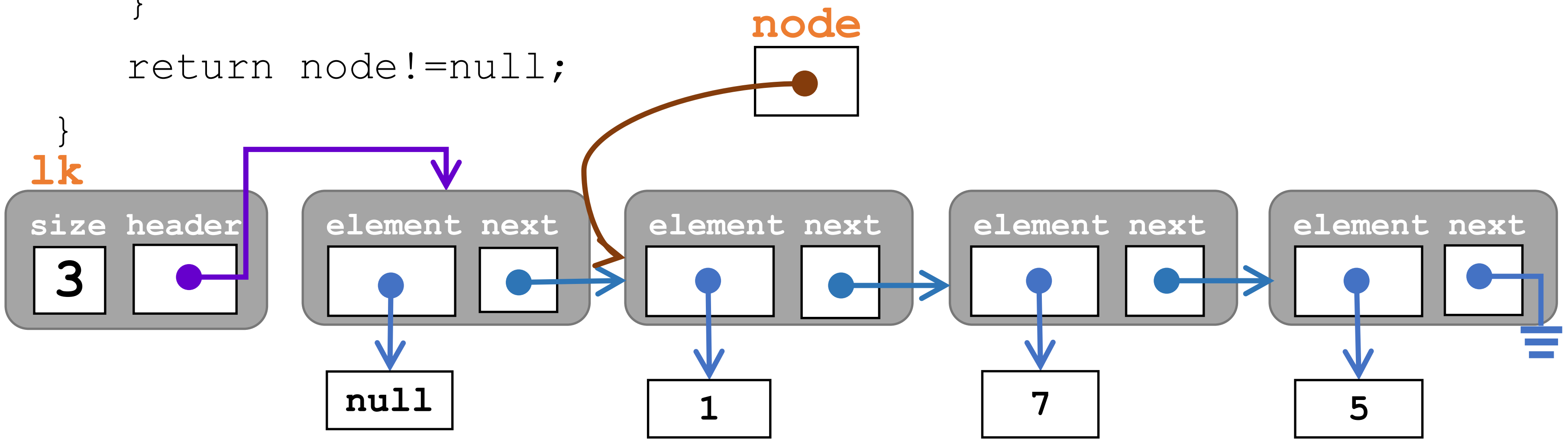


How contains works when the value is in the collection

Example 2

Example 2 : `lk.contains(7);`

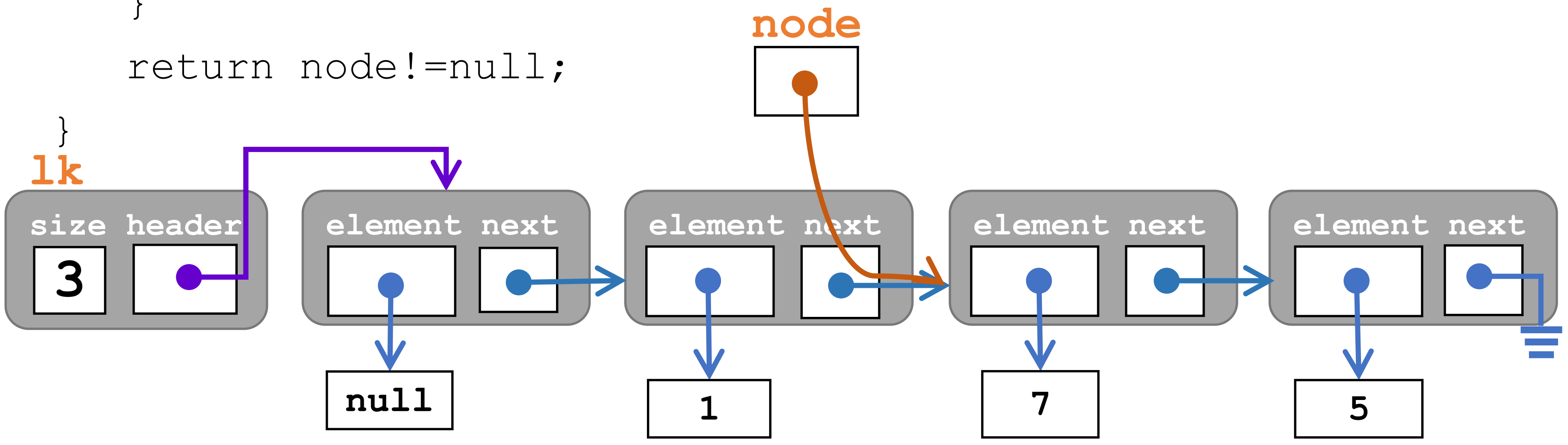
```
public boolean contains(Object e) {
    LinkedNode node = header.next;
    while (node!=null && !node.element.equals(e)) {
        node = node.next;
    }
    return node!=null;
}
```



lk.contains(7);

Example 2 : `lk.contains(7);`

```
public boolean contains(Object e) {
    LinkedNode node = header.next;
    while (node!=null && !node.element.equals(e)) {
        node = node.next;
    }
    return node!=null;
}
```

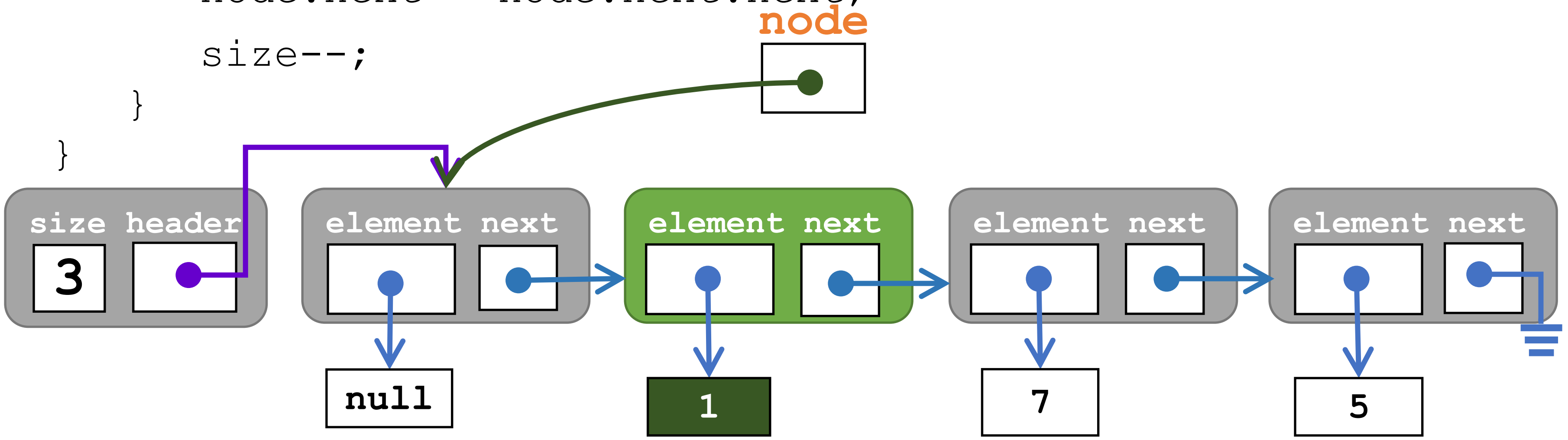


`lk.contains(7);`

Method remove

remove: Class LinkedCollection

```
public void remove(Object e) {
    LinkedNode node = header;        // we look at node.next
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;            // look at the next node
    if (node.next!=null) {      // found the node to be removed
        node.next = node.next.next;
        size--;
    }
}
```

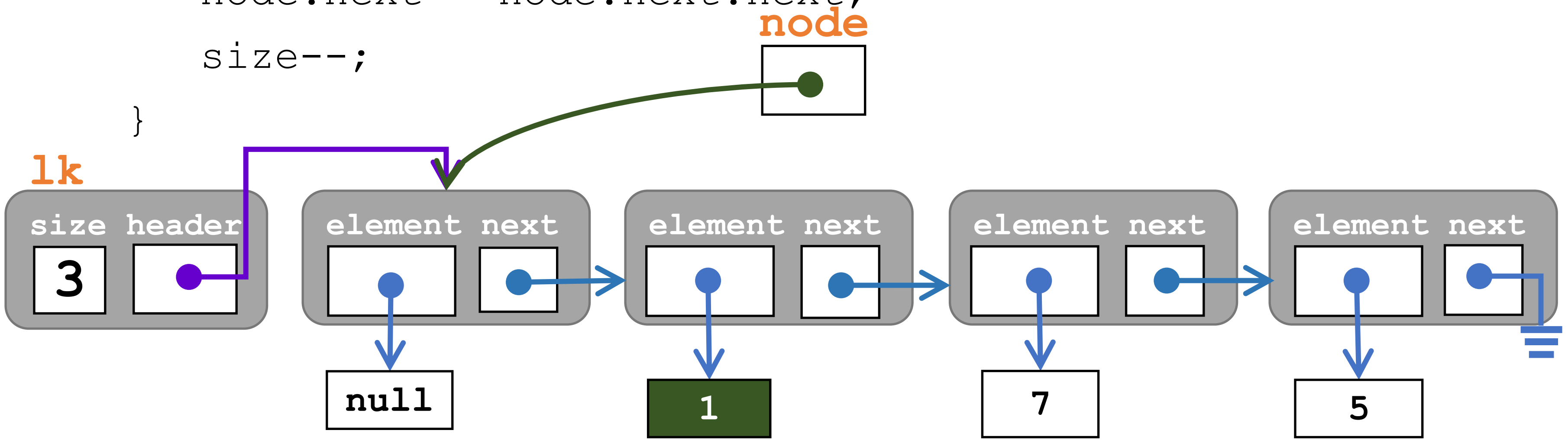


Remove the value which is in the collection

Example 1

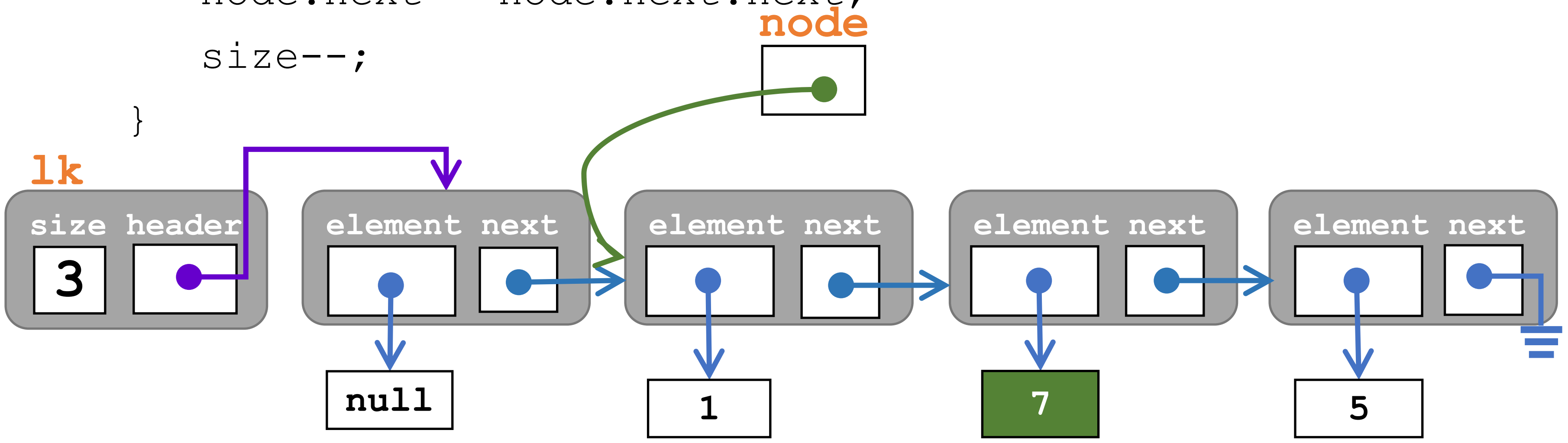
Example 1: `lk.remove(7);`

```
public void remove(Object e) {
    LinkedNode node = header;
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;
    if (node.next!=null) {
        node.next = node.next.next;
        size--;
    }
```



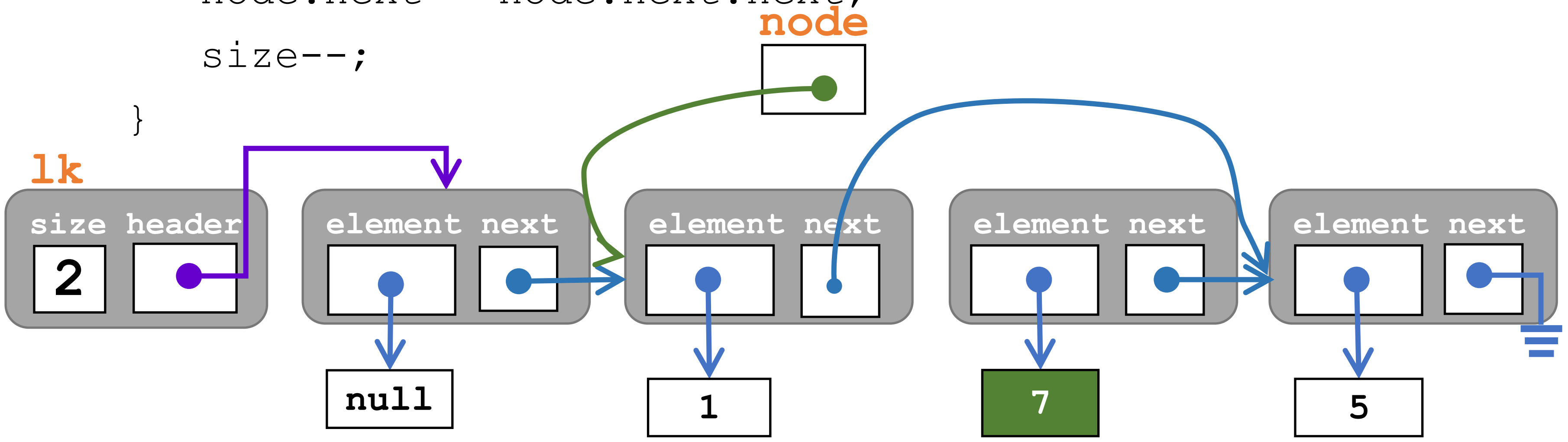
Example 1: `lk.remove(7);`

```
public void remove(Object e) {
    LinkedNode node = header;
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;
    if (node.next!=null) {
        node.next = node.next.next;
        size--;
    }
```



Example 1: `lk.remove(7);`

```
public void remove(Object e) {
   LinkedNode node = header;
   while (node.next!=null && !node.next.element.equals(e))
      node = node.next;
   if (node.next!=null) {
      node.next = node.next.next;
      size--;
   }
```



Exercises

method remove ให้ตัวชี้ node ไปที่โหนดก่อนโหนดที่จะตรวจสอบค่า

method contains ให้ตัวชี้ node ไปที่โหนดที่จะตรวจสอบค่าเลย

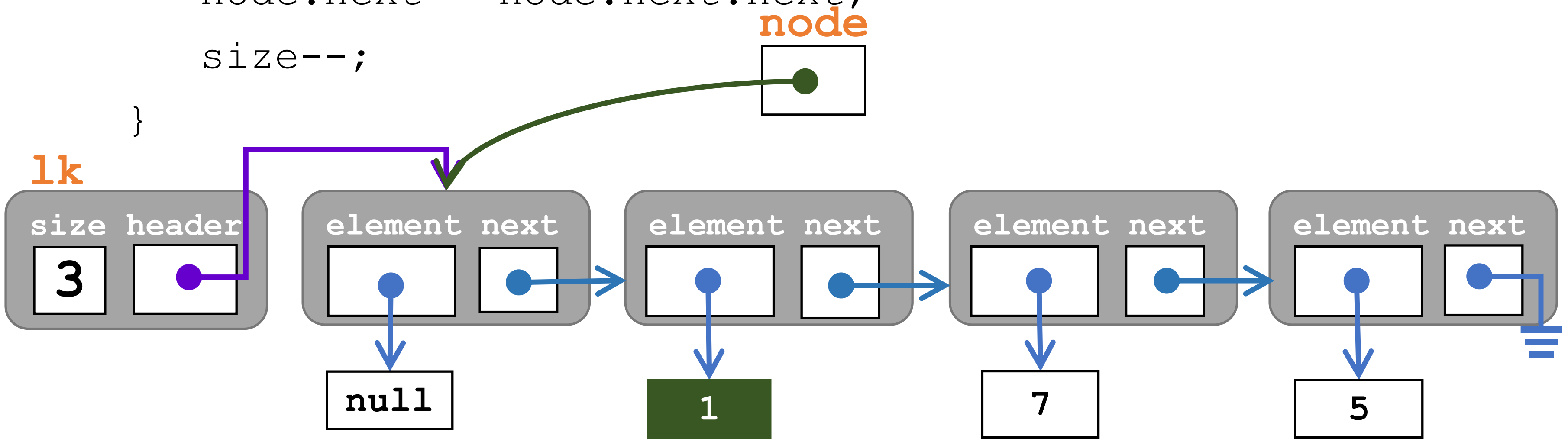
ทำไม ??

Remove the value which is not in the collection

Example 2

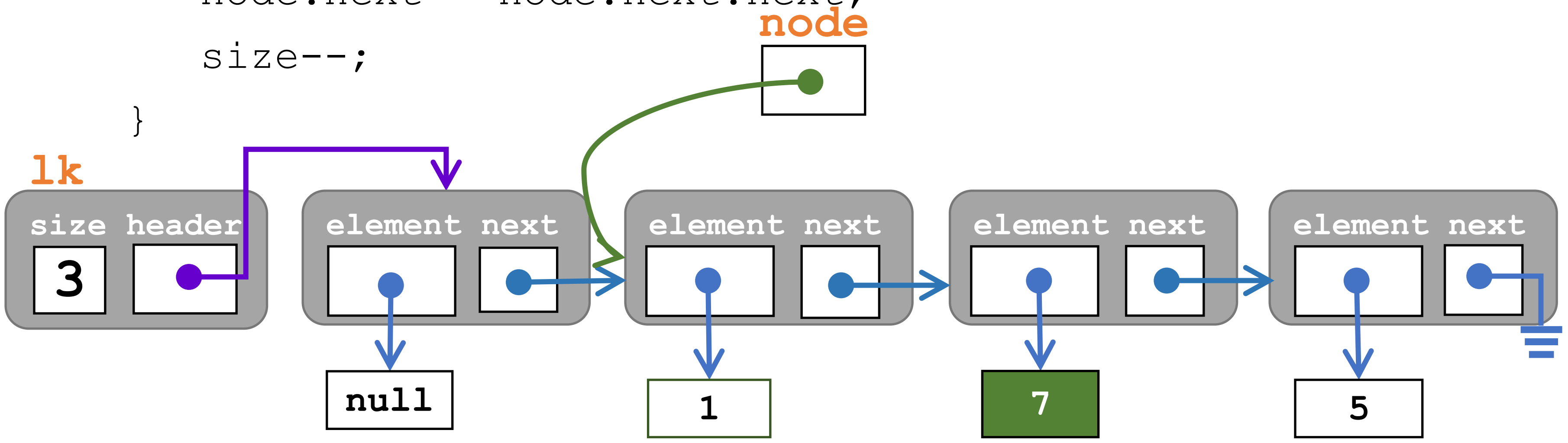
Example 2: `lk.remove(0);`

```
public void remove(Object e) {
    LinkedNode node = header;
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;
    if (node.next!=null) {
        node.next = node.next.next;
        size--;
    }
```



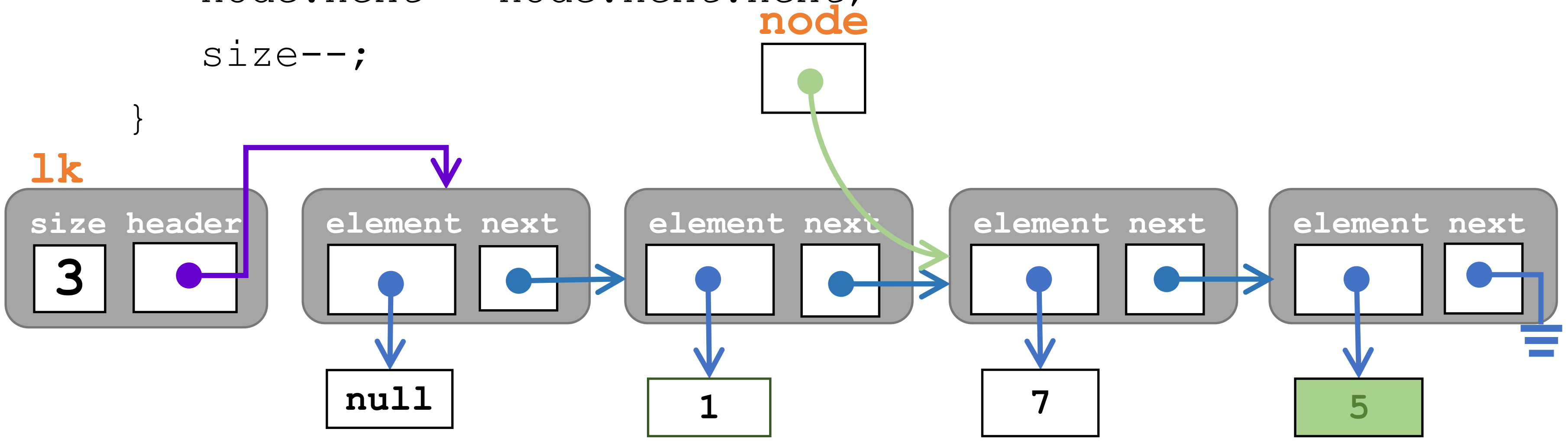
Example 2: `lk.remove(0);`

```
public void remove(Object e) {
    LinkedNode node = header;
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;
    if (node.next!=null) {
        node.next = node.next.next;
        size--;
    }
```



Example 2: `lk.remove(0);`

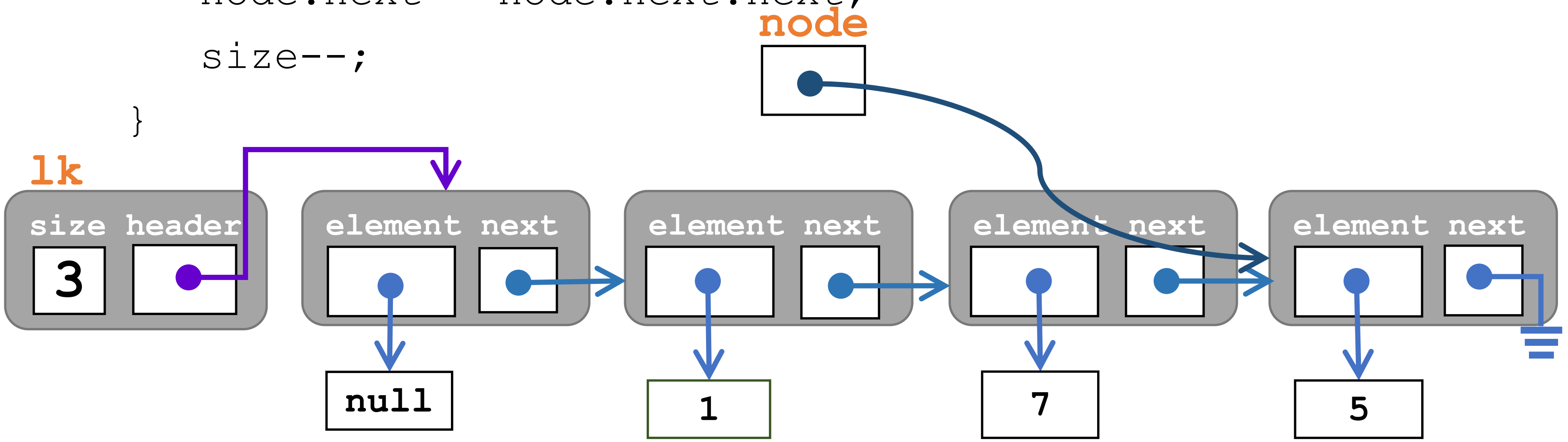
```
public void remove(Object e) {
    LinkedNode node = header;
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;
    if (node.next!=null) {
        node.next = node.next.next;
        size--;
    }
```



Example 2: `lk.remove(0);`

```
public void remove(Object e) {
    LinkedNode node = header;
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;
    if (node.next!=null) {
        node.next = node.next.next;
        size--;
    }
```

ไม่ทำ

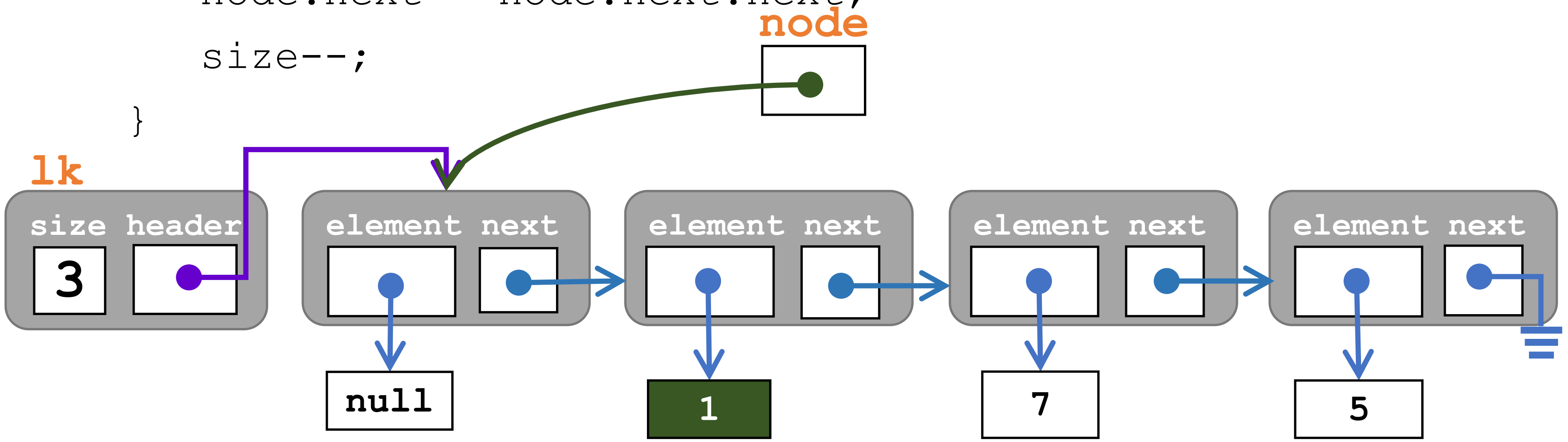


Remove the first node

Example 3

Example 3: Remove โหนดแรก

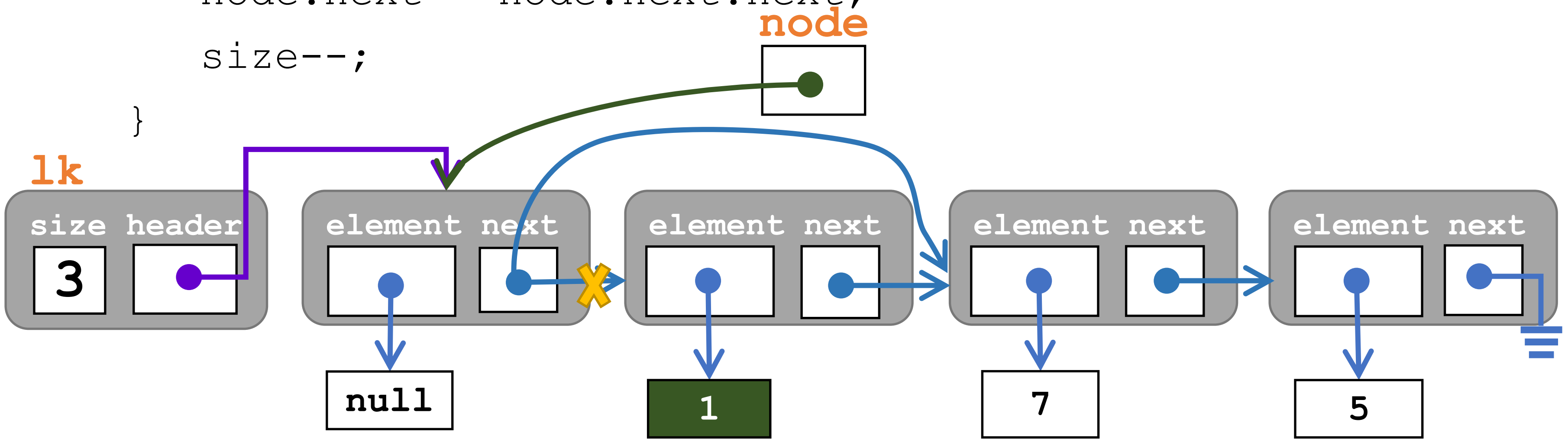
```
public void remove(Object e) {
    LinkedNode node = header;
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;
    if (node.next!=null) {
        node.next = node.next.next;
        size--;
    }
```



lk.remove(1);

Example 3: Remove โหนดแรก

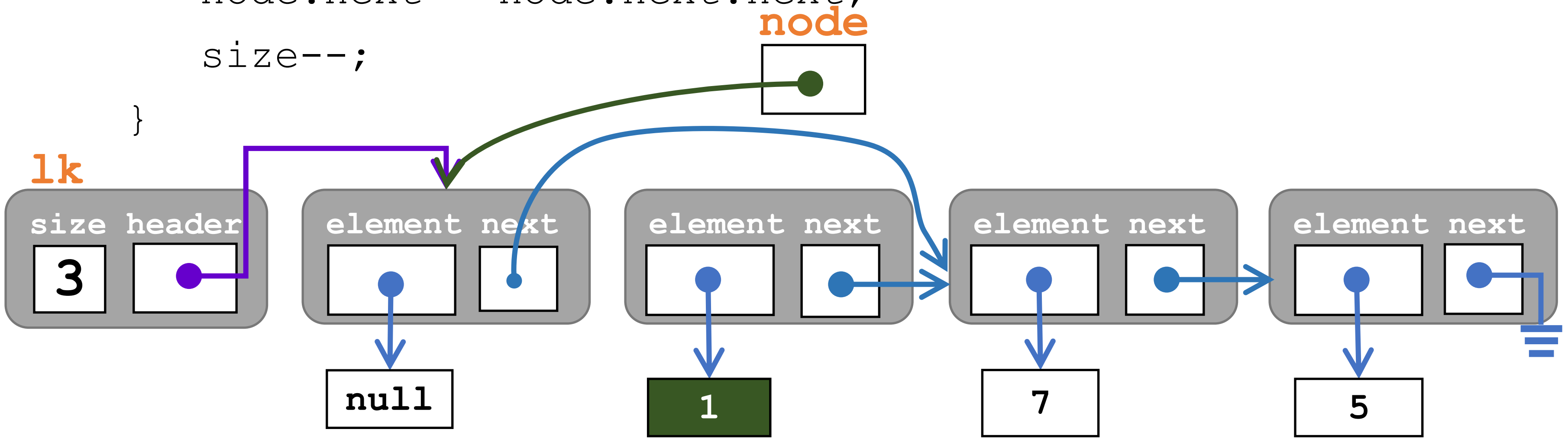
```
public void remove(Object e) {
    LinkedNode node = header;
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;
    if (node.next!=null) {
        node.next = node.next.next;
        size--;
    }
```



lk.remove(1);

Example 3: Remove โหนดแรก

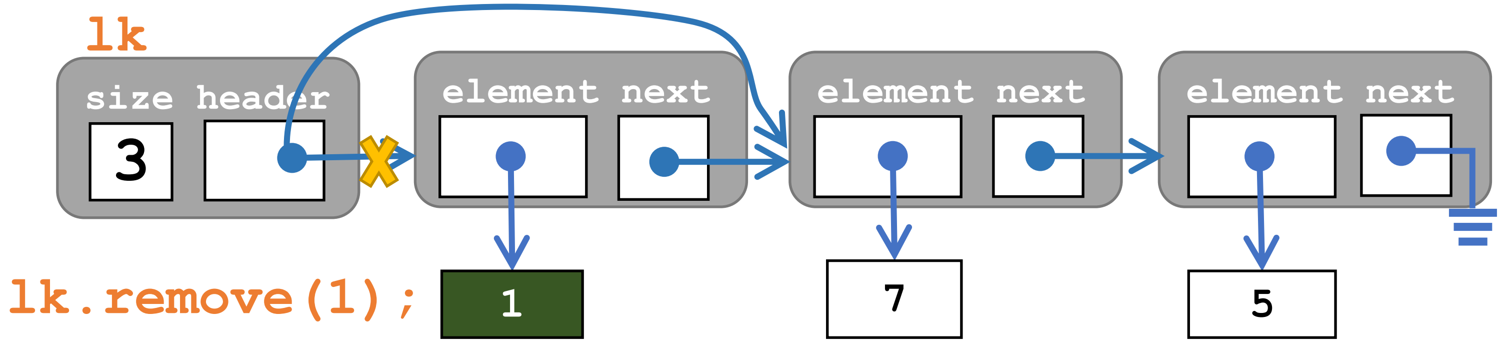
```
public void remove(Object e) {
    LinkedNode node = header;
    while (node.next!=null && !node.next.element.equals(e))
        node = node.next;
    if (node.next!=null) {
        node.next = node.next.next;
        size--;
    }
```



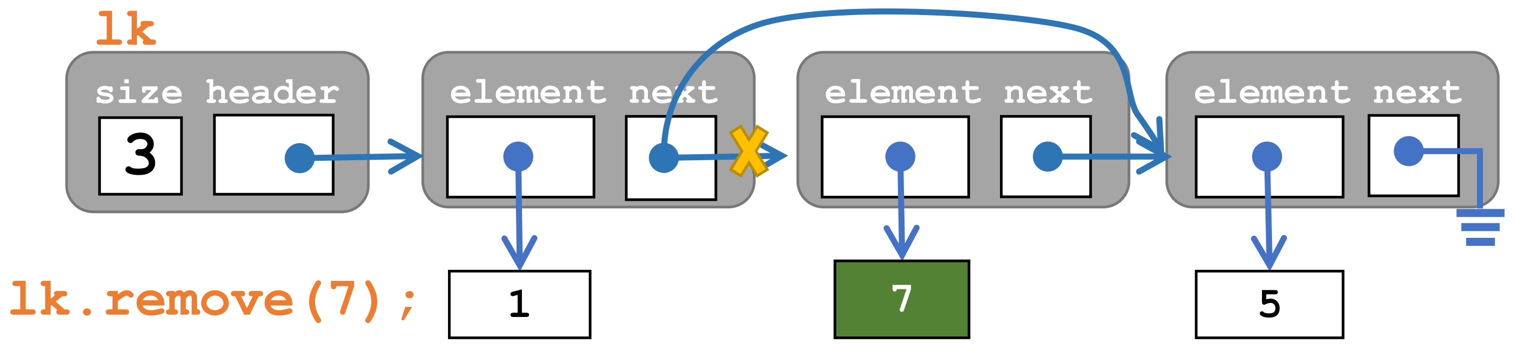
lk.remove(1);

กรณีที่ไม่มีโหนดแรกที่เก็บ null

- หาก remove โหนดแรก ต้องแก้ header ของ LinkedCollection



- หาก remove โหนดอื่น ต้องแก้ next ของ LinkedNode ก่อนหน้า



ทำให้ต้องเขียนโปรแกรมทำงานกับ 2 กรณีนี้แยกกัน

remove: ไม่มีโหนดแรกที่เก็บ null

```
public void remove(Object e) {     // more difficult
   if (header==null) return;
   if (header.element.equals(e) {  // remove header
      header = header.next;   size--;
   } else {                        // remove non-header
      LinkedNode node = header;
      while (node.next!=null && !node.next.element.equals(e))
         node = node.next;
      if (node.next!=null) {
         node.next = node.next.next;
         size--;
      }
   }
}
```

Exercises

- เขียน method `merge(a)` ที่รวม LinkedCollection `a` เข้ากับ LinkedCollection ที่กำหนด
- เขียน method `addAfter(a,b)` ที่เอาค่า `b` ใส่ต่อหลังค่า `a`