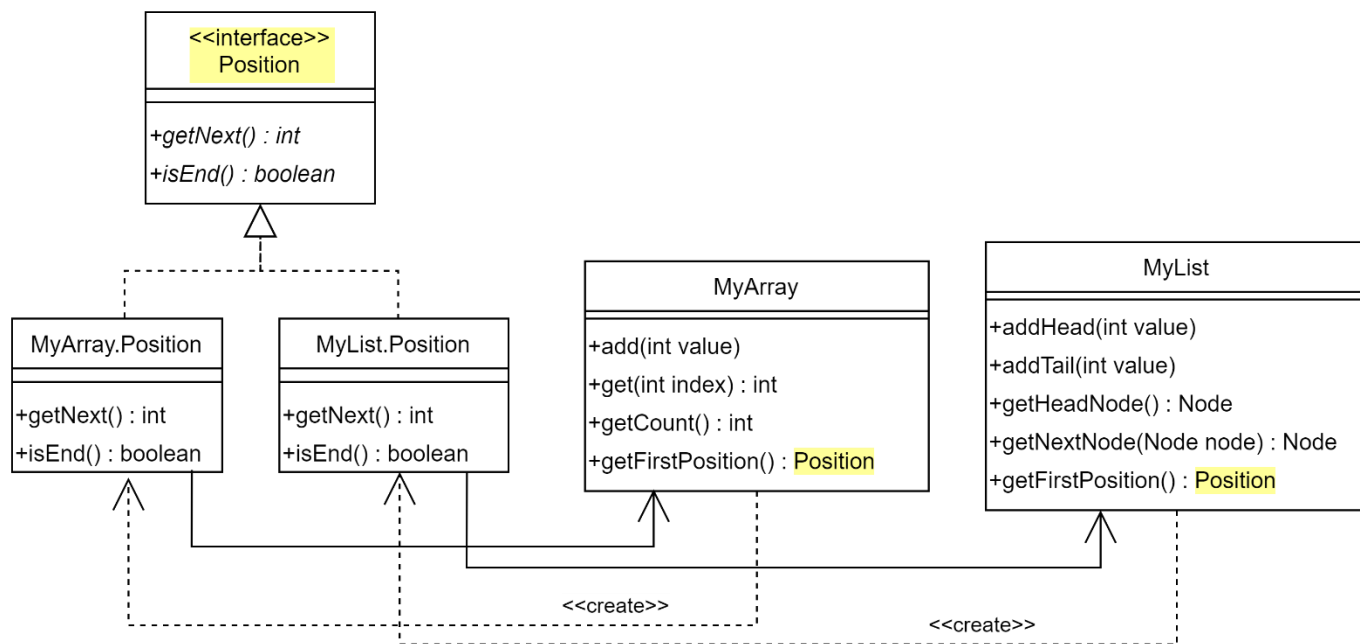


1) 개요

MyArray.Position, MyList.Position 클래스에 다형성 구현



2) Position.java

```
1 package iterator.e5;
2
3 public interface Position {
4     int getNext();
5     boolean isEnd();
6 }
```

3) MyArray.java

```
1 package iterator.e5;
2
3 import java.util.Arrays;
4
5 public class MyArray {
6     private int[] data;
7     private int count;
8
9     public MyArray() {
10         this(8);
11     }
12
13     public MyArray(int size) {
14         data = new int[size];
15         count = 0;
16     }
17
18     private void expand() {
19         data = Arrays.copyOf(data, data.length * 2);
20     }
21
22     public void add(int value) {
23         if (count == data.length) expand();
24         data[count++] = value;
25     }
26
27     public int get(int index) {
28         return data[index];
29     }
30
31     public int getCount() {
32         return count;
33     }
34
35     private class MyArrayPosition implements Position {
36         private int current;
37
38         public MyArrayPosition() {
39             current = 0;
40         }
41
42         @Override
43         public int getNext() {
44             return data[current++];
45         }
46
47         @Override
48         public boolean isEnd() {
49             return current >= count;
50         }
51     }
52
53     public Position getFirstPosition() {
54         return new MyArrayPosition();
55     }
56 }
```

4) MyList.java

```
1 package iterator.e5;
2
3 public class MyList {
4     private static class Node {
5         private int data;
6         private Node prev, next;
7
8         Node(int data) {
9             this.data = data;
10        }
11    }
12
13    private Node dummy;
14
15    public MyList() {
16        dummy = new Node(Integer.MIN_VALUE);
17        dummy.prev = dummy.next = dummy;
18    }
19
20    public void addHead(int value) {
21        Node node = new Node(value);
22        node.next = dummy.next;
23        node.prev = dummy;
24        dummy.next.prev = node;
25        dummy.next = node;
26    }
27
28    public void addTail(int value) {
29        Node node = new Node(value);
30        node.next = dummy;
31        node.prev = dummy.prev;
32        dummy.prev.next = node;
33        dummy.prev = node;
34    }
35
36    private class MyListPosition implements Position {
37        private Node current;
38
39        public MyListPosition() {
40            current = dummy.next;
41        }
42
43        @Override
44        public int getNext() {
45            int r = current.data;
46            current = current.next;
47            return r;
48        }
49
50        @Override
51        public boolean isEnd() {
52            return current == dummy;
53        }
54    }
55
56    public Position getFirstPosition() {
57        return new MyListPosition();
58    }
59 }
```

5) Example5.java

```
1 package iterator.e5;
2
3 public class Example5 {
4
5     static void print(Position pos) {
6         while (!pos.isEnd())
7             System.out.printf("%d ", pos.getNext());
8         System.out.println();
9     }
10
11    static void doSomething1(int count) {
12        MyArray a = new MyArray();
13        for (int i = 0; i < count; ++i)
14            a.add(i);
15
16        print(a.getFirstPosition());
17    }
18
19    static void doSomething2(int count) {
20        MyList list = new MyList();
21        for (int i = 0; i < count; ++i)
22            list.addTail(i);
23
24        print(list.getFirstPosition());
25    }
26
27    public static void main(String[] args) {
28        doSomething1(10);
29        doSomething2(10);
30    }
31 }
```

출력

```
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
```

다형성 구현의 효과는?