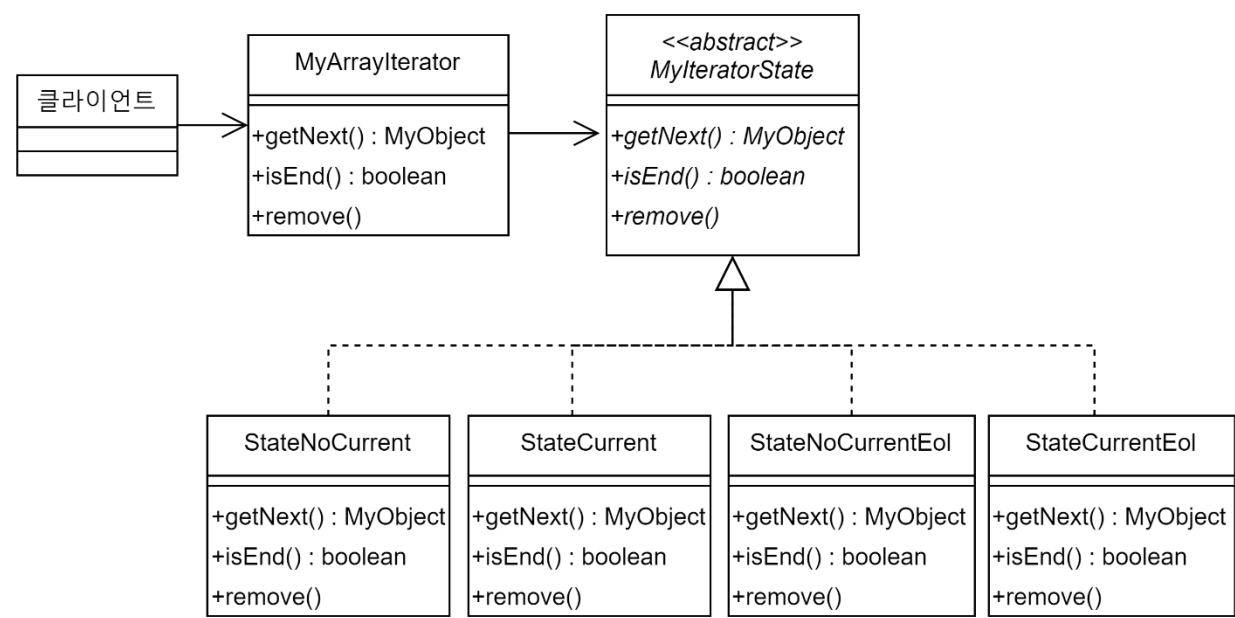


# 목차

1) 개요 .....	1
2) MyCollection.java.....	2
3) MyIterator.java.....	2
4) MyArray.java.....	2
5) MyArrayIterator.java.....	3
6) MyIteratorState.java.....	3
7) StateCurrent.java.....	4
8) StateCurrentEol.java.....	5
9) StateNoCurrent.java.....	5
10) StateNoCurrentEol.java.....	6
11) Example6.java.....	7

## 1) 개요



## 2) MyCollection.java

```
1 package state.e6;
2
3 public interface MyCollection {
4     void add(int value);
5     MyIterator getIterator();
6 }
```

## 3) MyIterator.java

```
1 package state.e6;
2
3 public interface MyIterator {
4     int getNext();
5     boolean isEnd();
6     void remove();
7 }
```

## 4) MyArray.java

```
1 package state.e6;
2
3 import java.util.Arrays;
4
5 public class MyArray implements MyCollection {
6     int[] data;
7     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     @Override
23     public void add(int value) {
24         if (count == data.length) expand();
25         data[count++] = value;
26     }
27
28     public int get(int index) {
29         return data[index];
30     }
31
32     public int getCount() {
33         return count;
34     }
35
36     @Override
37     public MyIterator getIterator() {
38         return new MyArrayIterator(this);
39     }
40 }
```

## 5) MyArrayIterator.java

```
1 package state.e6;
2
3 public class MyArrayIterator implements MyIterator {
4
5     MyArray myArray;
6     int current;
7     MyIteratorState state;
8
9     StateCurrent stateCurrent = new StateCurrent(this);
10    StateCurrentEol stateCurrentEol = new StateCurrentEol(this);
11    StateNoCurrent stateNoCurrent = new StateNoCurrent(this);
12    StateNoCurrentEol stateNoCurrentEol = new StateNoCurrentEol(this);
13
14    public MyArrayIterator(MyArray myArray) {
15        this.myArray = myArray;
16        this.current = 0;
17        this.state = myArray.getCount() > 0 ? stateNoCurrent : stateNoCurrentEol;
18    }
19
20    @Override
21    public int getNext() {
22        return state.getNext();
23    }
24
25    @Override
26    public boolean isEnd() {
27        return state.isEnd();
28    }
29
30    @Override
31    public void remove() {
32        state.remove();
33    }
34 }
```

## 6) MyIteratorState.java

```
1 package state.e6;
2
3
4 public abstract class MyIteratorState {
5
6     MyArrayIterator wrapper;
7
8     public MyIteratorState(MyArrayIterator myArrayIterator) {
9         this.wrapper = myArrayIterator;
10    }
11
12    abstract int getNext();
13    abstract boolean isEnd();
14    abstract void remove();
15 }
```

## 7) StateCurrent.java

```
1 package state.e6;
2
3 public class StateCurrent extends MyIteratorState {
4
5     public StateCurrent(MyArrayIterator myArrayIterator) {
6         super(myArrayIterator);
7     }
8
9     @Override
10    public int getNext() {
11        if (wrapper.current < wrapper.myArray.count - 1)
12            wrapper.state = wrapper.stateCurrent;
13        else
14            wrapper.state = wrapper.stateCurrentEol;
15        return wrapper.myArray.data[wrapper.current++];
16    }
17
18    @Override
19    public boolean isEnd() {
20        return false;
21    }
22
23    @Override
24    public void remove() {
25        wrapper.state = wrapper.stateNoCurrent;
26        --wrapper.current;
27        --wrapper.myArray.count;
28        for (int i = wrapper.current; i < wrapper.myArray.count; ++i)
29            wrapper.myArray.data[i] = wrapper.myArray.data[i + 1];
30    }
31 }
```

## 8) StateCurrentEol.java

```
1 package state.e6;
2
3 import java.util.NoSuchElementException;
4
5 public class StateCurrentEol extends MyIteratorState {
6
7     public StateCurrentEol(MyArrayIterator myArrayIterator) {
8         super(myArrayIterator);
9     }
10
11     @Override
12     public int getNext() {
13         throw new NoSuchElementException();
14     }
15
16     @Override
17     public boolean isEnd() {
18         return true;
19     }
20
21     @Override
22     public void remove() {
23         wrapper.state = wrapper.stateNoCurrentEol;
24         --wrapper.current;
25         --wrapper.myArray.count;
26         for (int i = wrapper.current; i < wrapper.myArray.count; ++i)
27             wrapper.myArray.data[i] = wrapper.myArray.data[i + 1];
28     }
29 }
```

## 9) StateNoCurrent.java

```
1 package state.e6;
2
3 import java.util.NoSuchElementException;
4
5 public class StateNoCurrent extends MyIteratorState {
6
7     public StateNoCurrent(MyArrayIterator myArrayIterator) {
8         super(myArrayIterator);
9     }
10
11     @Override
12     public int getNext() {
13         if (wrapper.current < wrapper.myArray.count - 1)
14             wrapper.state = wrapper.stateCurrent;
15         else
16             wrapper.state = wrapper.stateCurrentEol;
17         return wrapper.myArray.data[wrapper.current++];
18     }
19
20     @Override
21     public boolean isEnd() {
22         return false;
23     }
24
25     @Override
26     public void remove() {
27         throw new NoSuchElementException();
28     }
29 }
```

## 10) StateNoCurrentEol.java

```
1 package state.e6;
2
3 import java.util.NoSuchElementException;
4
5 public class StateNoCurrentEol extends MyIteratorState {
6
7     public StateNoCurrentEol(MyArrayIterator myArrayIterator) {
8         super(myArrayIterator);
9     }
10
11     @Override
12     public int getNext() {
13         throw new NoSuchElementException();
14     }
15
16     @Override
17     public boolean isEnd() {
18         return true;
19     }
20
21     @Override
22     public void remove() {
23         throw new NoSuchElementException();
24     }
25 }
```

## 11) Example6.java

```
1 package state.e6;
2
3 import java.util.function.Predicate;
4
5 public class Example6 {
6
7     static void add(MyCollection col, int count) {
8         for (int i = 0; i < count; ++i)
9             col.add(i);
10    }
11
12    static void print(MyIterator it) {
13        while (!it.isEnd())
14            System.out.printf("%d ", it.getNext());
15        System.out.println();
16    }
17
18    // predicate를 만족하는 항목들을 제거한다
19    static void remove(MyIterator it, Predicate<Integer> predicate) {
20        while (!it.isEnd())
21            if (predicate.test(it.getNext()))
22                it.remove();
23    }
24
25    static void doSomething1(MyCollection col) {
26        add(col, 10);
27        print(col.getIterator());
28        remove(col.getIterator(), value -> value < 5); // 5 미만 제거
29        print(col.getIterator());
30        remove(col.getIterator(), value -> value > 5); // 5 초과 제거
31        print(col.getIterator());
32    }
33
34    static void doSomething2(MyCollection col) {
35        add(col, 10);
36        MyIterator it = col.getIterator();
37        it.remove();
38        print(col.getIterator());
39    }
40
41    public static void main(String[] args) {
42        doSomething1(new MyArray());
43        doSomething2(new MyArray());
44    }
45 }
```

실행

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

```
Exception in thread "main" java.util.NoSuchElementException
    at state.e6.StateNoCurrent.remove(StateNoCurrent.java:27)
    at state.e6.MyArrayIterator.remove(MyArrayIterator.java:32)
    at state.e6.Example6.doSomething2(Example6.java:37)
    at state.e6.Example6.main(Example6.java:43)
```