

목차

1) MyCollection.java.....	2
2) MyIterator.java.....	2
3) MyArray.java.....	2
4) MyArrayIterator.java.....	3
5) Example5.java.....	5

1) MyCollection.java

```
1 package state.e5;
2
3 public interface MyCollection {
4     void add(int value);
5     MyIterator getIterator();
6 }
```

2) MyIterator.java

```
1 package state.e5;
2
3 public interface MyIterator {
4     int getNext();
5     boolean isEnd();
6     void remove();
7 }
```

3) MyArray.java

```
1 package state.e5;
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 }
```

4) MyArrayIterator.java

```
1 package state.e5;
2
3 import java.util.NoSuchElementException;
4
5 public class MyArrayIterator implements MyIterator {
6
7     enum IteratorState {
8         NO_CURRENT, CURRENT, NO_CURRENT_EOL, CURRENT_EOL
9     }
10
11     MyArray myArray;
12     int current;
13     IteratorState state;
14
15     public MyArrayIterator(MyArray myArray) {
16         this.myArray = myArray;
17         this.current = 0;
18         this.state = myArray.getCount() > 0 ?
19             IteratorState.NO_CURRENT : IteratorState.NO_CURRENT_EOL;
20     }
21
22     @Override
23     public int getNext() {
24         switch (state) {
25             case NO_CURRENT:
26             case CURRENT:
27                 if (current < myArray.count - 1)
28                     state = IteratorState.CURRENT;
29                 else
30                     state = IteratorState.CURRENT_EOL;
31                 break;
32             case NO_CURRENT_EOL:
33             case CURRENT_EOL:
34                 throw new NoSuchElementException();
35             }
36         return myArray.data[current++];
37     }
38
39     @Override
40     public boolean isEnd() {
41         boolean result = false;
42         switch (state) {
43             case NO_CURRENT:
44             case CURRENT:
45                 result = false;
46                 break;
47             case NO_CURRENT_EOL:
48             case CURRENT_EOL:
49                 result = true;
50                 break;
51             }
52         return result;
53     }
54
55     @Override
56     public void remove() {
57         switch (state) {
58             case NO_CURRENT:
59             case NO_CURRENT_EOL:
60                 throw new NoSuchElementException();
61             case CURRENT:
62                 state = IteratorState.NO_CURRENT;
63                 break;
64             case CURRENT_EOL:
65                 state = IteratorState.NO_CURRENT_EOL;
66                 break;
67             }
68         --current;
```

```
69     --myArray.count;
70     for (int i = current; i < myArray.count; ++i)
71         myArray.data[i] = myArray.data[i + 1];
72     }
73 }
```

안전한 구현

5) Example5.java

```
1 package state.e5;
2
3 import java.util.function.Predicate;
4
5 public class Example5 {
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.e5.MyArrayIterator.remove(MyArrayIterator.java:60)
    at state.e5.Example5.doSomething2(Example5.java:37)
    at state.e5.Example5.main(Example5.java:43)
```