목차

1) MyCollection.java

```java
package state.e4;

public interface MyCollection {
    void add(int value);
    MyIterator getIterator();
}
```

2) MyIterator.java

```java
package state.e4;

public interface MyIterator {
    int getNext();
    boolean isEnd();
    void remove();
}
```

3) MyArray.java

```java
package state.e4;

import java.util.Arrays;

public class MyArray implements MyCollection {
    int[] data;
    int count;

    public MyArray() {
        this(8);
    }

    public MyArray(int size) {
        data = new int[size];
        count = 0;
    }

    private void expand() {
        data = Arrays.copyOf(data, data.length * 2);
    }

    @Override
    public void add(int value) {
        if (count == data.length) expand();
        data[count++] = value;
    }

    public int get(int index) {
        return data[index];
    }

    public int getCount() {
        return count;
    }

    @Override
    public MyIterator getIterator() {
        return new MyArrayIterator(this);
    }
}
```

4) MyArrayIterator.java

```java
package state.e4;

class MyArrayIterator implements MyIterator {
    MyArray myArray;
    int current;

    public MyArrayIterator(MyArray myArray) {
        this.myArray = myArray;
        this.current = 0;
    }

    @Override
    public int getNext() {
        return myArray.data[current++];
    }

    @Override
    public boolean isEnd() {
        return current >= myArray.count;
    }

    @Override
    public void remove() {
        --current;
        --myArray.count;
        for (int i = current; i < myArray.count; ++i)
            myArray.data[i] = myArray.data[i + 1];
    }
}
```

안전하지 않은 구현

5) Example4.java

```java
package state.e4;

import java.util.function.Predicate;

public class Example4 {

    static void add(MyCollection col, int count) {
        for (int i = 0; i < count; ++i)
            col.add(i);
    }

    static void print(MyIterator it) {
        while (!it.isEnd())
            System.out.printf("%d ", it.getNext());
        System.out.println();
    }

    // predicate를 만족하는 항목들을 제거한다
    static void remove(MyIterator it, Predicate<Integer> predicate) {
        while (!it.isEnd())
            if (predicate.test(it.getNext()))
                it.remove();
    }

    static void doSomething1(MyCollection col) {
        add(col, 10);
        print(col.getIterator());
        remove(col.getIterator(), value -> value < 5); // 5 미만 제거
        print(col.getIterator());
        remove(col.getIterator(), value -> value > 5); // 5 초과 제거
        print(col.getIterator());
    }

    static void doSomething2(MyCollection col) {
        add(col, 10);
        MyIterator it = col.getIterator();
        it.remove();
        print(col.getIterator());
    }

    public static void main(String[] args) {
        doSomething1(new MyArray());
        doSomething2(new MyArray());
    }
}
```

실행

```
0 1 2 3 4 5 6 7 8 9
5 6 7 8 9
5
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index -1 out of bounds for length 16
        at state.e4.MyArrayIterator.remove(MyArrayIterator.java:27)
        at state.e4.Example4.doSomething2(Example4.java:37)
        at state.e4.Example4.main(Example4.java:43)
```