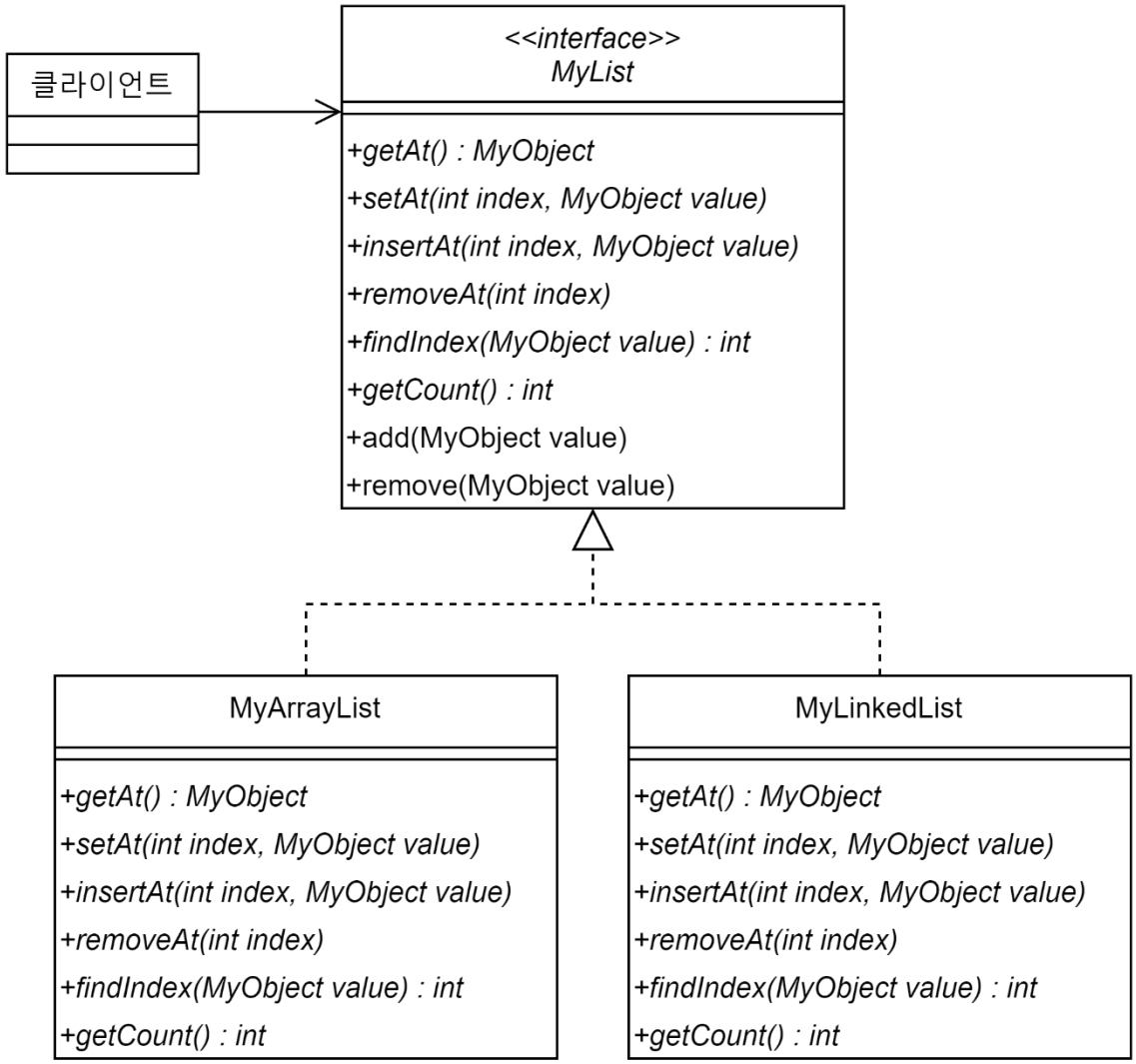
1) 개요


클라이언트
<<interface>>
MyList
+getAt() : MyObject
+setAt(int index, MyObject value)
+insertAt(int index, MyObject value)
+removeAt(int index)
+findIndex(MyObject value) : int
+getCount() : int
+add(MyObject value)
+remove(MyObject value)
MyArrayList
+getAt() : MyObject
+setAt(int index, MyObject value)
+insertAt(int index, MyObject value)
+removeAt(int index)
+findIndex(MyObject value) : int
+getCount() : int
MyLinkedList
+getAt() : MyObject
+setAt(int index, MyObject value)
+insertAt(int index, MyObject value)
+removeAt(int index)
+findIndex(MyObject value) : int
+getCount() : int

2) MyList.java

```
1  package proxy.e1;
2
3  interface MyList {
4
5      MyObject getAt(int index);
6      void setAt(int index, MyObject value);
7      void insertAt(int index, MyObject value);
8      void removeAt(int index);
9      int findIndex(MyObject value);
10     int getCount();
11
12     default void add(MyObject value) {
13         insertAt(getCount(), value);
14     }
15
16     default void remove(MyObject value) {
17         int index = findIndex(value);
18         if (index == -1)
19             return;
20         removeAt(index);
21     }
22 }
```

3) MyArrayList.java

```java
package proxy.e1;

import java.util.Arrays;

class MyArrayList implements MyList {
    private MyObject[] data;
    private int count;
    private int size;

    public MyArrayList() {
        this(10);
    }

    public MyArrayList(int size) {
        this.count = 0;
        this.size = size;
        this.data = new MyObject[size];
    }

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

    @Override
    public MyObject getAt(int index) {
        return data[index];
    }

    @Override
    public void setAt(int index, MyObject value) {
        data[index] = value;
    }

    @Override
    public void insertAt(int index, MyObject value) {
        if (count >= size)
            expand();
        for (int i = count - 1; i >= index; --i)
            data[i + 1] = data[i];
        data[index] = value;
        count++;
    }

    @Override
    public void removeAt(int index) {
        for (int i = index; i < count - 1; ++i)
            data[i] = data[i + 1];
        count--;
    }

    @Override
    public int findIndex(MyObject value) {
        for (int i = 0; i < count; ++i)
            if (value.equals(data[i]))
                return i;
        return -1;
    }

    @Override
    public int getCount() {
        return count;
    }
}
```

4) MyLinkedList.java

```
1  package proxy.e1;
2
3  public class MyLinkedList implements MyList {
4      private static class Node {
5          private MyObject data;
6          private Node prev, next;
7
8          Node(MyObject data) {
9              this.data = data;
10         }
11     }
12
13     private Node dummy;
14     private int count;
15
16     public MyLinkedList() {
17         dummy = new Node(null);
18         dummy.prev = dummy.next = dummy;
19         count = 0;
20     }
21
22     private Node getNode(int index) {
23         Node node = dummy;
24         if (index < count / 2)
25             for (int i = 0; i <= index; ++i)
26                 node = node.next;
27         else
28             for (int i = count-1; i >= index; --i)
29                 node = node.prev;
30         return node;
31     }
32
33     @Override
34     public MyObject getAt(int index) {
35         return getNode(index).data;
36     }
37
38     @Override
39     public void setAt(int index, MyObject value) {
40         getNode(index).data = value;
41     }
42
43     @Override
44     public void insertAt(int index, MyObject value) {
45         Node newNode = new Node(value);
46         Node node = getNode(index);
47         newNode.next = node;
48         newNode.prev = node.prev;
49         node.prev.next = newNode;
50         node.prev = newNode;
51         ++count;
52     }
53
54     @Override
55     public void removeAt(int index) {
56         Node node = getNode(index);
57         node.prev.next = node.next;
58         node.next.prev = node.prev;
59         --count;
60     }
61
62     @Override
63     public int findIndex(MyObject value) {
64         int index;
65         Node node = dummy.next;
66         for (index = 0; index < count; ++index) {
67             if (value.equals(node.data)) break;
68             node = node.next;
```

```
69          }
70          return index;
71      }
72
73      @Override
74      public int getCount() {
75          return count;
76      }
77  }
```

5) Example1A.java

```
1  package proxy.e1;
2
3  public class Example1A {
4
5      static void work(MyList list) {
6          for (int i=0; i < 1000; ++i) {
7              list.insertAt(0, new MyInt(999));
8              list.removeAt(0);
9          }
10     }
11
12     static void add(MyList list, int count) {
13         for (int i=0; i < count; ++i)
14             list.add(new MyInt(i));
15     }
16
17     static void print(MyList list) {
18         System.out.printf("Count: %dWn", list.getCount());
19         for (int i=0; i < list.getCount(); ++i)
20             System.out.printf("%s ", list.getAt(i));
21         System.out.println();
22     }
23
24     static void doSomething(MyList list) {
25         add(list, 100);
26         for (int i=0; i < 100; ++i)
27             work(list);
28         print(list);
29     }
30
31     public static void main(String[] args) {
32         doSomething(new MyArrayList());
33         doSomething(new MyLinkedList());
34     }
35
36 }
```

출력

```
Count: 100
MyInt(0) MyInt(1) MyInt(2) MyInt(3) MyInt(4) MyInt(5) MyInt(6) MyInt(7) MyInt(8) MyInt(9) MyInt(10) MyInt(11)
MyInt(12) MyInt(13) MyInt(14) MyInt(15) MyInt(16) MyInt(17) MyInt(18) MyInt(19) MyInt(20) MyInt(21) MyInt(22)
MyInt(23) MyInt(24) MyInt(25) MyInt(26) MyInt(27) MyInt(28) MyInt(29) MyInt(30) MyInt(31) MyInt(32) MyInt(33)
MyInt(34) MyInt(35) MyInt(36) MyInt(37) MyInt(38) MyInt(39) MyInt(40) MyInt(41) MyInt(42) MyInt(43) MyInt(44)
MyInt(45) MyInt(46) MyInt(47) MyInt(48) MyInt(49) MyInt(50) MyInt(51) MyInt(52) MyInt(53) MyInt(54) MyInt(55)
MyInt(56) MyInt(57) MyInt(58) MyInt(59) MyInt(60) MyInt(61) MyInt(62) MyInt(63) MyInt(64) MyInt(65) MyInt(66)
MyInt(67) MyInt(68) MyInt(69) MyInt(70) MyInt(71) MyInt(72) MyInt(73) MyInt(74) MyInt(75) MyInt(76) MyInt(77)
MyInt(78) MyInt(79) MyInt(80) MyInt(81) MyInt(82) MyInt(83) MyInt(84) MyInt(85) MyInt(86) MyInt(87) MyInt(88)
MyInt(89) MyInt(90) MyInt(91) MyInt(92) MyInt(93) MyInt(94) MyInt(95) MyInt(96) MyInt(97) MyInt(98) MyInt(99)
Count: 100
MyInt(0) MyInt(1) MyInt(2) MyInt(3) MyInt(4) MyInt(5) MyInt(6) MyInt(7) MyInt(8) MyInt(9) MyInt(10) MyInt(11)
MyInt(12) MyInt(13) MyInt(14) MyInt(15) MyInt(16) MyInt(17) MyInt(18) MyInt(19) MyInt(20) MyInt(21) MyInt(22)
MyInt(23) MyInt(24) MyInt(25) MyInt(26) MyInt(27) MyInt(28) MyInt(29) MyInt(30) MyInt(31) MyInt(32) MyInt(33)
MyInt(34) MyInt(35) MyInt(36) MyInt(37) MyInt(38) MyInt(39) MyInt(40) MyInt(41) MyInt(42) MyInt(43) MyInt(44)
MyInt(45) MyInt(46) MyInt(47) MyInt(48) MyInt(49) MyInt(50) MyInt(51) MyInt(52) MyInt(53) MyInt(54) MyInt(55)
MyInt(56) MyInt(57) MyInt(58) MyInt(59) MyInt(60) MyInt(61) MyInt(62) MyInt(63) MyInt(64) MyInt(65) MyInt(66)
MyInt(67) MyInt(68) MyInt(69) MyInt(70) MyInt(71) MyInt(72) MyInt(73) MyInt(74) MyInt(75) MyInt(76) MyInt(77)
MyInt(78) MyInt(79) MyInt(80) MyInt(81) MyInt(82) MyInt(83) MyInt(84) MyInt(85) MyInt(86) MyInt(87) MyInt(88)
MyInt(89) MyInt(90) MyInt(91) MyInt(92) MyInt(93) MyInt(94) MyInt(95) MyInt(96) MyInt(97) MyInt(98) MyInt(99)
```

Example1A.java

MyArrayList/MyLinkedList 목록 선두에 999 를 추가하고 제거하기를 1000번 반복
목록에 변화 없다

6) Example1B.java

```java
package proxy.e1;

import java.util.ArrayList;
import java.util.List;

public class Example1B {

    static void work(MyList list) {
        for (int i=0; i < 1000; ++i) {
            list.insertAt(0, new MyInt(999));
            list.removeAt(0);
        }
    }

    static void add(MyList list, int count) {
        for (int i = 0; i < count; ++i)
            list.add(new MyInt(i));
    }

    static void print(MyList list) {
        System.out.printf("Count: %dWn", list.getCount());
        for (int i = 0; i < list.getCount(); ++i)
            System.out.printf("%s ", list.getAt(i));
        System.out.println();
    }

    static void doSomething(MyList list) throws Exception {
        List<Thread> threads = new ArrayList<>();
        add(list, 100);
        for (int i = 0; i < 100; ++i) {
            Thread t = new Thread(() -> work(list));
            t.start();
            threads.add(t);
        }
        for (Thread t: threads)
            t.join();
        print(list);
    }

    public static void main(String[] args) throws Exception {
        doSomething(new MyArrayList());
        doSomething(new MyLinkedList());
    }

}
```

Example1B.java

위 작업을 multi thread 실행하여 충돌 발생

출력

```
Exception in thread "Thread-57" java.lang.ArrayIndexOutOfBoundsException: Index 162 out of bounds for length 160
        at proxy.e1.MyArrayList.insertAt(MyArrayList.java:40)
        at proxy.e1.Example1B.work(Example1B.java:10)
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
Exception in thread "Thread-55" java.lang.ArrayIndexOutOfBoundsException: Index 160 out of bounds for length 160
        at proxy.e1.MyArrayList.removeAt(MyArrayList.java:48)
        at proxy.e1.Example1B.work(Example1B.java:11)
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
Exception in thread "Thread-94" java.lang.ArrayIndexOutOfBoundsException: Index 160 out of bounds for length 160
        at proxy.e1.MyArrayList.removeAt(MyArrayList.java:48)
        at proxy.e1.Example1B.work(Example1B.java:11)
```

```
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
Exception in thread "Thread-52" java.lang.ArrayIndexOutOfBoundsException: Index 160 out of bounds for length 160
        at proxy.e1.MyArrayList.removeAt(MyArrayList.java:48)
        at proxy.e1.Example1B.work(Example1B.java:11)
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
Exception in thread "Thread-78" Exception in thread "Thread-96" Exception in thread "Thread-84" Exception in
thread "Thread-85" Exception in thread "Thread-87" Exception in thread "Thread-51"
java.lang.ArrayIndexOutOfBoundsException: Index 160 out of bounds for length 160
        at proxy.e1.MyArrayList.removeAt(MyArrayList.java:48)
        at proxy.e1.Example1B.work(Example1B.java:11)
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
java.lang.ArrayIndexOutOfBoundsException: Index 160 out of bounds for length 160
        at proxy.e1.MyArrayList.removeAt(MyArrayList.java:48)
        at proxy.e1.Example1B.work(Example1B.java:11)
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
java.lang.ArrayIndexOutOfBoundsException: Index 160 out of bounds for length 160
        at proxy.e1.MyArrayList.removeAt(MyArrayList.java:48)
        at proxy.e1.Example1B.work(Example1B.java:11)
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
java.lang.ArrayIndexOutOfBoundsException: Index 160 out of bounds for length 160
        at proxy.e1.MyArrayList.removeAt(MyArrayList.java:48)
        at proxy.e1.Example1B.work(Example1B.java:11)
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
java.lang.ArrayIndexOutOfBoundsException: Index 160 out of bounds for length 160
        at proxy.e1.MyArrayList.removeAt(MyArrayList.java:48)
        at proxy.e1.Example1B.work(Example1B.java:11)
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
java.lang.ArrayIndexOutOfBoundsException: Index 160 out of bounds for length 160
        at proxy.e1.MyArrayList.removeAt(MyArrayList.java:48)
        at proxy.e1.Example1B.work(Example1B.java:11)
        at proxy.e1.Example1B.lambda$0(Example1B.java:31)
        at java.base/java.lang.Thread.run(Thread.java:833)
Count: 2
MyInt(999) MyInt(999)
Count: 6043
MyInt(999) MyInt(999) MyInt(999) MyInt(0) MyInt(1) MyInt(2) MyInt(3) MyInt(4) MyInt(5) MyInt(6) MyInt(7) MyInt(8)
MyInt(9) MyInt(10) MyInt(11) MyInt(12) MyInt(13) MyInt(14) MyInt(15) MyInt(16) MyInt(17) MyInt(18) MyInt(19)
MyInt(20) MyInt(21) MyInt(22) MyInt(23) MyInt(24) MyInt(25) MyInt(26) MyInt(27) MyInt(28) MyInt(29) MyInt(30)
MyInt(31) MyInt(32) MyInt(33) MyInt(34) MyInt(35) MyInt(36) MyInt(37) MyInt(38) MyInt(39) MyInt(40) MyInt(41)
MyInt(42) MyInt(43) MyInt(44) MyInt(45) MyInt(46)

생략....
```