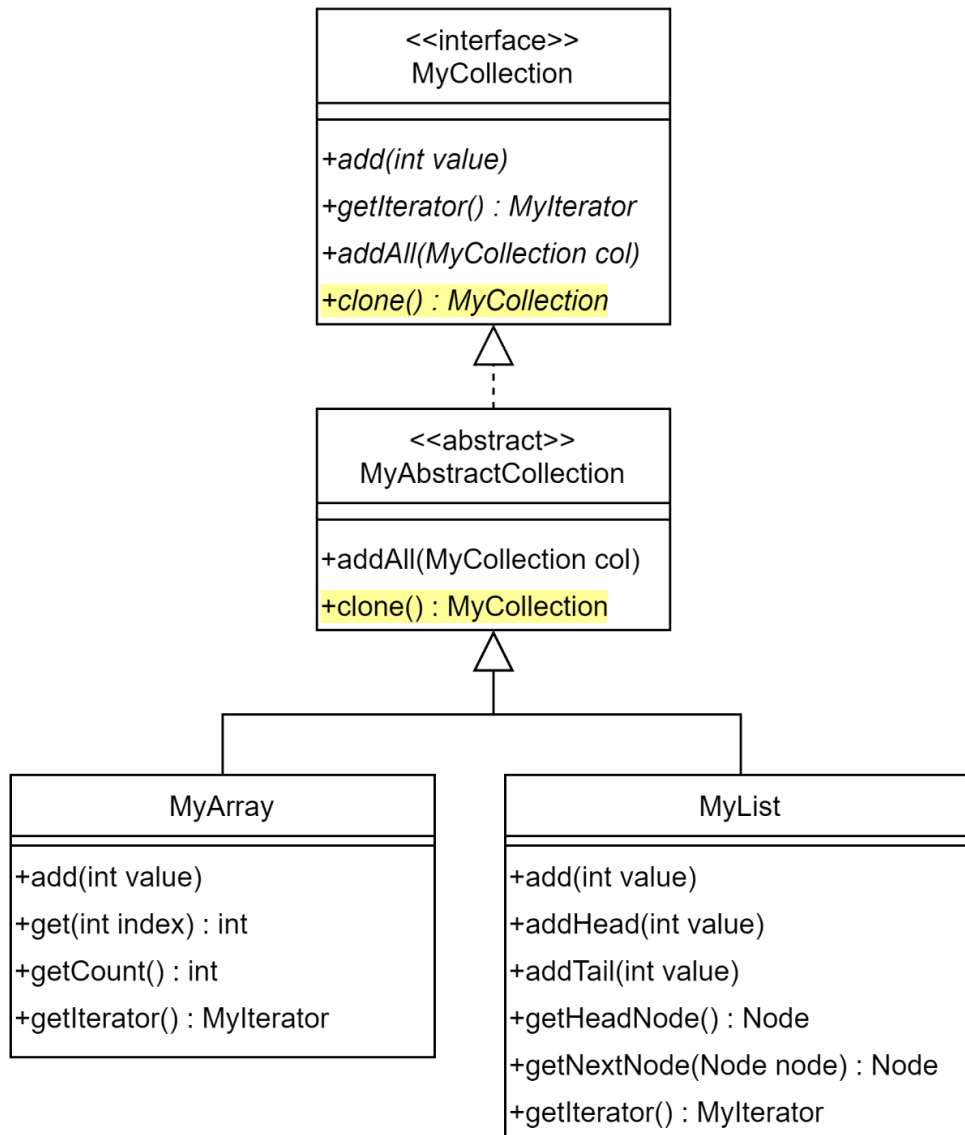


1) 개요



2) MyCollection.java

```
1 package prototype.e1;
2
3 public interface MyCollection extends Cloneable {
4     void add(int value);
5     MyIterator getIterator();
6     void addAll(MyCollection col);
7     MyCollection clone() throws CloneNotSupportedException;
8 }
```

만약 MyCollection 인터페이스에 default method 문법으로 clone 메소드를 구현하면,
java의 Object 클래스로부터 상속된 clone 메소드와
MyCollection 인터페이스로부터 상속된 clone 메소드가 충돌하게 된다. (default method conflict)

그래서 MyAbstractCollection 클래스에 clone 메소드를 구현해야 한다.

3) MyAbstractCollection.java

```
1 package prototype.e1;
2
3 public abstract class MyAbstractCollection implements MyCollection {
4
5     @Override
6     public void addAll(MyCollection col) {
7         MyIterator it = col.getIterator();
8         while (!it.isEnd())
9             add(it.getNext());
10    }
11
12    @Override
13    public MyCollection clone() throws CloneNotSupportedException {
14        return (MyAbstractCollection)super.clone();
15    }
16
17 }
```

Java 언어의 경우, 부모 클래스에 위와 같이 clone 메소드를 구현하면
그 자식 클래스들에 clone 메소드 구현이 상속된다.
이 구현은 shallow copy 이다.

4) Example1.java

```
1 package prototype.e1;
2
3 public class Example1 {
4
5     static void print(MyIterator it) {
6         while (!it.isEnd())
7             System.out.printf("%d ", it.getNext());
8         System.out.println();
9     }
10
11     static void doSomething(MyCollection col, int count) throws CloneNotSupportedException {
12         for (int i = 0; i < count / 2; ++i)
13             col.add(i);
14
15         MyCollection col2 = col.clone();
16
17         for (int i = count / 2; i < count; ++i)
18             col2.add(i);
19
20         print(col.getIterator());
21         print(col2.getIterator());
22     }
23
24     public static void main(String[] args) throws CloneNotSupportedException {
25         doSomething(new MyArray(), 10);
26         doSomething(new MyList(), 10);
27     }
28 }
29 }
```

출력

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

녹색 출력은 MyArray 사례이고, 하늘색 출력은 MyList 사례이다.

이 두 컬렉션 객체는 서로 성능이나 효율만 다를 뿐,
외부에 보여지는 동작이나 기능은 동일해야 한다.

(1) 위 코드 줄15에서 col2 객체가 col 객체의 shallow copy 이라면,
녹색 출력과 하늘색 출력 중의 어느쪽이 맞는가?

(2) clone 메소드 구현에 어떤 문제가 있는가?

5) 참고

```
1 package prototype.e2;
2
3 import java.util.ArrayList;
4 import java.util.LinkedList;
5
6 @SuppressWarnings("unchecked")
7 public class Example2a {
8
9     static void testArrayList() {
10         ArrayList<Integer> list1 = new ArrayList<>();
11         for (int i = 0; i < 5; ++i)
12             list1.add(i);
13
14         ArrayList<Integer> list2 = (ArrayList<Integer>)list1.clone();
15
16         for (int i = 5; i < 10; ++i)
17             list2.add(i);
18
19         System.out.println(list1.toString());
20         System.out.println(list2.toString());
21     }
22
23     static void testLinkedList() {
24         LinkedList<Integer> list1 = new LinkedList<>();
25         for (int i = 0; i < 5; ++i)
26             list1.add(i);
27
28         LinkedList<Integer> list2 = (LinkedList<Integer>)list1.clone();
29
30         for (int i = 5; i < 10; ++i)
31             list2.add(i);
32
33         System.out.println(list1.toString());
34         System.out.println(list2.toString());
35     }
36
37     public static void main(String[] args) throws CloneNotSupportedException {
38         testArrayList();
39         testLinkedList();
40     }
41
42 }
```

출력

```
[0, 1, 2, 3, 4]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 1, 2, 3, 4]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

힌트

Java 배열을 shallow copy 한 것이라면,
위 출력이 맞는가?