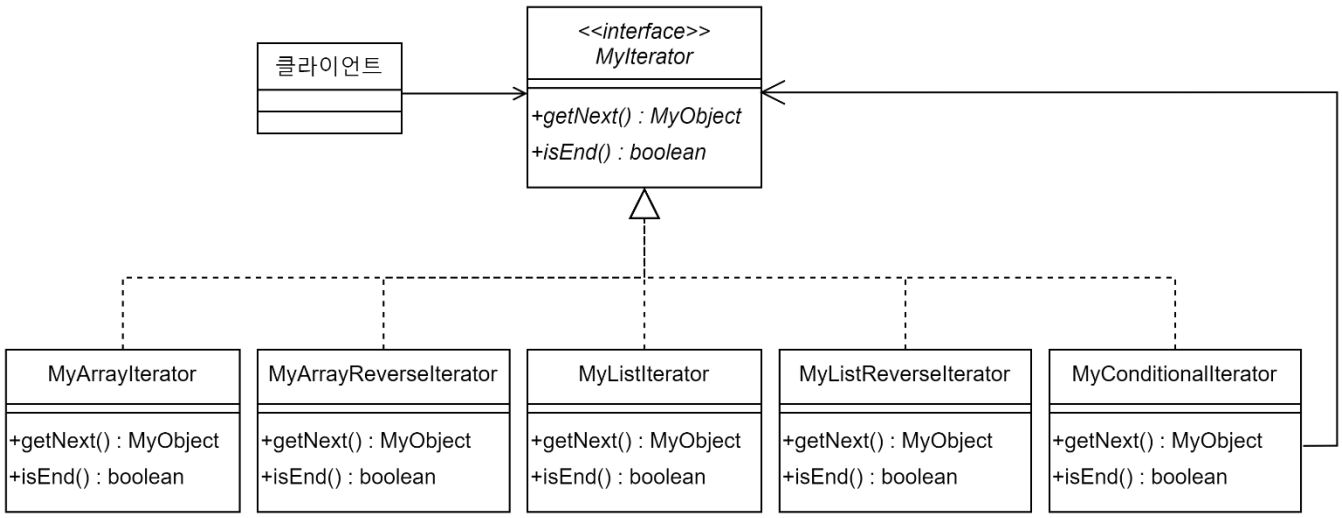
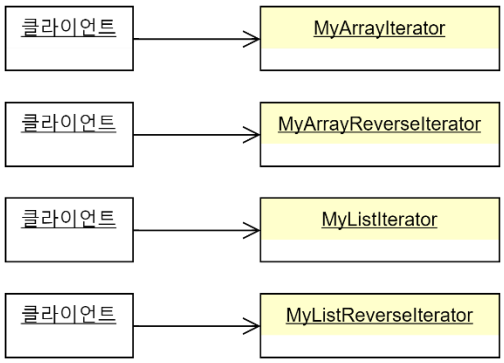
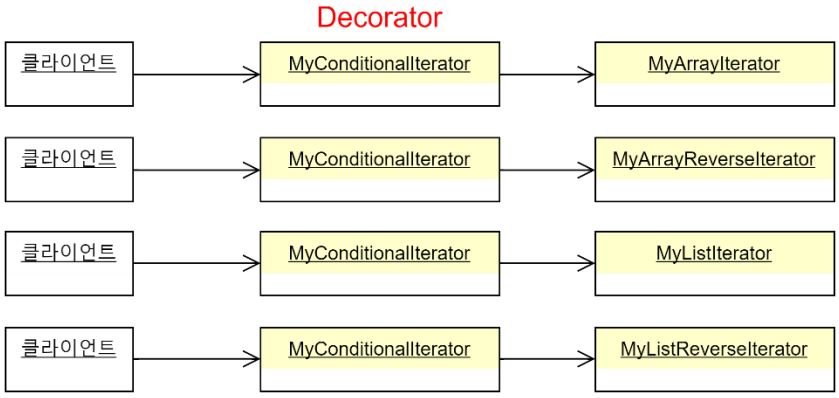
목차

1) 개요


클라이언트
<<interface>>
MyIterator
+getNext() : MyObject
+isEnd() : boolean
MyArrayIterator
+getNext() : MyObject
+isEnd() : boolean
MyArrayReverseIterator
+getNext() : MyObject
+isEnd() : boolean
MyListIterator
+getNext() : MyObject
+isEnd() : boolean
MyListReverseIterator
+getNext() : MyObject
+isEnd() : boolean
MyConditionalIterator
+getNext() : MyObject
+isEnd() : boolean


클라이언트
MyArrayIterator
클라이언트
MyArrayReverseIterator
클라이언트
MyListIterator
클라이언트
MyListReverseIterator


Decorator
클라이언트
MyConditionalIterator
MyArrayIterator
클라이언트
MyConditionalIterator
MyArrayReverseIterator
클라이언트
MyConditionalIterator
MyListIterator
클라이언트
MyConditionalIterator
MyListReverseIterator

2) MyIterator.java

```java
package decorator.i2;

public interface MyIterator {
    MyObject getNext();
    boolean isEnd();
}
```

3) MyArrayIterator.java

```java
package decorator.i2;

class MyArrayIterator implements MyIterator {
    private final MyArray myArray;
    private int current;

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

    @Override
    public MyObject getNext() {
        return this.myArray.get(current++);
    }

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

4) MyArrayReverseIterator.java

```java
package decorator.i2;

class MyArrayReverseIterator implements MyIterator {
    private final MyArray myArray;
    private int current;

    public MyArrayReverseIterator(MyArray myArray) {
        this.myArray = myArray;
        current = this.myArray.getCount() - 1;
    }

    @Override
    public MyObject getNext() {
        return this.myArray.get(current--);
    }

    @Override
    public boolean isEnd() {
        return current < 0;
    }
}
```

5) MyListIterator.java

```java
package decorator.i2;

public class MyListIterator implements MyIterator {
    protected MyList myList;
    protected MyList.Node current;

    public MyListIterator(MyList myList) {
        this.myList = myList;
        this.current = myList.dummy.next;
    }

    @Override
    public MyObject getNext() {
        MyObject r = current.data;
        current = current.next;
        return r;
    }

    @Override
    public boolean isEnd() {
        return current == myList.dummy;
    }

}
```

6) MyListReverseIterator.java

```java
package decorator.i2;

public class MyListReverseIterator implements MyIterator {
    private MyList myList;
    private MyList.Node current;

    public MyListReverseIterator(MyList myList) {
        this.myList = myList;
        this.current = myList.dummy.prev;
    }

    @Override
    public MyObject getNext() {
        MyObject r = current.data;
        current = current.prev;
        return r;
    }

    @Override
    public boolean isEnd() {
        return current == myList.dummy;
    }

}
```

7) MyConditionalIterator.java

```java
package decorator.i2;

import java.util.function.Predicate;

class MyConditionalIterator implements MyIterator {
    private MyIterator iterator;
    private Predicate<MyObject> predicate;
    private MyObject value;
    private boolean end;

    public MyConditionalIterator(MyIterator iterator, Predicate<MyObject> predicate) {
        this.iterator = iterator;
        this.predicate = predicate;
        this.value = findNext();
        this.end = false;
    }

    private MyObject findNext() {
        while (!iterator.isEnd()) {
            MyObject value = iterator.getNext();
            if (predicate.test(value)) return value;
        }
        this.end = true;
        return null;
    }

    @Override
    public MyObject getNext() {
        MyObject r = value;
        value = findNext();
        return r;
    }

    @Override
    public boolean isEnd() {
        return end;
    }
}
```

8) Example2.java

```
1  package decorator.i2;
2
3  public class Example2 {
4
5      static void print(MyIterator it) {
6          while (!it.isEnd())
7              System.out.printf("%s ", it.getNext());
8          System.out.println();
9      }
10
11     static void doSomething(MyCollection col, int count) {
12         for (int i = 0; i < count; ++i)
13             col.add(i % 2 == 0 ? new MyInt(i) : new MyStr(i));
14
15         print(col.getIterator());
16         print(col.getReverseIterator());
17         print(new MyConditionalIterator(col.getIterator(), (obj) -> obj instanceof MyInt));
18         print(new MyConditionalIterator(col.getReverseIterator(), (obj) -> obj instanceof MyStr));
19     }
20
21     public static void main(String[] args) {
22         doSomething(new MyArray(), 10);
23         doSomething(new MyList(), 10);
24     }
25 }
```

출력

```
MyInt(0) MyStr(1) MyInt(2) MyStr(3) MyInt(4) MyStr(5) MyInt(6) MyStr(7) MyInt(8) MyStr(9)
MyStr(9) MyInt(8) MyStr(7) MyInt(6) MyStr(5) MyInt(4) MyStr(3) MyInt(2) MyStr(1) MyInt(0)
MyInt(0) MyInt(2) MyInt(4) MyInt(6) MyInt(8)
MyStr(9) MyStr(7) MyStr(5) MyStr(3) MyStr(1)
MyInt(0) MyStr(1) MyInt(2) MyStr(3) MyInt(4) MyStr(5) MyInt(6) MyStr(7) MyInt(8) MyStr(9)
MyStr(9) MyInt(8) MyStr(7) MyInt(6) MyStr(5) MyInt(4) MyStr(3) MyInt(2) MyStr(1) MyInt(0)
MyInt(0) MyInt(2) MyInt(4) MyInt(6) MyInt(8)
MyStr(9) MyStr(7) MyStr(5) MyStr(3) MyStr(1)
```