1) MylteratorFactory.java

```
package abstract_factory.e2;
2
    import java.util.function.Predicate;
4
5
    public interface MylteratorFactory {
        Mylterator getIterator();
6
        Mylterator getReverselterator();
7
8
9
        default Mylterator getConditionalIterator(Predicate<MyObject> predicate) {
            return new MyConditionalIterator(getIterator(), predicate);
10
11
12
        default MyIterator getConditionalReverseIterator(Predicate<MyObject> predicate) {
13
            return new MyConditionalIterator(getReverseIterator(), predicate);
14
15
16
```

## 2) MyArray I terator Factory. java

```
package abstract_factory.e2;
2
3
    public class MyArrayIteratorFactory implements MyIteratorFactory {
4
        MyArray myArray;
5
6
        public MyArrayIteratorFactory(MyArray myArray) {
7
            this.myArray = myArray;
8
9
        @Override
10
        public Mylterator getIterator() {
11
12
            return new MyArrayIterator(myArray);
13
14
15
        @Override
        public Mylterator getReverselterator() {
16
17
            return new MyArrayReverseIterator(myArray);
18
19
    }
20
```

## 3) MyListIteratorFactory.java

```
package abstract_factory.e2;
1
2
3
     public class MyListIteratorFactory implements MyIteratorFactory {
4
          MyList myList;
5
          public MyListIteratorFactory(MyList myList) {
6
               this.myList = myList;
7
8
9
10
          @Override
          public MyIterator getIterator() {
    return new MyListIterator(myList);
11
12
13
14
          @Override
15
          public Mylterator getReverselterator() {
    return new MyListReverselterator(myList);
16
17
          }
18
19
     }
20
```