1) MyIteratorFactory.java

```java
package abstract_factory.e2;

import java.util.function.Predicate;

public interface MyIteratorFactory {
    MyIterator getIterator();
    MyIterator getReverseIterator();

    default MyIterator getConditionalIterator(Predicate<MyObject> predicate) {
        return new MyConditionalIterator(getIterator(), predicate);
    }

    default MyIterator getConditionalReverseIterator(Predicate<MyObject> predicate) {
        return new MyConditionalIterator(getReverseIterator(), predicate);
    }
}
```

2) MyArrayIteratorFactory.java

```java
package abstract_factory.e2;

public class MyArrayIteratorFactory implements MyIteratorFactory {
    MyArray myArray;

    public MyArrayIteratorFactory(MyArray myArray) {
        this.myArray = myArray;
    }

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

    @Override
    public MyIterator getReverseIterator() {
        return new MyArrayReverseIterator(myArray);
    }

}
```

3) MyListIteratorFactory.java

```java
package abstract_factory.e2;

public class MyListIteratorFactory implements MyIteratorFactory {
    MyList myList;

    public MyListIteratorFactory(MyList myList) {
        this.myList = myList;
    }

    @Override
    public MyIterator getIterator() {
        return new MyListIterator(myList);
    }

    @Override
    public MyIterator getReverseIterator() {
        return new MyListReverseIterator(myList);
    }

}
```