luv2code

Assertions: Arrays, Iterables and Lines

Assertions

Method name	Description
`assertArrayEquals`	Assert that both object arrays are deeply equal

Code to Test

DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

    private String[] firstThreeLettersOfAlphabet = {"A", "B", "C"};

    public String[] getFirstThreeLettersOfAlphabet() {
        return firstThreeLettersOfAlphabet;
    }
}
```

Make comparisons against this array

DemoUtilsTest.java

```
package com.luv2code.junitdemo;

import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;

class DemoUtilsTest {

    DemoUtils demoUtils;

    ...

    @DisplayName("Array Equals")
    @Test
    void testArrayEquals() {
        String[] stringArray = {"A", "B", "C"};

        assertArrayEquals(stringArray, demoUtils.getFirstThreeLettersOfAlphabet(), "Arrays should be the same");
    }

}
```

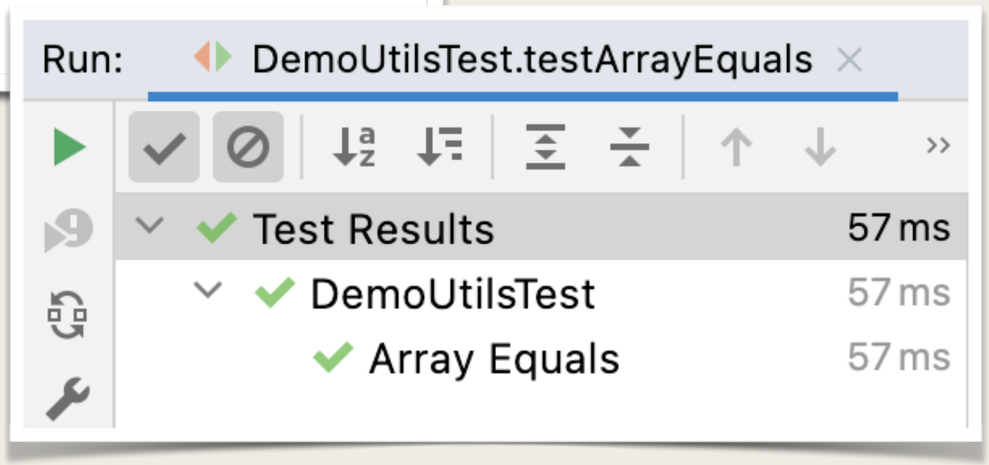
DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

    private String[] firstThreeLettersOfAlphabet = {"A", "B", "C"};

    public String[] getFirstThreeLettersOfAlphabet() {
        return firstThreeLettersOfAlphabet;
    }
}
```



Assertions

Method name	Description
`assertIterableEquals`	Assert that both object iterables are deeply equal

An "iterable" is an instance of a class that implements the `java.lang.Iterable` interface

Examples: `ArrayList`, `LinkedList`, `HashSet`, `TreeSet` ...

Code to Test

DemoUtils.java

```
package com.luv2code.junitdemo;

import java.util.List;

public class DemoUtils {

    private List<String> academyInList = List.of("luv", "2", "code");

    public List<String> getAcademyInList() {
        return academyInList;
    }
}
```

Make comparisons against this List

DemoUtilsTest.java

```java
package com.luv2code.junitdemo;

import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;
import java.util.List;

class DemoUtilsTest {

    DemoUtils demoUtils;

    ...

    @DisplayName("Iterable equals")
    @Test
    void testIterableEquals() {
        List<String> theList = List.of("luv", "2", "code");

        assertIterableEquals(theList, demoUtils.getAcademyInList(), "Expected list should be same as actual list");
    }
```

DemoUtils.java

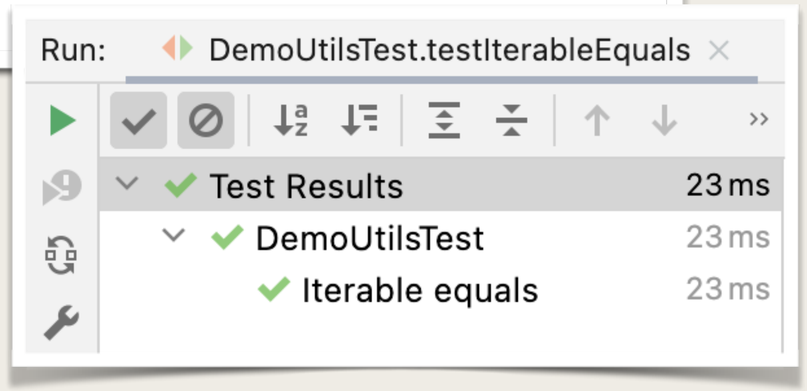
```java
package com.luv2code.junitdemo;

import java.util.List;

public class DemoUtils {

    private List<String> academyInList = List.of("luv", "2", "code");

    public List<String> getAcademyInList() {
        return academyInList;
    }
}
```



Assertions

Method name	Description
`assertLinesMatch`	Assert that both lists of strings match

Code to Test

DemoUtils.java

```
package com.luv2code.junitdemo;

import java.util.List;

public class DemoUtils {

    private List<String> academyInList = List.of("luv", "2", "code");

    public List<String> getAcademyInList() {
        return academyInList;
    }
}
```

Make comparisons against this List

DemoUtilsTest.java

```java
package com.luv2code.junitdemo;

import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;
import java.util.List;

class DemoUtilsTest {

    DemoUtils demoUtils;

    ...

    @DisplayName("Lines match")
    @Test
    void testLinesMatch() {
        List<String> theList = List.of("luv", "2", "code");

        assertLinesMatch(theList, demoUtils.getAcademyInList(), "Lines should match");
    }
```

DemoUtils.java

```java
package com.luv2code.junitdemo;

import java.util.List;

public class DemoUtils {

    private List<String> academyInList = List.of("luv", "2", "code");

    public List<String> getAcademyInList() {
        return academyInList;
    }
}
```