

SWE-619 Assignment-13

Fall 2019

-Amish Papneja

-Avinash Arunachalam A Murugappan

- Rushil Nandan Dubey

We have decided to perform Question-6 for this assignment.

Goal: Applying lessons learned.

6. Convert an existing JUnit test set to JUnit Theories. If you wish to construct JUnit from scratch for this, that's fine too.

For assignment 13, we converted our assignment 10 IntSet test cases to use theories. In doing so, we added an array of IntSets to be used as the DataPoints. It is demonstrated in IntSetTheoryTest.java, which is attached below.

To convert the testEquals, testClone, and testHashCode methods, we added an IntSet parameter and an assumption that the parameter cannot be null.

We also performed property testing: reflexive, symmetry, transitivity in testEquals and testHashCode methods.

Test Results for IntSetTheoryTest.java:

The screenshot displays the IntelliJ IDEA IDE with the `IntSetTheoryTest.java` file open. The code is as follows:

```
1 import org.junit.Test;
2 import org.junit.experimental.theories.DataPoint;
3 import org.junit.experimental.theories.DataPoints;
4 import org.junit.experimental.theories.Theories;
5 import org.junit.experimental.theories.Theory;
6 import org.junit.runner.RunWith;
7
8 import static org.junit.Assert.assertEquals;
9 import static org.junit.Assert.assertTrue;
10 import static org.junit.Assume.assumeTrue;
11
12
13 @RunWith(Theories.class)
14 public class IntSetTheoryTest {
15
16     @DataPoints
17     public static IntSet[] set2 = {new IntSet(), new IntSetSub(), new IntSet(), new IntSet()};
18
19
20
21     @Theory
22     public void testClone(IntSet set) {
23         assumeTrue( ! set.isNull());
24
25         IntSet clone = set.clone();
26
27         assertTrue( condition: set!=clone);
28         assertEquals(set.getClass(), clone.getClass());
29     }
30
31     @Theory
32     // IntSetTheoryTest
```

The bottom panel shows the test results for `IntSetTheoryTest`. All tests passed successfully.

Test Name	Duration
IntSetTheoryTest	19 ms
testClone	15 ms
testHashCode	2 ms
testEquals	2 ms

Summary: Tests passed: 3 of 3 tests - 19 ms. Process finished with exit code 0.

Code implementation:

IntSet.java:

```
import java.util.*;

public class IntSet implements Cloneable {
    private List<Integer> els;

    @Override public boolean equals(Object obj) {
        if (!(obj.getClass()==this.getClass()))
            return false;
        IntSet s = (IntSet) obj;
        return els.equals(s.els);
    }

    @Override public int hashCode() {
        int result = 0;
        for (Integer i : els) {
            result += i.hashCode(); // from class
            result += 31 * result + i.hashCode(); // bloch
        }
        return result;
    }

    public IntSet () { els = new ArrayList<Integer>(); }

    private IntSet (List<Integer> list) { els = list; }

    // previous implementation
    // @Override public IntSet clone() {
    //     return new IntSet (new ArrayList<Integer>(els));
    // }

    // our implementation
    @Override public IntSet clone() {
        try {
            IntSet s = (IntSet) super.clone();
            s.els.addAll(els);
            return s;
        } catch (CloneNotSupportedException e) {
            throw new IllegalStateException();
        }
    }
}
```

IntSetSub.java:

```
public class IntSetSub extends IntSet{

    public IntSetSub() {
        super();
    }

}
```

IntSetTest.java:

```
import org.junit.Test;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;

public class IntSetTest {

    @Test
    public void testOldClone() {
        IntSetSub set = new IntSetSub();
        IntSetSub sub =(IntSetSub) set.clone();
        assertTrue(sub!=set);
        assertEquals(set.getClass(),sub.getClass());
    }

    @Test
    public void testClone() {
        IntSet set = new IntSet();
        IntSet clone = set.clone();

        assertTrue(set!=clone);
        assertEquals(set.getClass(),clone.getClass());
    }

    @Test
    public void testEquals() {
        IntSet set = new IntSet();
        IntSet clone = set.clone();

        //reflexive
        assertTrue(set.equals(set));

        // symmetry
        assertTrue(set.equals(clone));
        assertTrue(clone.equals(set));

        // transitivity
        IntSet clone2 = clone.clone();
        assertTrue(set.equals(clone2));
        assertTrue(clone2.equals(clone));
    }

    @Test
```

```

    public void testHashCode() {
        IntSet set = new IntSet();
        IntSet clone = set.clone();

        // reflexive
        assertTrue(set.hashCode() == set.hashCode());

        // symmetry
        assertEquals(set.hashCode(), clone.hashCode());

        // transitivity
        IntSet clone2 = clone.clone();
        assertEquals(set.hashCode(), clone2.hashCode());
        assertEquals(clone2.hashCode(), clone.hashCode());
    }
}

```

IntSetTheoryTest.java:

```

import org.junit.Test;
import org.junit.experimental.theories.DataPoint;
import org.junit.experimental.theories.DataPoints;
import org.junit.experimental.theories.Theories;
import org.junit.experimental.theories.Theory;
import org.junit.runner.RunWith;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;
import static org.junit.Assume.assumeTrue;

@RunWith(Theories.class)
public class IntSetTheoryTest {

    @DataPoints
    public static IntSet[] set2 = {new IntSet(), new IntSetSub(), new IntSet(), new
IntSet()};

    @Theory
    public void testClone(IntSet set) {
        assumeTrue(set != null);

        IntSet clone = set.clone();

        assertTrue(set != clone);
        assertEquals(set.getClass(), clone.getClass());
    }

    @Theory
    public void testEquals(IntSet set) {
        assumeTrue(set != null);

        IntSet clone = set.clone();
        IntSet clone2 = clone.clone();
    }
}

```

```

    // reflexive
    assertTrue(set.equals(set));

    // symmetry
    assertTrue(set.equals(clone));
    assertTrue(clone.equals(set));

    // transitivity
    assertTrue(set.equals(clone2));
    assertTrue(clone2.equals(clone));
}

@Theory
public void testHashCode(IntSet set) {
    assumeTrue(set != null);

    IntSet clone = set.clone();

    // reflexive
    assertTrue(set.hashCode() == set.hashCode());

    // symmetry
    assertEquals(set.hashCode(), clone.hashCode());

    // transitivity
    IntSet clone2 = clone.clone();
    assertEquals(set.hashCode(), clone2.hashCode());
    assertEquals(clone2.hashCode(), clone.hashCode());
}
}

```

Contributions:

- Amish Papneja

- Converted the testHashCode method to use JUnit Theory.
- Contributed to discussion, regarding this assignment.

- Avinash Arunachalam A Murugappan

- Converted the TestEquals test method to use JUnit Theory.
- Contributed to discussion, regarding this assignment.

- Rushil Nandan Dubey

- Converted the testClone method to use JUnit Theory.
- Contributed to discussion, regarding this assignment.

All group members contributed substantially and equally.