Cpt S 422: Software Engineering Principles II

Testing levels – Unit Testing

Dr. Venera Arnaoudova

WASHINGTON STATE UNIVERSITY

JUnit

- Naming conventions
  - "Test" as suffix (e.g., Date -> DateTest)
- Common assertions

JUnit 4	JUnit 5
assertTrue / assertFalse	
assertNull / assertNotNull	
assertEquals / assertNotEquals	

  - For more see class Assert on
    - ✓ JUnit 4: http://junit.org/junit4/javadoc/latest/index.html
    - ✓ Junit 5: http://junit.org/junit5/docs/current/api/

JUnit (cont.)

- Common annotations

JUnit 4	JUnit 5
@Test	
@Test (expected=<exception>) @Test(timeout=<time in ms>)	
@Before / @After	@BeforeEach / @AfterEach
@BeforeClass / @AfterClass	@BeforeAll / @AfterAll
@Ignore / @Ignore(<Message explaining why disabled>)	@Disable / @Disable(<Message explaining why disabled>)

- For more see
  - ✓ JUnit 4: http://junit.org/junit4/javadoc/latest/index.html
  - ✓ Junit 5: http://junit.org/junit5/docs/current/api/

JUnit 4

```java
import org.junit.*;
import static org.junit.Assert.*;

public class SampleTest {

    private java.util.List emptyList;

    /**
     * Sets up the test fixture.
     * (Called before every test case method.)
     */
    @Before
    public void setUp() {
        emptyList = new java.util.ArrayList();
    }

    /**
     * Tears down the test fixture.
     * (Called after every test case method.)
     */
    @After
    public void tearDown() {
        emptyList = null;
    }

    @Test
    public void testSomeBehavior() {
        assertEquals("Empty list should have 0 elements", 0, emptyList.size());
    }

    @Test(expected=IndexOutOfBoundsException.class)
    public void testForException() {
        Object o = emptyList.get(0);
    }
}
```

http://junit.org/junit4/faq.html#atests_16

JUnit 5

```java
class StandardTests {

    @BeforeAll
    static void initAll() {
    }

    @BeforeEach
    void init() {
    }

    @Test
    void succeedingTest() {
    }

    @Test
    void failingTest() {
        fail("a failing test");
    }

    @Test
    @Disabled("for demonstration purposes")
    void skippedTest() {
        // not executed
    }

    @AfterEach
    void tearDown() {
    }

    @AfterAll
    static void tearDownAll() {
    }

}
```

http://junit.org/junit5/docs/current/user-guide/#writing-tests-standard

Tasks for Today

- TDD
  - isLeap (see next slide for instructions)
  - lastDayOfMonth
  - dateToDayNumber
  - dateToDayName
- Unit Testing
  - zodiacSign
  - validCombination
  - … (everything else)

TDD Example

- Implement method **isLeap** in class **Date** by following a TDD model using the following tasks:
  - A year that is divisible by 4 is a leap year
  - A year not divisible by 4 is a common year
  - A century year not divisible by 400 is a common year
  - A century year divisible by 400 is a leap year
- Remember:
  - Write test cases before implementing the functionality
  - Refactor whenever possible