Announcements

- Remember to submit your resume <u>this week</u>!

- Google @ WSU next week:

  - **Register for the event <u>HERE</u>.**

  - **Include a link to your resume in the form!**

Cpt S 422: Software Engineering Principles II

Testing levels – Unit Testing – part III

Dr. Venera Arnaoudova

WASHINGTON STATE UNIVERSITY

Tasks from last time

1. Refactor your code to match the partial implementation of Date-v.2-partial.java (using the refactoring feature of Eclipse!)

2. Implement testInitializeAttributes() to test method initializeAttributes(int dd, int mm, int yyyy)

3. Refactor Date.java as follows (and **adapt your tests at each step!**):

   ➢ Check if a date is valid before creating an object

   ➢ ~~static~~ dateToDayName(~~int mm,int dd,int yyyy~~)

   ➢ ~~static~~ dateToDayNumber(~~int mm,int dd,int yyyy~~)

   ➢ ~~static~~ lastDayOfMonth(~~int mm,int yyyy~~)

   ➢ zodiacSign(~~int mm, int dd~~)

   ➢ initializeAttributes(~~int dd, int mm, int yyyy~~)

Limitations of Mockito

❑ Cannot mock

  ➢ Constructors

  ➢ Static methods

  ➢ Private methods

  ➢ …

PowerMockito

❑ Download the .jar files from the course webpage and add them to your testing project

❑ Add the following imports

```
import org.easymock.EasyMock;
import org.junit.runner.RunWith;
import org.powermock.api.easymock.PowerMock;
import org.powermock.core.classloader.annotations.PrepareForTest;
import org.powermock.modules.junit4.PowerMockRunner;
```

❑ Annotate the test class as follows:

```
@RunWith(PowerMockRunner.class)
@PrepareForTest(Date.class)
public class DateTest {
```

Testing validCombination

```java
@Test
public void testValidCombination() {
        PowerMock.mockStaticPartial(Date.class, "isLeap");
        EasyMock.expect(Date.isLeap(2012)).andReturn(true);
        PowerMock.replayAll();
        assertTrue(Date.validCombination(29,02,2012));

        PowerMock.mockStaticPartial(Date.class, "isLeap");
        EasyMock.expect(Date.isLeap(2017)).andReturn(false);
        PowerMock.replayAll();
        assertFalse(Date.validCombination(29,02,2017));
        assertTrue(Date.validCombination(31,01,2017));
        assertFalse(Date.validCombination(31,06,2017));
}
```

Tasks for today

❑ Test

  ➢ isValidDate

  ➢ validCombination

  ➢ validRangeForDay

❑ Finish **unit testing** all your code <u>before</u> next class!