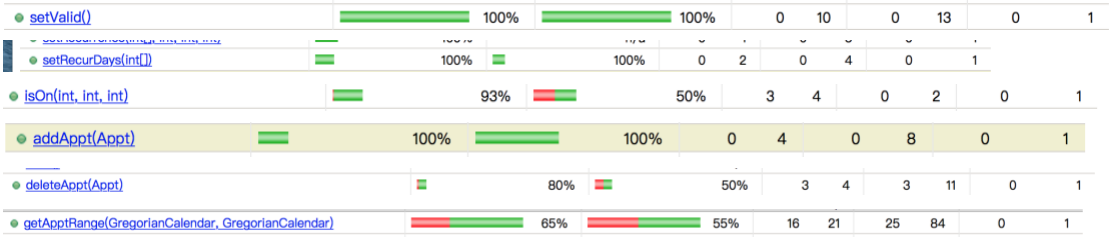
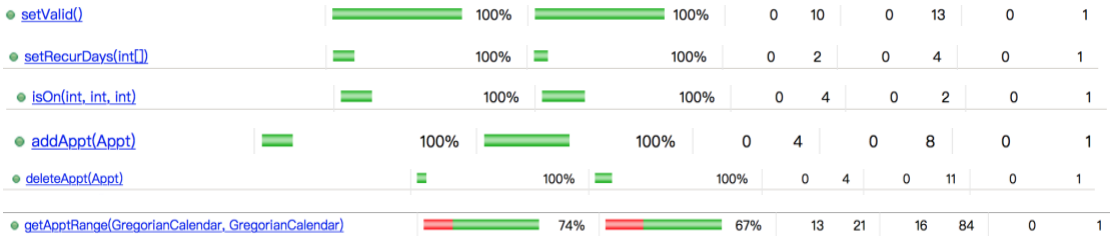
CS362
Assignment 4
Donghao Lin(lindo)

Assignment 2 coverage



Assignment 4 coverage



1.Random Testers

I change all the constructor values in unit test from setting them a specific value to random value in random test. Random test improves the coverage by randomly generating values many times instead of manually input the value that may not cover all cases. It saves a lot of time to test the program.

I failed to cover getApptRange to 100% because failed to cover the code below.

```
        //If the data handler isn't initialized return null
        if (isValid() == false) {
            return null;

        if (diagnose) {
            System.out.println("=====================================");
            System.out.println("DEBUGGING GETTING OF APPOINTMENTS      ");

        if (diagnose) {
            System.out.println("Root node: " + root.getTagName());
            System.out.println("All following nodes should be appt nodes.");
```

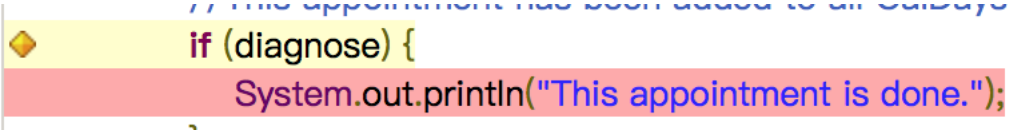
```
        if (diagnose) {
            System.out.println("Nodes under the root: " +
                currentAppt.getTagName());
        }

        //For this appointment, get the values of all fields
        NodeList fieldNodes = currentAppt.getChildNodes();
        Hashtable<String, String> fields = new Hashtable<String, String>(
        if (diagnose) {
            System.out.println("Preparing to read each field for the appt");
        }
        for (int j = 0; j < fieldNodes.getLength(); j++) {
            Element currentField = (Element) fieldNodes.item(j);
            String fieldName = currentField.getTagName();
            if (diagnose) {
                System.out.println("Reading field: " + fieldName);
            }
            String fieldValue = "";
            NodeList fieldValueNodes = currentField.getChildNodes();
            for (int k = 0; k < fieldValueNodes.getLength(); k++) {
                Text text = (Text)fieldValueNodes.item(k);
                fieldValue += text.getData();
            }
            if (diagnose) {
                System.out.println("Reading field's value: " + fieldValue);
            }

        if (diagnose) {
            System.out.println("Calculating appointment occurrences.");
        }
        //Figure out which days the appointment occurs on
        LinkedList<GregorianCalendar>  apptOccursOnDays =
            getApptOccurences(appt, firstDay, lastDay);

        if (diagnose) {
            System.out.println("This appointment occurs on: ");
        }

            if (diagnose) {
                System.out.println("\t" + apptOccursOn);
            }
```



For isValid(), I try to create an appt with all Null, but it is still true, so I do not know how to set isValid() to false.
The other code I did not cover because they are all basically have same condition that diagnose should be true, but it can not be changed from false to true in the method.

2. Unit VS Random

Unit tests should have manually data input to test some cases, but easier to determine valid and invalid cases. Random tests have higher coverage and save a lot of time by generating random test cases, but it is harder to determine the boundary cases due to the data is randomly generated.

I think unit tests have better fault detection capability. Because there are less test cases generated manually than random tests, and data can be controlled rather than randomly generated.