Chapter 4

Team 3: Meg, Mitch, Itzayana

CSCI 362 Project: OpenMrs

Experience

Our experience working on OpenMrs continues to grow our knowledge of both open source projects and automated testing. Through this small sprint of work, we learned a few valuable lessons about consistently using the same operating systems, understanding method requirements, and scripting. We learned that even writing what seems to be an ordinary txt file in windows can have an effect on how that text file is read later.

Updates to Project

After our presentation on Thursday, we had a couple of changes to make to our project. First we decided to replace the compareDrugsByName method in DrugsByNameComparitor class with the LastMomentOfDay method found in the OpenmrsUtils class. The main reason we decided to swap the methods was due to the unpredictability of the output of the compareDrugsByName method. According to it's requirements it would return a positive number both when the second drug came before and after the first drug alphabetically. Since drastically different inputs could result in the same output it became very hard to test. In addition there were no instructions on how this function came up with the number it output, which again made it very hard to test. As a result, we decided to toss it and replace it with the LastMomentOfDay method, which takes a date and shifts the time of day to be the last second of the day. This proved to be much easier to calculate the actual output.

The second thing we made changes to was the script. The first issue with our script was that it was not actually comparing the input and output, but rather just defaulting to pass. This was not what we wanted, but now our script has been updated to actually compare the two values

and correctly determine if the test passed or failed. In this change our team has discovered an odd quirk in the DateUtil method. When the script is run on Mitch's computer the test fails, claiming that the expected output is one second ahead of the actual output. However when the script runs on Meg's or Itzayana's computer the test passes. We're not sure why this happens, but we think it has something to do with the Date class which has been deprecated.

Finally, the last thing we did was create drivers and test cases for the other method. This didn't take too long, since we just modeled them after the first example we had already completed. Luckily, because of the way we changed our script all the test cases and drivers are working as planned.

Methods, Drivers and Test Cases

lastMomentOfDay

```
48          /**
49           * Gets the date having the last millisecond of a given day. Meaning that the hours, seconds,
50           * and milliseconds are the latest possible for that day.
51           *
52           * @param day the day.
53           * @return the date with the last millisecond of the day.
54           */
55          public static Date getLastMomentOfDay(Date day) {
56                  Calendar calender = Calendar.getInstance();
57                  calender.setTime(day);
58                  calender.set(Calendar.HOUR_OF_DAY, 23);
59                  calender.set(Calendar.MINUTE, 59);
60                  calender.set(Calendar.SECOND, 59);
61                  calender.set(Calendar.MILLISECOND, 999);
62
63                  return calender.getTime();
64          }
65
```

Figure 1: lastMomentOfDay Method found in the OpenmrsUtil Class

```java
package testCaseExecutables;
import java.util.Calendar;
import java.util.Date;
import java.time.Instant;

public class getLastMomentOfDayDriver{
        public static void main(String[] args) {
                if(args.length < 1){
                        System.out.println("No input to test");
                }
                else{
                        long dateVal = Long.parseLong(args[0]);
                        Date date1 = new Date(dateVal);
                        System.out.println(OpenmrsUtil.getLastMomentOfDay(date1));
                }
        }
}
```

Figure 2: truncateToSeconds Driver Class

Test Cases
01 OpenmrsUtil This method will take a date, convert it to a calendar, then shift the time to be the last second of the day on that date. getLastMomentOfDay 10000 Wed Dec 31 23:59:59 EST 1969 getLastMomentOfDayDriver
02 OpenmrsUtil This method will take a date, convert it to a calendar, then shift the time to be the last second of the day on that date. getLastMomentOfDay -10000000000 Sun Sep 07 23:59:59 EDT 1969 getLastMomentOfDayDriver
03 OpenmrsUtil This method will take a date, convert it to a calendar, then shift the time to be the last second

of the day on that date. getLastMomentOfDay 0 Wed Dec 31 23:59:59 EST 1969 getLastMomentOfDayDriver
04 OpenmrsUtil This method will take a date, convert it to a calendar, then shift the time to be the last second of the day on that date. getLastMomentOfDay 1000000000000 Sat Sep 08 23:59:59 EDT 2001 getLastMomentOfDayDriver
05 OpenmrsUtil This method will take a date, convert it to a calendar, then shift the time to be the last second of the day on that date. getLastMomentOfDay 914209500000 Sun Dec 20 23:59:59 EST 1998 getLastMomentOfDayDriver

truncateToSecond

```java
package testCaseExecutables;
import java.time.Instant;
import java.time.temporal.ChronoUnit;
import java.util.Date;

/**
 * Utility classes that provide date-related methods
 * @since 2.0
 */
public class DateUtil {

	public DateUtil() {
	}

	/**
	 * @param date
	 * @return date truncated to second precision (e.g. with milliseconds dropped)
	 */
	public static Date truncateToSeconds(Date date) {
		Instant instant = date.toInstant().truncatedTo(ChronoUnit.SECONDS);
		return Date.from(instant);
	}
}
```

Figure 3: truncateToSeconds Method found in the DateUtil Class

```java
package testCaseExecutables;
import java.util.Date;
import java.time.Instant;


public class DateUtilDriver{

        public static void main(String args[]){
                long dateVal = Long.parseLong(args[0]);
                Date date1 = new Date(dateVal);
                DateUtil date = new DateUtil();

                System.out.println(date.truncateToSeconds(date1));
        }
}
```

Figure 4: truncateToSeconds Driver Class

Test Cases
06 DateUtil This method will return a Date with the milliseconds truncated truncateToSeconds 1234094 Wed Dec 31 19:20:34 EST 1969 DateUtilDriver
07 DateUtil This method will return a Date with the milliseconds truncated truncateToSeconds 12340940087655343 Sun Oct 28 13:14:15 EDT 393038 DateUtilDriver
08 DateUtil This method will return a Date with the milliseconds truncated truncateToSeconds 0 Wed Dec 31 19:00:00 EST 1969 DateUtilDriver

09 DateUtil This method will return a Date with the milliseconds truncated truncateToSeconds -847350384 Sun Dec 21 23:37:30 EST 1969 DateUtilDriver
10 DateUtil This method will return a Date with the milliseconds truncated truncateToSeconds -949474704827 Thu Nov 30 12:01:36 EST 1939 DateUtilDriver

containsOnlyDigits

```
131
132         /**
133          * @param test the string to test
134          * @return true if the passed string contains only numeric characters
135          * <strong>Should</strong> return true if string contains only digits
136          * <strong>Should</strong> return false if string contains any non-digits
137          */
138         public static boolean containsOnlyDigits(String test) {
139                 if (test != null) {
140                         for (char c : test.toCharArray()) {
141                                 if (!Character.isDigit(c)) {
142                                         return false;
143                                 }
144                         }
145                 }
146                 return !test.isEmpty();
147         }
148
```

Figure 5: containsOnlyDigits Method found in the OpenmrsUtil Class

```java
package testCaseExecutables;
public class containsOnlyDigitsDriver{

	public static void main(String[] args) {
		String input = args[0];
			if(args[0].toLowerCase().compareTo("null") == 0){
				input = null;
			}
		OpenmrsUtil oMrs = new OpenmrsUtil();
		boolean output = oMrs.containsOnlyDigits(input);
		System.out.println(output);
	}
}
```

Figure 6: contiansOnlyDigits Driver Class

Test Cases
11 OpenmrsUtil This method returns true if a string only contains digits containsOnlyDigits 808760 true containsOnlyDigitsDriver
12 OpenmrsUtil This method returns true if a string only contains digits containsOnlyDigits -842 false containsOnlyDigitsDriver
13 OpenmrsUtil This method returns true if a string only contains digits containsOnlyDigits 8940 true containsOnlyDigitsDriver
14

OpenmrsUtil This method returns true if a string only contains digits containsOnlyDigits 34hi80 false containsOnlyDigitsDriver
15 OpenmrsUtil This method returns true if a string only contains digits containsOnlyDigits 60 78 false containsOnlyDigitsDriver

convertToInteger

```
149                 /**
150                  * This method converts the given Long value to an Integer. If the Long value will not fit in an
151                  * Integer an exception is thrown
152                  *
153                  * @param longValue the value to convert
154                  * @return the long value in integer form.
155                  * @throws IllegalArgumentException if the long value does not fit into an integer
156                  */
157                 public static Integer convertToInteger(Long longValue) {
158                         if (longValue < Integer.MIN_VALUE || longValue > Integer.MAX_VALUE) {
159                                 return null;
160                                 //throw new IllegalArgumentException(longValue + " cannot be cast to Integer without changing its value.");
161                         }
162                         return longValue.intValue();
163                 }
164 }
```

Figure 7: convertToInteger Method found in the OpenmrsUtil Class

```java
package testCaseExecutables;
public class convertToIntegerDriver{

	public static void main(String args[]){

			OpenmrsUtil oMrs = new OpenmrsUtil();
			Long input = Long.parseLong(args[0]);   //Turns strings pass in to Longs
				//Stores the expected val in var
			Integer actual_output = oMrs.convertToInteger(input);   //Runs the method and saves output
			String actual_output_string;

			//If the output is null save that as the output sting
			System.out.println(actual_output);

		}
	}
```

Figure 8: convertToInteger Driver Class

Test Cases
16 OpenmrsUtil This method turns a long into an integer, as long as it does not exceed the int memory limit convertToInteger 600 600 convertToIntegerDriver
17 OpenmrsUtil This method turns a long into an integer, as long as it does not exceed the int memory limit convertToInteger 6000000000 null convertToIntegerDriver
18 OpenmrsUtil This method turns a long into an integer, as long as it does not exceed the int memory limit convertToInteger -83878527402 null convertToIntegerDriver
19 OpenmrsUtil This method turns a long into an integer, as long as it does not exceed the int memory limit

convertToInteger 2147483647 2147483647 convertToIntegerDriver
20 OpenmrsUtil This method turns a long into an integer, as long as it does not exceed the int memory limit convertToInteger -2147483648 -2147483648 convertToIntegerDriver

containsUpperAndLower

```
116                 /**
117              * @param test the string to test
118              * @return true if the passed string contains both upper and lower case characters
119              * <strong>Should</strong> return true if string contains upper and lower case
120              * <strong>Should</strong> return false if string does not contain lower case characters
121              * <strong>Should</strong> return false if string does not contain upper case characters
122              */
123             public static boolean containsUpperAndLowerCase(String test) {
124                     if (test != null) {
125                             Pattern pattern = Pattern.compile("^(?=.*?[A-Z])(?=.*?[a-z])[\\w|\\W]*$");
126                             Matcher matcher = pattern.matcher(test);
127                             return matcher.matches();
128                     }
129                     return false;
130             }
131
```

Figure 9: convertToInteger Method found in the OpenmrsUtil Class

```java
package testCaseExecutables;
public class containsOnlyDigitsDriver{

	public static void main(String[] args) {
		String input = args[0];
			if(args[0].toLowerCase().compareTo("null") == 0){
				input = null;
			}
		OpenmrsUtil oMrs = new OpenmrsUtil();
		boolean output = oMrs.containsOnlyDigits(input);
		System.out.println(output);
	}
}
```

Figure 10: convertToInteger Driver Class

Test Cases
11 OpenmrsUtil This method returns true if a string only contains digits containsOnlyDigits 808760 true containsOnlyDigitsDriver
12 OpenmrsUtil This method returns true if a string only contains digits containsOnlyDigits -842 false containsOnlyDigitsDriver
13 OpenmrsUtil This method returns true if a string only contains digits containsOnlyDigits 8940 true containsOnlyDigitsDriver
14 OpenmrsUtil

This method returns true if a string only contains digits containsOnlyDigits 34hi80 false containsOnlyDigitsDriver
15 OpenmrsUtil This method returns true if a string only contains digits containsOnlyDigits 60 78 false containsOnlyDigitsDriver

Table Output

Team 3 | Carrillo, Krawczyk, Suzara

Test Results

Test ID	Class Name	Summary	Method Type	Inputs	Expected Outputs	Driver	Result	Pass/Fail
01	OpenmrsUtil	This method will take a date convert it a calender, then shift the time to be the last second of the day on that date.	getLastMomentOfDay	10000	Wed Dec 31 23:59:59 EST 1969	getLastMomentOfDayDriver	Wed Dec 31 23:59:59 EST 1969	Passed
02	OpenmrsUtil	This method will take a date convert it a calender, then shift the time to be the last second of the day on that date.	getLastMomentOfDay	-10000000000	Sun Sep 07 23:59:59 EDT 1969	getLastMomentOfDayDriver	Sun Sep 07 23:59:59 EDT 1969	Passed
03	OpenmrsUtil	This method will take a date convert it a calender, then shift the time to be the last second of the day on that date.	getLastMomentOfDay	0	Wed Dec 31 23:59:59 EST 1969	getLastMomentOfDayDriver	Wed Dec 31 23:59:59 EST 1969	Passed
04	OpenmrsUtil	This method will take a date convert it a calender, then shift the time to be the last second of the day on that date.	getLastMomentOfDay	1000000000000	Sat Sep 08 23:59:59 EDT 2001	getLastMomentOfDayDriver	Sat Sep 08 23:59:59 EDT 2001	Passed
05	OpenmrsUtil	This method will take a date convert it a calender, then shift the time to be the last second of the day on that date.	getLastMomentOfDay	914209500000	Sun Dec 20 23:59:59 EST 1998	getLastMomentOfDayDriver	Sun Dec 20 23:59:59 EST 1998	Passed
06	DateUtil	This method will return a Date with the milliseconds tuncated	truncateToSeconds	1234094	Wed Dec 31 19:20:34 EST 1969	DateUtilDriver	Wed Dec 31 19:20:34 EST 1969	Passed
07	DateUtil	This method will return a Date with the milliseconds tuncated	truncateToSeconds	1234094008765343	Sun Oct 28 13:14:15 EDT 303038	DateUtilDriver	Sun Oct 28 13:14:15 EDT 303038	Passed
08	DateUtil	This method will return a Date with the milliseconds tuncated	truncateToSeconds	0	Wed Dec 31 19:00:00 EST 1969	DateUtilDriver	Wed Dec 31 19:00:00 EST 1969	Passed
09	DateUtil	This method will return a Date with the milliseconds tuncated	truncateToSeconds	-847350384	Sun Dec 21 23:37:30 EST 1969	DateUtilDriver	Sun Dec 21 23:37:30 EST 1969	Passed
10	DateUtil	This method will return a Date with the milliseconds tuncated	truncateToSeconds	-949474704827	Thu Nov 30 12:01:36 EST 1939	DateUtilDriver	Thu Nov 30 12:01:36 EST 1939	Passed
11	OpenmrsUtil	This method returns true if a string only contains digits	containsOnlyDigits	808760	true	containsOnlyDigitsDriver	true	Passed
12	OpenmrsUtil	This method returns true if a string only contains digits	containsOnlyDigits	-842	false	containsOnlyDigitsDriver	false	Passed
13	OpenmrsUtil	This method returns true if a string only contains digits	containsOnlyDigits	8940	true	containsOnlyDigitsDriver	true	Passed
14	OpenmrsUtil	This method returns true if a string only contains digits	containsOnlyDigits	34hi80	false	containsOnlyDigitsDriver	false	Passed
15	OpenmrsUtil	This method returns true if a string only contains digits	containsOnlyDigits	60 78	false	containsOnlyDigitsDriver	false	Passed
16	OpenmrsUtil	This method turns a long into an integer, as long as it does not exceed the int memory limit	convertToInteger	600	600	convertToIntegerDriver	600	Passed
17	OpenmrsUtil	This method turns a long into an integer, as long as it does not exceed the int memory limit	convertToInteger	6000000000	null	convertToIntegerDriver	null	Passed
18	OpenmrsUtil	This method turns a long into an integer, as long as it does not exceed the int memory limit	convertToInteger	-83878527402	null	convertToIntegerDriver	null	Passed
19	OpenmrsUtil	This method turns a long into an integer, as long as it does not exceed the int memory limit	convertToInteger	2147483647	2147483647	convertToIntegerDriver	2147483647	Passed
20	OpenmrsUtil	This method turns a long into an integer, as long as it does not exceed the int memory limit	convertToInteger	-2147483648	-2147483648	convertToIntegerDriver	-2147483648	Passed
21	OpenmrsUtil	This method returns true if a string has upper and lower case letters	containsUpperAndLowerCase	HELLO THERE	false	containsUpperAndLowerCaseDriver	false	Passed
22	OpenmrsUtil	This method returns true if a string has upper and lower case letters	containsUpperAndLowerCase	how are you today	false	containsUpperAndLowerCaseDriver	false	Passed
23	OpenmrsUtil	This method returns true if a string has upper and lower case letters	containsUpperAndLowerCase	I'm well how are YOU?	true	containsUpperAndLowerCaseDriver	true	Passed
24	OpenmrsUtil	This method returns true if a string has upper and lower case letters	containsUpperAndLowerCase	Hi32	true	containsUpperAndLowerCaseDriver	true	Passed
25	OpenmrsUtil	This method returns true if a string has upper and lower case letters	containsUpperAndLowerCase	hejfkjsd83eh	false	containsUpperAndLowerCaseDriver	false	Passed

Updated on 11/12/2020 01:12:57 PM EST by meglucas