CSCI362: Software Engineering

Automated Testing Framework for OpenMRS

Ricky Ramos, Matthew Schwarz, Matthew Kay

About OpenMRS

OpenMRS is a medical record system written in Java, which focuses on giving health care providers a free and customizable electronic medical record system.

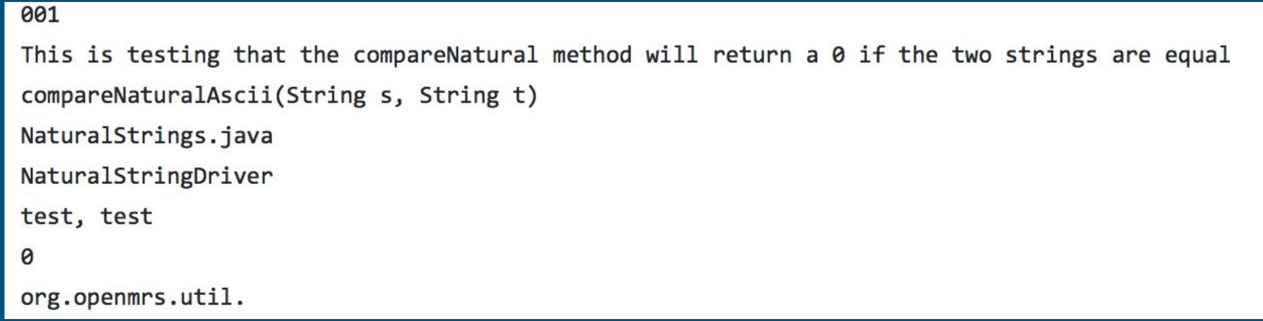
OpenMRS's mission statement is to "improve healthcare delivery in resource-constrained environments, by creating a robust, scalable, user-driven, open source medical record system platform".

About Our Testing Framework

- Our testing framework is executed by a bash script named runAllTests.sh which is located in the scripts directory
- The script parses the test case text files (located in testCases directory) and grabs the provided information from the text files
- The script then uses this information to compile and run the test drivers, which run the actual tests for the methods that are being tested.
- The result is then retrieved from the drivers, and the result, along with the information previously grabbed, is input into a HTML table and displayed in a browser to the user.

Test Case Input

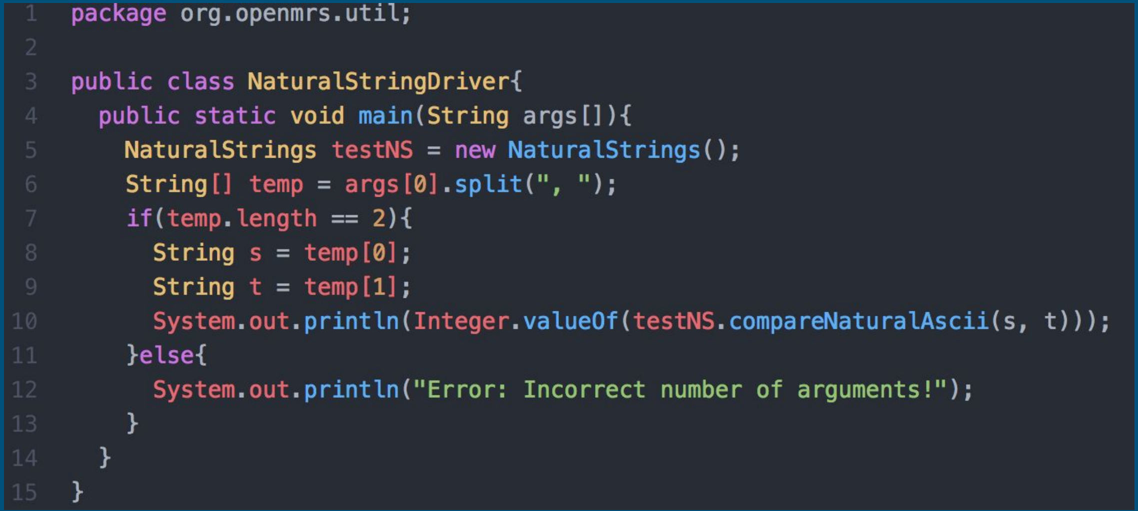
1. Test Case ID
2. Requirement being tested
3. Component being tested
4. Class
5. Driver being used
6. Test Input(s)
7. Expected Result
8. package



```
001
This is testing that the compareNatural method will return a 0 if the two strings are equal
compareNaturalAscii(String s, String t)
NaturalStrings.java
NaturalStringDriver
test, test
0
org.openmrs.util.
```

Driver for first 5 test cases: NaturalStringDriver

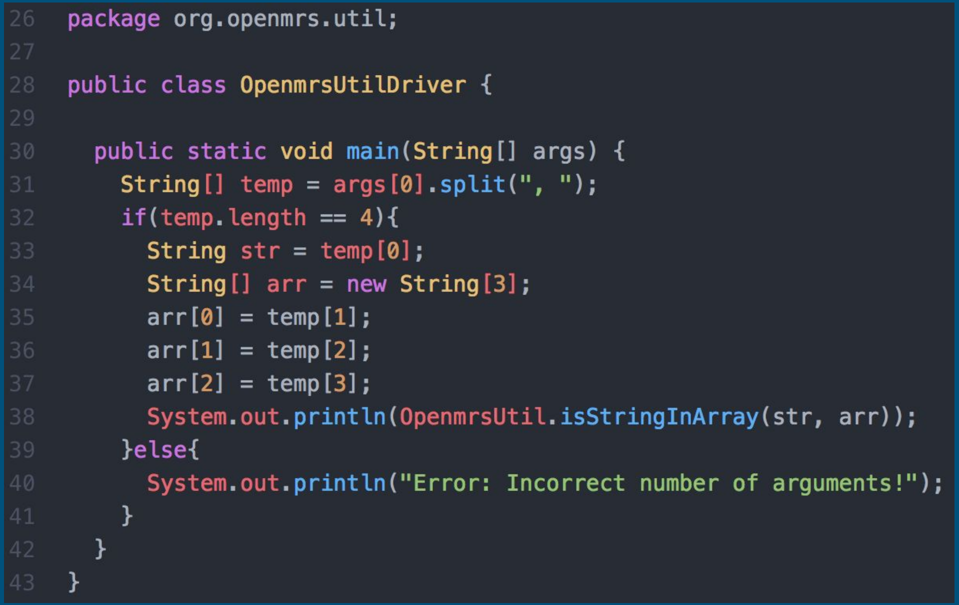
- This driver runs the first five test cases using the method compareNaturalAscii(String s, String t).
- The method the driver is testing takes two strings as inputs and returns 0 if the two strings are identical, returns a positive number if the first string lexicographically follows the second string (number is based on distance of ASCII characters), and a negative number if the first string lexicographically precedes the second string.
- The method is case sensitive.



```java
package org.openmrs.util;

public class NaturalStringDriver{
  public static void main(String args[]){
    NaturalStrings testNS = new NaturalStrings();
    String[] temp = args[0].split(", ");
    if(temp.length == 2){
      String s = temp[0];
      String t = temp[1];
      System.out.println(Integer.valueOf(testNS.compareNaturalAscii(s, t)));
    }else{
      System.out.println("Error: Incorrect number of arguments!");
    }
  }
}
```

Driver for next 5 Test Cases: OpenmrsUtilDriver

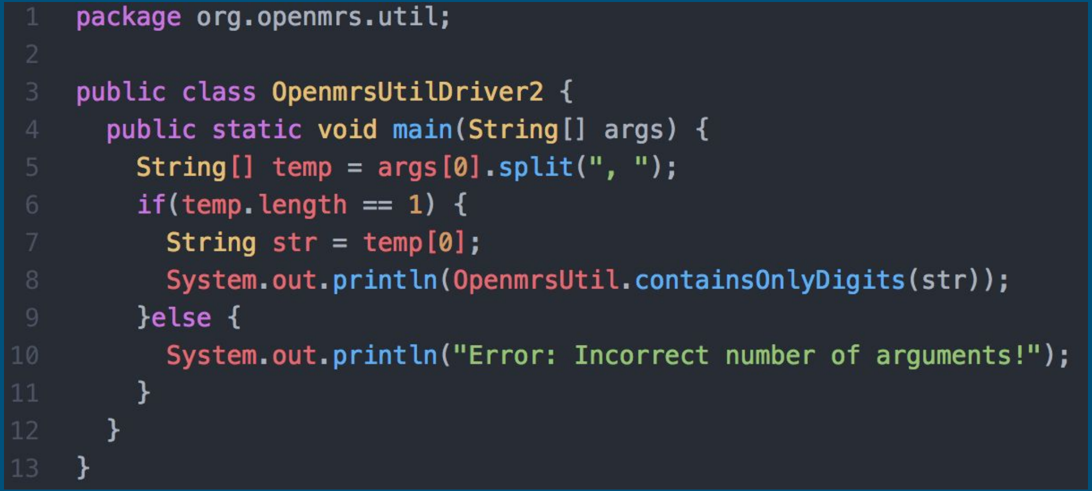
- This driver runs test cases 006-010 using the method isStringInArray(String str, String[] arr).
- The method the driver is testing takes a string and a string array as inputs, and tests if the input string is in the array.
- If the string is in the array, the method returns true.
- If the string is not in the array, the method returns false.



```java
package org.openmrs.util;

public class OpenmrsUtilDriver {

  public static void main(String[] args) {
    String[] temp = args[0].split(", ");
    if(temp.length == 4){
      String str = temp[0];
      String[] arr = new String[3];
      arr[0] = temp[1];
      arr[1] = temp[2];
      arr[2] = temp[3];
      System.out.println(OpenmrsUtil.isStringInArray(str, arr));
    }else{
      System.out.println("Error: Incorrect number of arguments!");
    }
  }
}
```

Driver for next 5 Test Cases: OpenmrsUtilDriver2

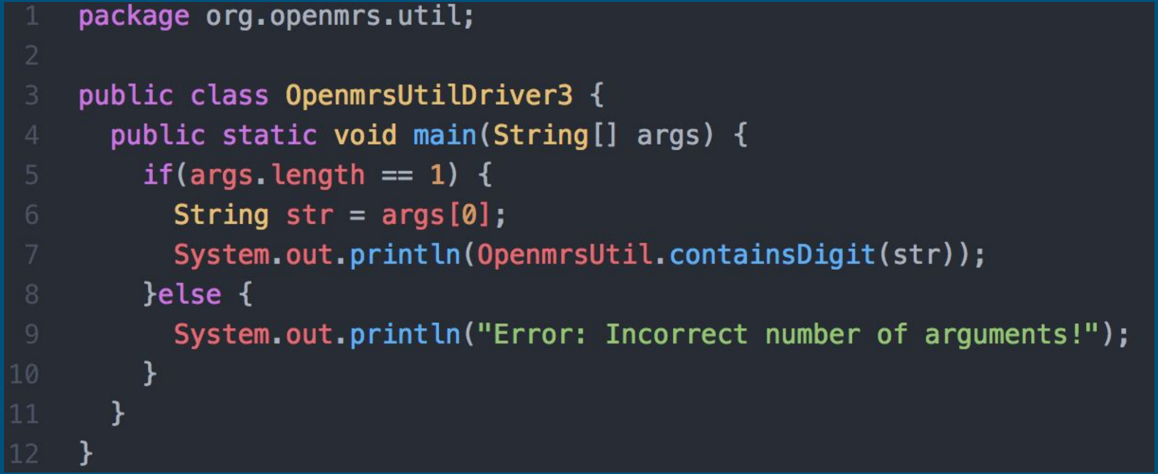
- This driver runs test cases 011-015 using the method containsOnlyDigits(String str).
- This method takes a string in as input and checks if the string is made up of only numerical digits
- If the string contains only digits, then it returns true.
- If the string contains letters, or other items that are not digits, then it returns false.



```java
package org.openmrs.util;

public class OpenmrsUtilDriver2 {
  public static void main(String[] args) {
    String[] temp = args[0].split(", ");
    if(temp.length == 1) {
      String str = temp[0];
      System.out.println(OpenmrsUtil.containsOnlyDigits(str));
    }else {
      System.out.println("Error: Incorrect number of arguments!");
    }
  }
}
```

Driver for Next 5 Test Cases: OpenmrsUtilDriver3

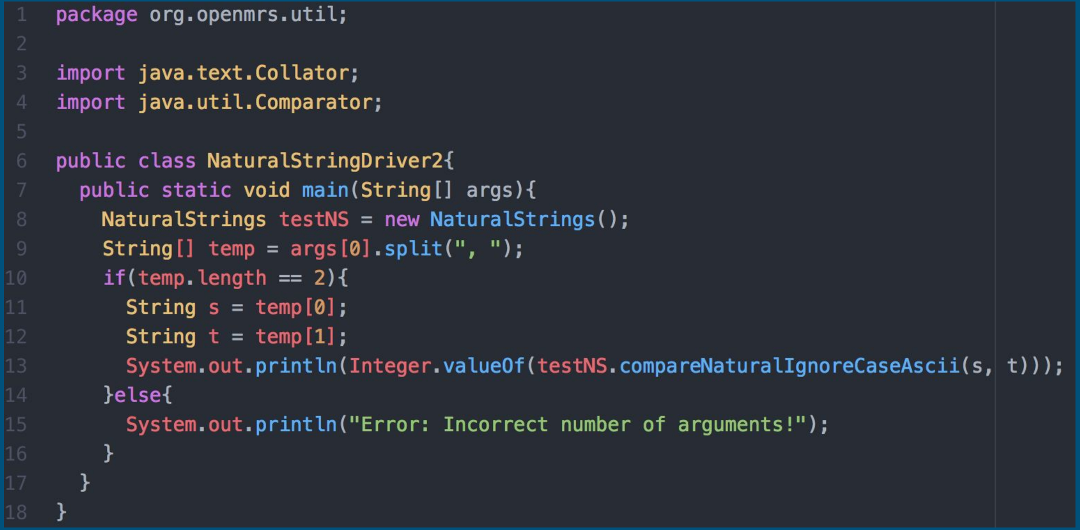
- This driver runs test cases 016-020 using the method containsDigit(String str).
- This method takes a string in as input and tests whether or not the string contains a digit in it somewhere.
- The method will return true if the string does contain at least one digit.
- The method will return false if the string does not contain any digits in it.



```java
package org.openmrs.util;

public class OpenmrsUtilDriver3 {
  public static void main(String[] args) {
    if(args.length == 1) {
      String str = args[0];
      System.out.println(OpenmrsUtil.containsDigit(str));
    }else {
      System.out.println("Error: Incorrect number of arguments!");
    }
  }
}
```

Driver for last 5 Test Cases: NaturalStringDriver2

- This driver runs test cases 021-025 using the method compareNaturalIgnoreCaseAscii(String s, String t).
- This method is very similar to the first driver (NaturalStringDriver) and performs the same function based on two input strings, with the exception that this method does not take the case of letters into count, where as NaturalStringDriver is case sensitive.
- This means that inputs of tESt and test would return 0 meaning they are the same.



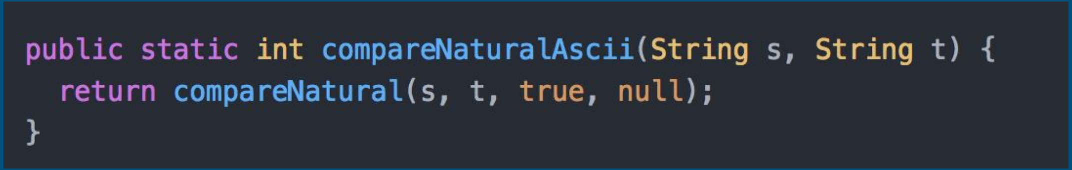
```java
package org.openmrs.util;

import java.text.Collator;
import java.util.Comparator;

public class NaturalStringDriver2{
  public static void main(String[] args){
    NaturalStrings testNS = new NaturalStrings();
    String[] temp = args[0].split(", ");
    if(temp.length == 2){
      String s = temp[0];
      String t = temp[1];
      System.out.println(Integer.valueOf(testNS.compareNaturalIgnoreCaseAscii(s, t)));
    }else{
      System.out.println("Error: Incorrect number of arguments!");
    }
  }
}
```

Team 404-Error Openmrs Testing Results

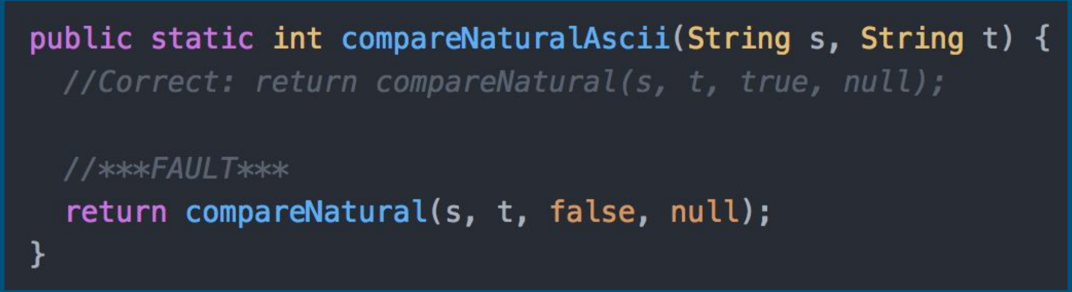
Test	Requirement	Method	Class	Driver	Inputs	Expected Output	Result	
Test #001	This is testing that the compareNatural method will return a 0 if the two strings are equal	compareNaturalAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver	test, test	0	0	Pass
Test #002	Testing that the compareNatural method will return a negative number if s lexicographically precedes t	compareNaturalAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver	test, white	-3	-3	Pass
Test #003	Testing that the compareNatural method will return a positive number if s lexicographically follows t	compareNaturalAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver	test, string	1	1	Pass
Test #004	Testing that the compareNatural method will return a positive number if s lexicographically follows t	compareNaturalAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver	test, check	17	17	Pass
Test #005	Testing that the compareNatural method will return a positive number if s lexicographically follows t	compareNaturalAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver	test, pneumonia	4	4	Pass
Test #006	This is testing that the correct boolean value will be printed from the isStringInArray(String str, String[] arr) method in OpenmrsUtil.java	isStringInArray(String str, String[] arr)	OpenmrsUtil.java	OpenmrsUtilDriver	random, random, test, computer	true	true	Pass
Test #007	This is testing that the correct boolean value will be printed from the isStringInArray(String str, String[] arr) method in OpenmrsUtil.java	isStringInArray(String str, String[] arr)	OpenmrsUtil.java	OpenmrsUtilDriver	test, random, test, computer	true	true	Pass
Test #008	This is testing that the correct boolean value will be printed from the isStringInArray(String str, String[] arr) method in OpenmrsUtil.java	isStringInArray(String str, String[] arr)	OpenmrsUtil.java	OpenmrsUtilDriver	computer, random, test, computer	true	true	Pass
Test #009	This is testing that the correct boolean value will be printed from the isStringInArray(String str, String[] arr) method in OpenmrsUtil.java	isStringInArray(String str, String[] arr)	OpenmrsUtil.java	OpenmrsUtilDriver	allergy, random, test, computer	false	false	Pass
Test #010	This is testing that the correct boolean value will be printed from the isStringInArray(String str, String[] arr) method in OpenmrsUtil.java	isStringInArray(String str, String[] arr)	OpenmrsUtil.java	OpenmrsUtilDriver	medicine, random, test, computer	false	false	Pass
Test #011	This is testing that the correct boolean value will be printed from the containsOnlyDigits(String str) method in OpenmrsUtil.java	containsOnlyDigits(String str)	OpenmrsUtil.java	OpenmrsUtilDriver2	12345	true	true	Pass
Test #012	This is testing that the correct boolean value will be printed from the containsOnlyDigits(String str) method in OpenmrsUtil.java	containsOnlyDigits(String str)	OpenmrsUtil.java	OpenmrsUtilDriver2	67930	true	true	Pass
Test #013	This is testing that the correct boolean value will be printed from the containsOnlyDigits(String str) method in OpenmrsUtil.java	containsOnlyDigits(String str)	OpenmrsUtil.java	OpenmrsUtilDriver2	99999999	true	true	Pass
Test #014	This is testing that the correct boolean value will be printed from the containsOnlyDigits(String str) method in OpenmrsUtil.java	containsOnlyDigits(String str)	OpenmrsUtil.java	OpenmrsUtilDriver2	This is a string!	false	false	Pass
Test #015	This is testing that the correct boolean value will be printed from the containsOnlyDigits(String str) method in OpenmrsUtil.java	containsOnlyDigits(String str)	OpenmrsUtil.java	OpenmrsUtilDriver2	123412341234a	false	false	Pass
Test #016	This is testing that the correct boolean value will be printed from the containsDigit(String str) method in OpenmrsUtil.java	containsDigit(String str)	OpenmrsUtil.java	OpenmrsUtilDriver3	asdf3	true	true	Pass
Test #017	This is testing that the correct boolean value will be printed from the containsDigit(String str) method in OpenmrsUtil.java	containsDigit(String str)	OpenmrsUtil.java	OpenmrsUtilDriver3	ddddddd9	true	true	Pass
Test #018	This is testing that the correct boolean value will be printed from the containsDigit(String str) method in OpenmrsUtil.java	containsDigit(String str)	OpenmrsUtil.java	OpenmrsUtilDriver3	11234	true	true	Pass
Test #019	This is testing that the correct boolean value will be printed from the containsDigit(String str) method in OpenmrsUtil.java	containsDigit(String str)	OpenmrsUtil.java	OpenmrsUtilDriver3	string	false	false	Pass
Test #020	This is testing that the correct boolean value will be printed from the containsDigit(String str) method in OpenmrsUtil.java	containsDigit(String str)	OpenmrsUtil.java	OpenmrsUtilDriver3	asdfasdlkajsdf	false	false	Pass
Test #021	This is testing that the compareNatural method will return a 0 if the two strings are equal	compareNaturalIgnoreCaseAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver2	test, test	0	0	Pass
Test #022	Testing that the compareNatural method will return a negative number if s lexicographically precedes t	compareNaturalIgnoreCaseAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver2	Test, whITe	-3	-3	Pass
Test #023	Testing that the compareNatural method will return a positive number if s lexicographically follows t	compareNaturalIgnoreCaseAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver2	test, strING	1	1	Pass
Test #024	Testing that the compareNatural method will return a positive number if s lexicographically follows t	compareNaturalIgnoreCaseAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver2	TEST, Check	17	17	Pass
Test #025	Testing that the compareNatural method will return a positive number if s lexicographically follows t	compareNaturalIgnoreCaseAscii(String s, String t)	NaturalStrings.java	NaturalStringDriver2	tESt, pneUMOnia	4	4	Pass

Fault Injection #1

- Our first fault injection was for the compareNaturalAscii(String s, String t) method.
- The third (boolean) parameter in the method called within compareNaturalAscii(String s, String t) was switched from true to false. That parameter deals with case sensitivity (true if you want it, false if you don't).
- The fault, in this case, had no effect on our test results because all of our test cases used input strings that were all already lowercase to begin with.



```java
public static int compareNaturalAscii(String s, String t) {
  return compareNatural(s, t, true, null);
}
```



```java
public static int compareNaturalAscii(String s, String t) {
  //Correct: return compareNatural(s, t, true, null);

  //***FAULT***
  return compareNatural(s, t, false, null);
}
```

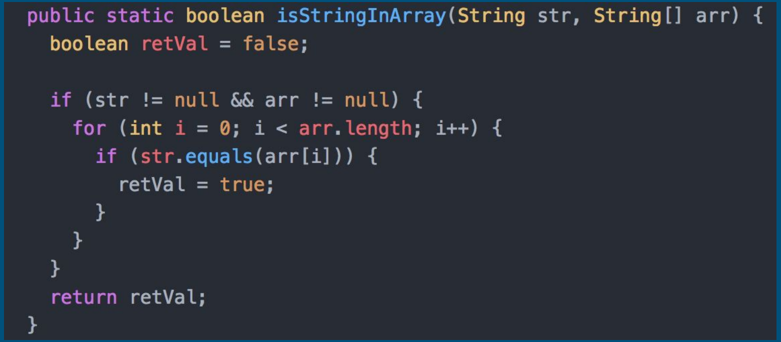
Test Cases: 001 - 005
Test Case Results:
    Original: 5 Passed / 0 Failed
    After Fault: 5 Passed / 0 Failed

Fault Injection #2

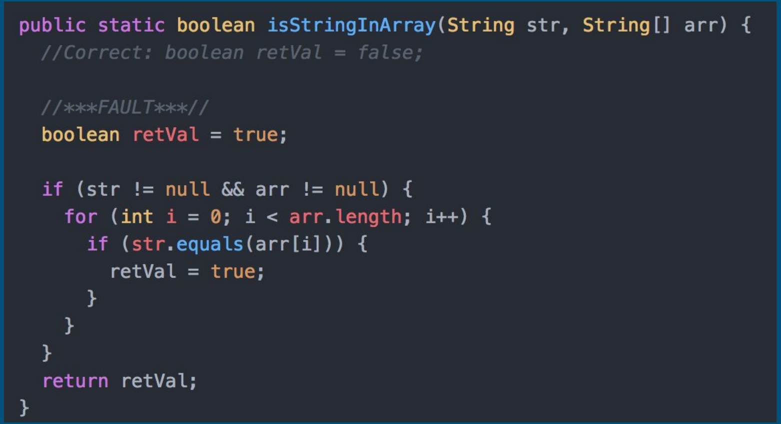
- Our second fault injection was for the isStringInArray(String str, String[] arr) method.

- The initial return value (retVal) was changed from false to true. Essentially, this method will always return true, indicating that the input String is always within the input String array.

- This fault caused 2 test cases to fail.



```java
public static boolean isStringInArray(String str, String[] arr) {
  boolean retVal = false;

  if (str != null && arr != null) {
    for (int i = 0; i < arr.length; i++) {
      if (str.equals(arr[i])) {
        retVal = true;
      }
    }
  }
  return retVal;
}
```



```java
public static boolean isStringInArray(String str, String[] arr) {
  //Correct: boolean retVal = false;

  //***FAULT***//
  boolean retVal = true;

  if (str != null && arr != null) {
    for (int i = 0; i < arr.length; i++) {
      if (str.equals(arr[i])) {
        retVal = true;
      }
    }
  }
  return retVal;
}
```

Test Cases: 006 - 010

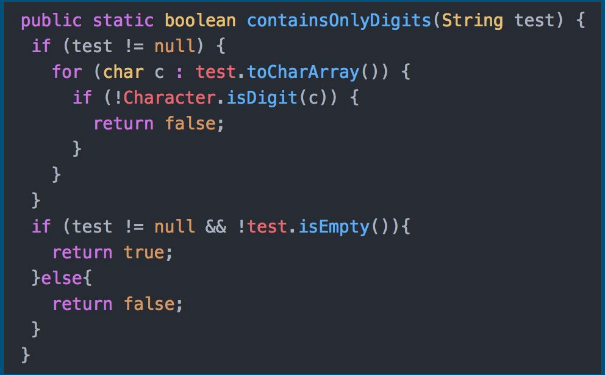
Test Case Results:

Original: 5 Passed / 0 Failed

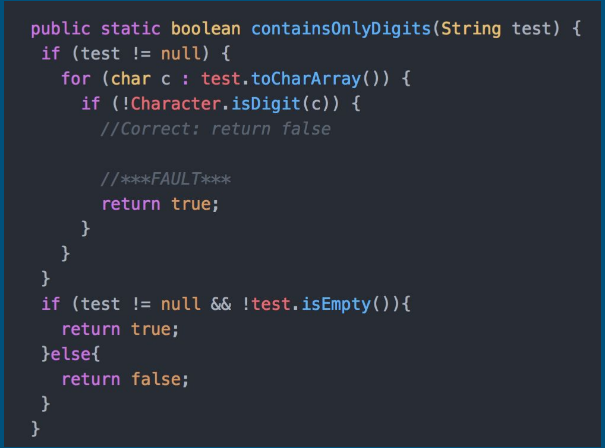
After Fault: 3 Passed (006,007,008) / 2 Failed (009,010)

Fault Injection #3

- Our third fault injection was for the containsOnlyDigits(String test) method.
- The method was changed to return true if a character that wasn't a digit was found within the string.
- This fault caused 2 test cases to fail.



```java
public static boolean containsOnlyDigits(String test) {
 if (test != null) {
  for (char c : test.toCharArray()) {
   if (!Character.isDigit(c)) {
    return false;
   }
  }
 }
 if (test != null && !test.isEmpty()){
  return true;
 }else{
  return false;
 }
}
```

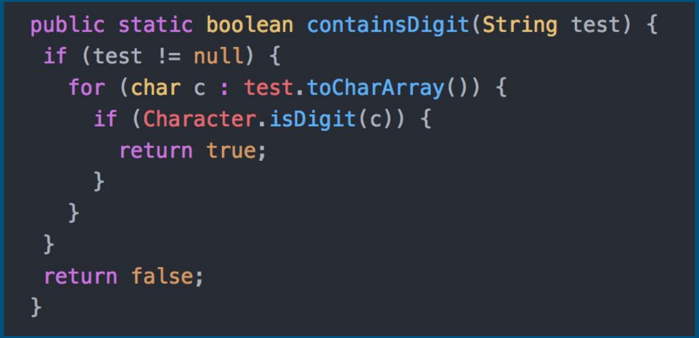


```java
public static boolean containsOnlyDigits(String test) {
 if (test != null) {
  for (char c : test.toCharArray()) {
   if (!Character.isDigit(c)) {
    //Correct: return false

    //***FAULT***
    return true;
   }
  }
 }
 if (test != null && !test.isEmpty()){
  return true;
 }else{
  return false;
 }
}
```

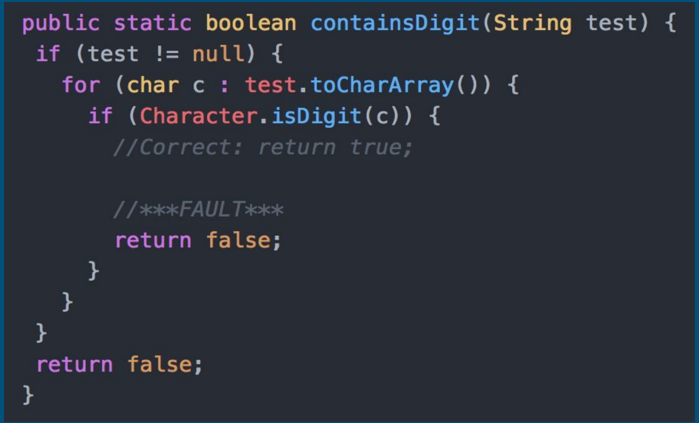
Test Cases: 011 - 015
Test Case Results:
Original: 5 Passed / 0 Failed
After Fault: 3 Passed (011,012,013) / 2 Failed (014,015)

Fault Injection #4

- Our fourth fault injection was for the containsDigit(String test) method.
- The method was changed to return false if a character is a digit. The method will always return false.
- This fault caused 3 of our test cases to fail.



```java
public static boolean containsDigit(String test) {
 if (test != null) {
   for (char c : test.toCharArray()) {
     if (Character.isDigit(c)) {
       return true;
     }
   }
 }
 return false;
}
```



```java
public static boolean containsDigit(String test) {
 if (test != null) {
   for (char c : test.toCharArray()) {
     if (Character.isDigit(c)) {
       //Correct: return true;

       //***FAULT***
       return false;
     }
   }
 }
 return false;
}
```

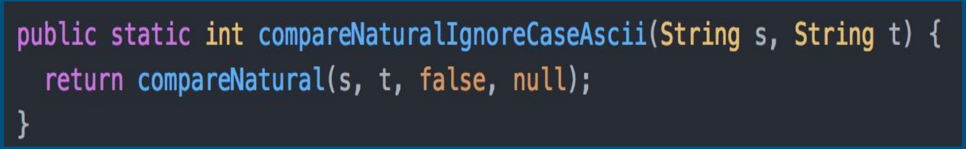
Test Cases: 016 - 020
Test Case Results:
Original: 5 Passed / 0 Failed
After Fault: 2 Passed (019,020) / 3 Failed (016,017,018)

Fault Injection #5

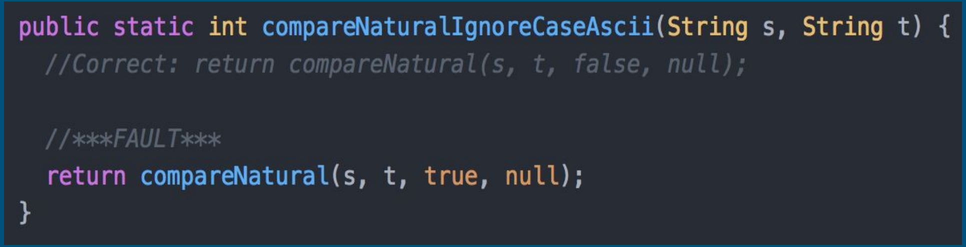
- Our fifth fault injection was for the compareNaturalIgnoreCaseAscii(String s, String t) method.

- The third (boolean) parameter in the method called within compareNaturalIgnoreCaseAscii(String s, String t) was switched from false to true. That parameter deals with case sensitivity (true if you want it, false if you don't).

- This fault caused 1 of our test cases to fail.



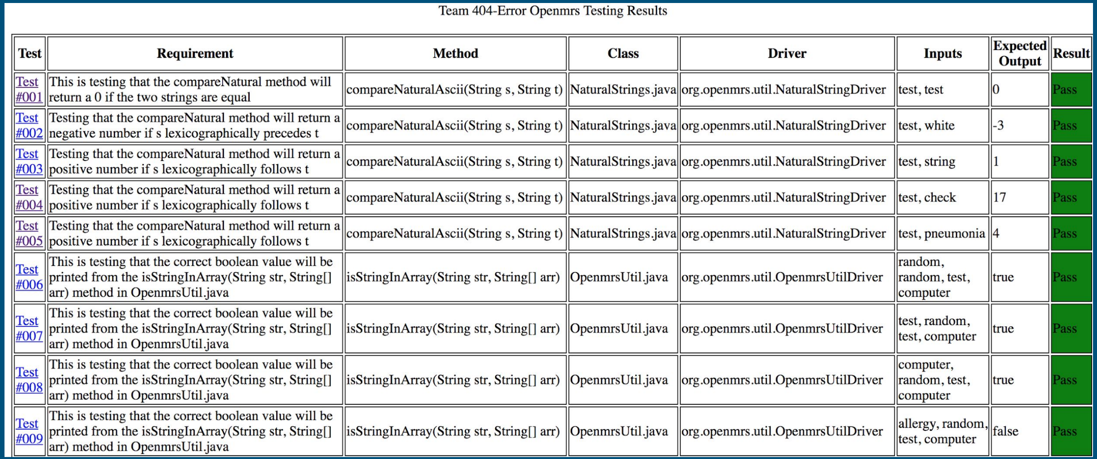
```java
public static int compareNaturalIgnoreCaseAscii(String s, String t) {
  return compareNatural(s, t, false, null);
}
```



```java
public static int compareNaturalIgnoreCaseAscii(String s, String t) {
  //Correct: return compareNatural(s, t, false, null);

  //***FAULT***
  return compareNatural(s, t, true, null);
}
```

Test Cases: 021 - 025
Test Case Results:
    Original: 5 Passed / 0 Failed
    After Fault: 4 Passed (021,023, 024, 025) / 1 Failed (022)

HTML



Team 404-Error Openmrs Testing Results

Test	Requirement	Method	Class	Driver	Inputs	Expected Output	Result
Test #001	This is testing that the compareNatural method will return a 0 if the two strings are equal	compareNaturalAscii(String s, String t)	NaturalStrings.java	org.openmrs.util.NaturalStringDriver	test, test	0	Pass
Test #002	Testing that the compareNatural method will return a negative number if s lexicographically precedes t	compareNaturalAscii(String s, String t)	NaturalStrings.java	org.openmrs.util.NaturalStringDriver	test, white	-3	Pass
Test #003	Testing that the compareNatural method will return a positive number if s lexicographically follows t	compareNaturalAscii(String s, String t)	NaturalStrings.java	org.openmrs.util.NaturalStringDriver	test, string	1	Pass
Test #004	Testing that the compareNatural method will return a positive number if s lexicographically follows t	compareNaturalAscii(String s, String t)	NaturalStrings.java	org.openmrs.util.NaturalStringDriver	test, check	17	Pass
Test #005	Testing that the compareNatural method will return a positive number if s lexicographically follows t	compareNaturalAscii(String s, String t)	NaturalStrings.java	org.openmrs.util.NaturalStringDriver	test, pneumonia	4	Pass
Test #006	This is testing that the correct boolean value will be printed from the isStringInArray(String str, String[] arr) method in OpenmrsUtil.java	isStringInArray(String str, String[] arr)	OpenmrsUtil.java	org.openmrs.util.OpenmrsUtilDriver	random, random, test, computer	true	Pass
Test #007	This is testing that the correct boolean value will be printed from the isStringInArray(String str, String[] arr) method in OpenmrsUtil.java	isStringInArray(String str, String[] arr)	OpenmrsUtil.java	org.openmrs.util.OpenmrsUtilDriver	test, random, test, computer	true	Pass
Test #008	This is testing that the correct boolean value will be printed from the isStringInArray(String str, String[] arr) method in OpenmrsUtil.java	isStringInArray(String str, String[] arr)	OpenmrsUtil.java	org.openmrs.util.OpenmrsUtilDriver	computer, random, test, computer	true	Pass
Test #009	This is testing that the correct boolean value will be printed from the isStringInArray(String str, String[] arr) method in OpenmrsUtil.java	isStringInArray(String str, String[] arr)	OpenmrsUtil.java	org.openmrs.util.OpenmrsUtilDriver	allergy, random, test, computer	false	Pass

- We kept our HTML table nice and simple.
- We had eight columns (Test, Requirement, Method, Class, Driver, Inputs, Expected Output, and Result)
- For our results column if the test case passed, the box turned green and if it failed it turned red.
- For each test case, we created a link that brought you to the test case description.

Issues And Struggles

- The first issue that our group encountered was getting our original project, TEAMMATES, to work due to the massive amount of external packages and libraries it used. This lead our group to transition to working on OpenMRS instead.
- The second major issue that we had was getting the java classes inside of the OpenMRS project that we needed to run our drivers to compile and run. We had issues with this for a while, and solved it by realizing that we had to compile from the directory containing the package that the classes were in, not just the classes themselves.
- The third issue we ran into was compiling the classes without directly stating the directory path in the script. We talked to Professor Bowring about this and he was extremely helpful. Afterwards, we were able to fix the issue by compiling from the information in the test case text files.

What We Learned

- Our team didn’t have much experience with writing bash scripts, so this assignment helped us to learn how to do that, and how helpful it can be to automate things with scripts.
- Our team gained extensive knowledge in writing tests to test code, and with developing an appropriate time table and testing plan.
- We learned how to correctly compile and run java files from the terminal, and from a script.
- We learned a lot about the importance of having a job for every team-member to perform, and the importance of planning ahead and not procrastinating in group projects.
- Most importantly, we learned that proper communication is one of the most important things in group work.

END