Unit test/TDD Test lab

Names:
Linnea Dahlgren
Gavin Nejsum

Date: 17-12-2020

Contents

Part 1

Question 3a1

The tool for code coverage measures branches and individual instruction on the java file it also take on the different path in the if/else method and the condition on “normal== null”. So, I would say it covers all 4(Statements, branches, path, and condition).

Question 3c1

When we did the final testing, I got 100% code coverage.

Question 3c2

Defect 1:

In the functions, enrov and derov. the second append will append a big “O” which is not correct according to www.rövarspråket.se where it says it is supposed to be a small “o”. For example if you translate the name “Linnea” in the program it will be translated as “LOLinonnonea” meanwhile according to the website it is supposed to translate too “LoLinonnonea”

This was the case for both translating too rövarspråket and from rövarspråket.

When translating from rövarspråket back to normal text it will only detect upper-case “O” so if you paste in an accurate translation too rövarspråket, like LoLinonnonea it will translate it to LoLinnea because it doesn’t remove the lower-case “o” properly.

Defect 2:

When testing for lower case letters the small case “g“ was missing and not working. So, for example it would not translate “gavin” properly. The name “gavin” should translate too “gogavovinon” but instead translated too “gavovinon”

Defect 3:

For the upper case letters the letter ”D” was missing. When you tried to translate a word that had a upper case “D”, for example “Destiny” it would not translate correctly. The word “Destiny” translated to “Desostotinony” while the correct translation is “DoDesostotinony”

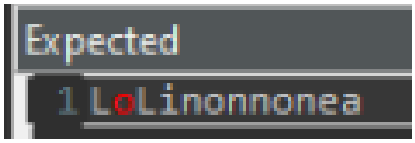
This defect would also apply to the “derov” method. But after finding the error while testing “enrov” when I fixed it I also fixed it for “derov”.

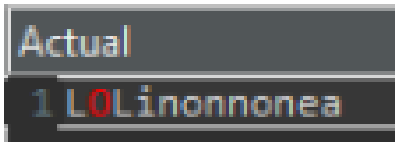
Question 3c3

Defect 1.1:

Test Case: `assertEquals("LoLinonnonea”)test.enrov("Linnea"));`

Result:



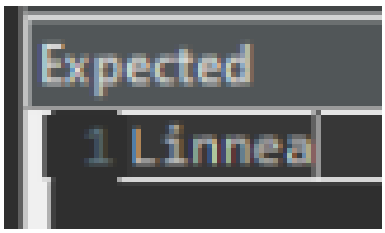


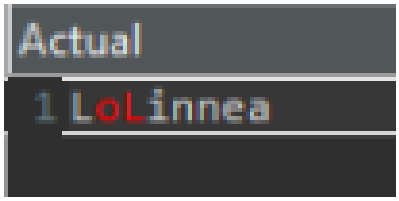
2nd letter differs between actual and expected

Defect 1.2

Test Case: `assertEquals("Linnea",test.derov("LoLinonnonea_"));`

Result:





Defect 2:

Test Case:

```
assertEquals("abobcocdodefofgoghohijojkoklolmomnonopopqoqrorsostotuvovwowxoxy",tes
t.enrov("abcdefghijklmnopqrstuvwxy"));
```

Result:

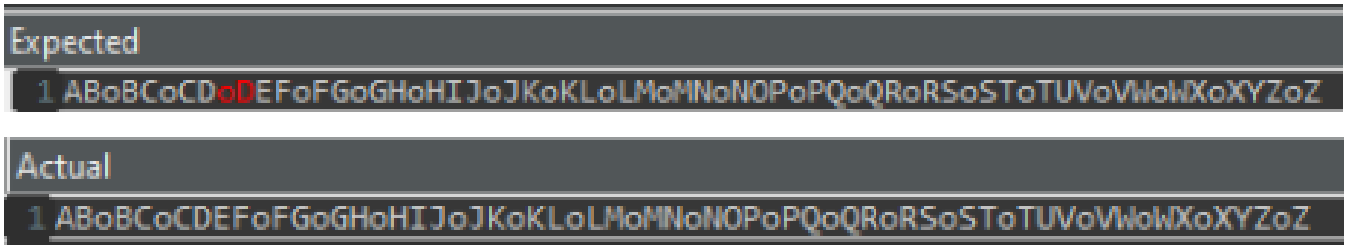
After the “g” there is no “og” in actual.

Defect 3:

Test Case:

```
assertEquals("ABoBCoCDoDEFoFGoGHoHIJoJKoKLoLNoNMoMOPoPQoQRoRSoSToTUVoVWoWYZoZ",tes
t.enrov("ABCDEFGHIJKLNMOPQRSTUVWYZ"));
```

Result:



After the “D” there is no “oD” in actual.

Question 3c4

Defect 1.1:

To correct defect 1, we changed, in the function “enrov”, the upper case “O” in the append on row 25 to a lower case “o”.

Defect 1.2:

To correct defect 1, we changed, in the function “derov”, the upper case “O” in the append on row 49 to a lower case “o”.

Defect 2:

For this defect we only had to add the missing letter, lower case “g”, into the string called “lower_consonants” (row 7).

Defect 3:

Like defect 2, we needed to add the missing later, in this case upper-case “D”, to the string called “upper_consonants” (row 8)

Question 3c5

After the defects were fixed all the passed.

Part 2

Question 4c1

-How many state transitions (compare 0-switch, 1-switch, etc. mentioned in the lecture) are reasonable to test in this case?

To call the clock class well tested and make sure it works properly a reasonable amount of state transitions would be 12:
6 state transitions for the legal transitions.
6 state transitions for the illegal transitions.

Question 4f1

We got 100% coverage as shown in the picture below

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
clock	100.0 %	705	0	705
Time.java	100.0 %	77	0	77
Date.java	100.0 %	80	0	80
Clock.java	100.0 %	138	0	138
TestClock.java	100.0 %	410	0	410

Figure 1 -Coverage from tests for the class Clock