
Testing und Debugging: Übung 08

Abgabetermin siehe Canvas

1 Systematische Fehlersuche

In der Vorlesung haben uns mit einigen Fehlern in unten stehendem Programm beschäftigt und diese gefixt.

Gefixte Programm aus der Vorlesung ('remove_html_markup.py')

```

1 def remove_html_markup(s):
2     tag = False
3     quote = False
4     out = ""
5     for c in s:
6         if c == '<' and not quote:
7             tag = True
8         elif c == '>' and not quote:
9             tag = False
10        elif (c == '"' or c == "'") and tag:
11            quote = not quote
12        elif not tag:
13            out = out + c
14    return out

```

Beim Testen dieses Programms sind die folgenden zwei Fehlverhalten aufgefallen:

Eingabe	Erwartet	Ausgabe	Erwartung erfüllt?
5>3	5>3	53	✗
<input value="5">'3"> leer		'3"	✗

Getestet wurde mit folgendem Aufruf:

```
print(remove_html_markup("5>3"))
print(remove_html_markup('"""<input value="5">'3">"""))
```



Achtung

Benutzen Sie bitte die Implementation in der Datei `remove_html_markup.py` für Ihre Experiment bei den schriftlichen Aufgaben.

1.1 Fixen Sie das Fehlverhalten für die Eingabe 5>3 (60 Punkte)

In der Vorlesung haben Sie gelernt, dass das systematische Verfolgen der Änderungen der Zustände (Variablenbelegung) des Programms Rückschlüsse auf den Fehler im Quellcode geben kann.

Gehen Sie dabei das Programm Schritt für Schritt durch und notieren Sie das jeweils ausgeführte Kommando. In der jeweils nachfolgenden Zeile notieren Sie wie sich die Variablen durch diesen Programmschritt geändert haben.

Variablen, die noch nicht belegt wurden bekommen ein '?' als Eintrag.

1.1.1 Vervollständigen Sie die folgende Tabelle (30 Punkte)

[illegible]

1.1.2 In welcher der folgenden Aussagen treffen zu (4 + 3 + 3 Punkte):

1. Jede Infektion endet in einem Fehlverhalten.
2. Jede Infektion kann zu einem Defekt zurückverfolgt werden, der die Infektion verursacht hat.
3. Jeder Defekt endet in einem Fehlverhalten.

Geben Sie zudem für die **nicht zutreffenden** Aussagen jeweils ein Beispiel an, welches Ihre Antwort untermauert.

[illegible]

1.1.3 Markieren Sie in der Tabelle aus 1.1.1 den Defekt, die Infektion und zeigen Sie, wie sich die Infektion bis zum Fehlverhalten hin ausbreitet. (10 Punkte)

Achten Sie darauf dass klar zu Erkennen ist, was der Defekt, die Infektion und deren Verlauf sowie das Fehlverhalten ist.

1.1.4 Wie ist der Defekt zu beheben? (10 Punkte)

[illegible]

1.2 Fixen Sie das Fehlverhalten für die Eingabe `<input value="5">'3'` (50 Punkte)

Als zweite Möglichkeit haben Sie in der Vorlesung die wissenschaftliche Methode kennengelernt.

Bei der wissenschaftlichen Methode gilt es Hypothesen aufzustellen und anhand der Hypothesen Voraussagen zu treffen. Diese Voraussagen müssen dann im Experiment überprüft werden. Das Ergebnis des Experiments entscheidet, ob die Hypothese weiter verfeinert werden muss oder ob die Hypothese verworfen werden muss.

Dieser Zyklus wird solange wiederholt bis, wie in unserem Fall, die Ursache des Fehlverhaltens gefunden und behoben wurde.

1.2.1 Finden Sie mit dieser Methode die Ursache dieses Fehlverhaltens. (40 Punkte)

Notieren Sie dabei, wie in der Vorlesung beschrieben Ihre Hypothesen, Experimente sowie die notwendigen Änderungen im Programmcode für die Durchführung Ihrer Experimente.

- 1.2.2 Durch welches Code Coverage Verfahren, dass Sie im ersten Teil der Vorlesung kennen gelernt haben, können Sie sicher Stellen, dass die Bugfixe für diese beiden Probleme vollständig getestet worden sind? Begründen Sie Ihre Antwort (10 Punkte)

·
·
·
·
·
·
·
·
·
·

2 Programmieraufgabe: Umsetzung der Bugfixe (100 Punkte)

Dateien für die Programmieraufgabe



Grading/init.py	Hinweisdatei für Python: Das Verzeichnis ist ein Modul
Grading/Grading.py	Quellcode der Testklasse, die in <i>run_test.py</i> verwendet wird
uebung08.py *	In dieser Datei müssen Sie die Funktion <code>remove_html_markup()</code> gemäß Aufgabenstellung implementieren
uebung08_unit.py *	In dieser Datei müssen Sie die Unittests Aufgabenstellung implementieren

Die mit [*] markierten Dateien müssen **bearbeitet** und **abgegeben** werden.

2.1 Aufgabenstellung

Setzen Sie die beiden Bugfixe aus dem ersten Teil der Aufgabe in der Datei *uebung08.py* um.

Schreiben Sie für die beiden Bugfixes jeweils einen eigenen Test in der Datei *uebung08_unit.py*

2.2 Bewertung

50 % der Punkte erhalten Sie für die Bugfixes und 50% für die Testfunktionen.

Viel Erfolg
