

Exercise Sheet 4

Birgit Pohl 574353 (MO. 9-11)
Kevin Trogant 572451 (Mo. 15-17)
Ronja Enseleit 572404 (Mo. 15-17)
Dustin Hebecker 571271 (MO. 9-11)



Group: *Iron Man*

19. November 2016

Aufgabe 1

Der Akzeptanztest zielt darauf ab beim Kunden Vertrauen in das Produkt zu erzeugen, während der Funktionstest überprüft ob das entwickelte Produkt den Anforderungen der Anforderungsdefinition entspricht.

Im Gegensatz zum Funktionstest wird der Akzeptanztest vom Kunden in der Umgebung durchgeführt, in der das Produkt eingesetzt werden soll. Da vor dem Akzeptanztest bereits durch den Funktionstest sichergestellt wurde, dass das Produkt der Anforderungsdefinition entspricht, sollte im Akzeptanztest keine Abweichung vom definierten Verhalten mehr entdeckt werden. Das heißt, im Akzeptanztest wird vor allem überprüft ob das Produkt auf die Weise benutzt werden kann, die sich der Kunde vorgestellt hat.

Schlägt der Funktionstest fehl, heißt das, dass das Produkt eine funktionale Anforderung nicht erfüllt (also zum Beispiel eine falsche Ausgabe liefert). Schlägt der Akzeptanztest fehl, heißt das, dass das Produkt vom Kunden nicht benutzt werden kann (zB. weil die Bedienung zu umständlich ist).

Aufgabe 2

Bildung von gültigen und ungültigen Äquivalenzklassen:

Eingabe	gültige Äquivalenzklassen	ungültige Äquivalenzklassen
1. String	1) String aus Buchstaben und/oder Ziffern, der der gleiche String wie der 2. String ist 2) String aus Buchstaben und/oder Ziffern, der ein echter Substring von dem 2. String ist 3) String aus Buchstaben und/oder Ziffern, der kein Substring von dem 2. String ist 4) String mit Sonderzeichen, der der gleiche String wie der 2. String ist 5) String mit Sonderzeichen, der ein echter Substring von dem 2. String ist 6) String mit Sonderzeichen, der kein Substring von dem 2. String ist 7) Leerer String 8) String mit Leerzeichen (vlt. auch Sonderzeichen), der der gleiche String wie der 2. String ist 9) String mit Leerzeichen (vlt. auch Sonderzeichen), der ein echter Substring von dem 2. String ist 10) String mit Leerzeichen (vlt. auch Sonderzeichen), der kein Substring von dem 2. String ist	11) Kein String
2. String	12) String aus Buchstaben und/oder Ziffern, der der gleiche String wie der 1. String ist 13) String aus Buchstaben und/oder Ziffern, der ein echter Substring von dem 1. String ist 14) String aus Buchstaben und/oder Ziffern, der kein Substring von dem 1. String ist 15) String mit Sonderzeichen, der der gleiche String wie der 1. String ist 16) String mit Sonderzeichen, der ein echter Substring von dem 1. String ist 17) String mit Sonderzeichen, der kein Substring von dem 1. String ist 18) Leerer String 19) String mit Leerzeichen (vlt. auch Sonderzeichen), der der gleiche String wie der 1. String ist 20) String mit Leerzeichen (vlt. auch Sonderzeichen), der ein echter Substring von dem 1. String ist 21) String mit Leerzeichen (vlt. auch Sonderzeichen), der kein Substring von dem 1. String ist	22) Kein String

Äquivalenzklassentestschema:

Strings sind hier der Übersichtlichkeit halber von Anführungszeichen umrandet.

Testfall- nummer	Getestete Äquivalenzklassen	Eingabe 1	Eingabe 2	Ergebnis/Kommentar
0.	1, 12	„0d3r“	„0d3r“	Kein Substring
1.	2, 14	„f5jk“	„ghf5jk79“	1. String ist ein Substring des 2.
2.	2, 14	„allo“	„hallo“	1. String ist ein Substring des 2.
3.	2, 14	„hall“	„hallo“	1. String ist ein Substring des 2.
4.	2, 14	„a“	„hallo“	1. String ist ein Substring des 2.
5.	2, 17	„allo“	„%hallo+“	1. String ist ein Substring des 2.
6.	2, 21	„Hallo“	„Hallo Welt!“	1. String ist ein Substring des 2.
7.	3, 13	„ghf5jk79“	„hf5“	2. String ist ein Substring des 1.
8.	3, 14	„ghf5jk79“	„796g“	Kein Substring
9.	3, 17	„ghf5jk79“	„hf5&“	Kein Substring
10.	3, 18	„ghf5jk79“	„“	Kein Substring
11.	3, 21	„ghf5jk79“	„1 0h“	Kein Substring
12.	3, 22	„ghf5jk79“	42	Falsche Eingabe
13.	4, 15	„Hallo!“	„Hallo!“	Kein Substring
14.	5, 17	„Hallo!“	„Hallo!!!“	1. String ist ein Substring des 2.
15.	5, 21	„!Hallo“	„!Hallo Welt!“	1. String ist ein Substring des 2.
16.	6, 13	„Hallo!“	„Hallo“	2. String ist ein Substring des 1.
17.	6, 14	„Hallo!“	„hallo0“	Kein Substring
18.	6, 16	„Hallo,d“	„Hallo,“	2. String ist ein Substring des 1.
19.	7, 17	„“	„1%“	Kein Substring
20.	7, 22	„“	1.56	Falsche Eingabe
21.	8, 19	„Hallo Welt!“	„Hallo Welt!“	Kein Substring
22.	9, 21	„Hallo “	„Hallo Welt!“	1. String ist ein Substring des 2.
23.	10, 20	„Hallo Welt!“	„Hallo “	2. String ist ein Substring des 1.
24.	11, 14	127	„127“	Falsche Eingabe
25.	11, 22	1337	1337	Falsche Eingabe

Das sind 25 Tests. An sich können alle Tests auch für den Funktionstest verwendet werden, denn wenn sie fehlschlagen, dann stimmt irgendetwas an dem Programm nicht.

Aufgabe 3

Äquivalenzklassen für den Modi “Bearbeitung”

Gültige Äquivalenzklassen: Zeit positiv Distanz positiv Kosten numerisch 2 unterschiedliche städte

Ungültige Äquivalenzklassen: negative Zeit/Distanz alphabetische Zeit/Distanz/kosten ein wert nicht angegeben 2 identische städte

Tests für die Äquivalenzklassen des Modi “Bearbeitung”

Äquivalenzklassen für den Modi “Abfrage”

Gültige Äquivalenzklassen: wert bezüglich (zeit/distanz/kosten/agonie) bekannt zwei äquivalente routen möglich kürzeste route führt über alle anderen/eine weitere städte

Ungültige Äquivalenzklassen: Start und zielort identisch Start oder zielort unbekannt es gibt keine verbindung zwischen start und zielort

Tests für die Äquivalenzklassen des Modi “Abfrage”