

LU05c - Äquivalenzklassen-Analyse



Die Äquivalenzklassen-Analyse ist eine Technik zur Definition von Testfällen. Bei der Äquivalenzklassen-Analyse definieren wir für jede Eingabe die Gruppen von Werten, die vom Programm gleich verarbeitet werden (sollten).

Grundprinzip

Eine Äquivalenzklasse ist eine Menge von Eingabewerten, die das System auf ähnliche Weise verarbeitet. Die Annahme ist, dass das Testen eines repräsentativen Werts aus der Klasse ausreicht, um die gesamte Klasse abzudecken. Die Werte innerhalb einer Klasse sind dabei typischerweise entweder alle gültig oder alle ungültig.

Angenommen wir haben ein Eingabefeld für eine Ganzzahl zwischen 0 und 100. Bei der Äquivalenzklassen-Analyse gehen wir davon aus, dass jede Gruppe von Zahlen gleich verarbeitet werden:

- alle gültigen Eingaben
- alle zu kleinen Zahlen
- alle zu grossen Zahlen
- Alle formal falschen Eingaben (Buchstaben, Sonderzeichen, ...).

Anstatt nun jede denkbare Zahl und jede Kombination von Buchstaben zu testen, brauchen wir pro Klasse nur je einen Testfall. Zum Beispiel

- 34 (gültige Zahl)
- -5 (zu kleine Zahl)
- 251 (zu grosse Zahl)
- a3 (formal ungültig)

Äquivalenzklassen definieren

Je nach Art der Eingabe benötigen wir unterschiedliche Äquivalenzklassen.

Zahlen

- 1 Klasse für gültige Eingaben
- 1 Klasse für eine zu kleine Zahl
- 1 Klasse für eine zu grosse Zahl
- 1 Klasse für formale falsche Eingaben

Text

- 1 Klasse für gültige Eingaben
- 1 Klasse für eine zu kurze Eingabe
- 1 Klasse für eine zu lange Eingabe
- 1 Klasse für ungültige Zeichen

Auswahl

Steht dem Benutzer eine beschränkte Auswahl von Werten zur Verfügung, sollte jeder gültige Wert separat getestet werden.

- 1 Klasse pro gültigem Wert
- 1 Klasse für ungültige Werte

Beispiel: Personalien

Eingaben und Validerungsregeln

Eingabe	Datentyp	Regeln
Vorname	Text	3-50 Zeichen, Buchstaben und Umlaute
Geschlecht	Auswahl	M, W oder D
Alter	Ganzzahl	0 - 150
Grösse in Meter	Dezimalzahl	0.2 - 2.99

Äquivalenzklassen

Nr	Eingabe	Klasse
Gültige Klassen		
1	Vorname	Text mit 3-50 Buchstaben und Umlauten
2	Geschlecht	M
3	Geschlecht	W
4	Geschlecht	D
5	Alter	Ganzzahl von 0 bis 150
6	Grösse	Dezimalzahl von 0.2 bis 2.99
Ungültige Klassen		
7	Vorname	Text mit 2 Buchstaben oder weniger
8	Vorname	Text mit mehr als 50 Buchstaben
9	Vorname	Text mit ungültigen Zeichen
10	Geschlecht	alle anderen Werte
11	Alter	Ganzzahl kleiner 0
12	Alter	Ganzzahl grösser 150
13	Alter	Keine Ganzzahl
14	Grösse	Dezimalzahl kleiner 0.2
15	Grösse	Dezimalzahl grösser 2.99

Nr	Eingabe	Klasse
Gültige Klassen		
16	Grösse	Keine Zahl

Testfälle ableiten

Aus den Äquivalenzklassen können wir nun die Testfälle ableiten. Als Eingabewerte wählen wir „typische“ Werte die klar innerhalb der Klasse liegen. Die Grenzen der Klassen werden später in separaten Testfällen (Grenzwertanalyse) getestet.

Klassen mit gültigen Werten

Zunächst definieren wir Testfälle mit ausschliesslich gültigen Klassen. Dabei muss jede gültige Klasse in mindestens einem Testfall vorkommen.

Im oben stehenden Beispiel können wir Testfälle mit den folgende Äquivalenzklassen definieren:

Nr	Klassen
1	1, 2, 5, 6
2	1, 3, 5, 6
3	1, 4, 5, 6

Testfälle mit ungültigen Werten

Wir testen immer nur einen ungültigen Werte auf einmal. Daher muss für jede ungültige Klasse ein eigener Testfall definiert werden.

Beispiel: Personalien

Nr	1	
Klassen	1,2,5,6	
Eingaben	Vorname	Hanna
	Geschlecht	M
	Alter	35
	Grösse	1.35
Nr	2	
Klassen	1,3,5,6	
Eingaben	Vorname	Peter
	Geschlecht	W
	Alter	56
	Grösse	0.76
Nr	3	
Klassen	1,4,5,6	

Eingaben	Vorname	Diego
	Geschlecht	D
	Alter	15
	Grösse	2.35
Nr		4
Klassen		7
Eingaben	Vorname	H
	Geschlecht	M
	Alter	14
	Grösse	1.86
Nr		5
Klassen		8
Eingaben	Vorname	Anna Maria Davide Juan Marco Julius William Bruce akodfaisdfhaosdihgaoisdgh
	Geschlecht	M
	Alter	6
	Grösse	1.33
Nr		9
Klassen		9
Eingaben	Vorname	Mar\$el
	Geschlecht	D
	Alter	19
	Grösse	0.74

M450-LU05



Marcel Suter

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/modul/m450/learningunits/lu05/aequivalenzklassen>

Last update: **2024/12/03 08:48**

