LU04e - Verknüpfte Bedingungen

Einige Selektionen lassen sich nicht mit einer einzigen Bedingung ausdrücken.

Zum Beispiel: Das Alter einer Person muss zwischen 18 und 27 sein.

Diese Bedingung kann in Form von zwei Selektionen dargestellt werden:

Programmablaufplan	Pseudocode	Java

	`falls (alter >= 18)`
	`  falls (alter <= 27)`
	`  Befehle`
	`  ende`
	`ende`
	
	`Befehle`
		`if (alter >= 18) {`
		`  if (alter <= 27) {`
		`  // Befehle`
		`  }`
		`}`
		
		`// Befehle`

Da die zweite Selektion abhängig von der ersten Selektion ist, spricht man von **verschachtelten** Selektionen. Eine andere Möglichkeit um die Bedingung auszudrücken, ist eine **verknüpfte** Bedingung.

Programmablaufplan	Pseudocode	Java

	`falls ( (alter >= 18) UND (alter <= 27) )`
	`  Befehle`
	`ende`
	
	`Befehle`
		`if ( (alter >= 18) && (alter <= 27) )`
		`{`
		`  // Befehle`
		`}`
		
		`// Befehle`

Die beiden Teilbedingungen werden mittels `UND` verknüpft. Nur wenn beide Teilbedingungen erfüllt sind, ist die Bedingung als Ganzes erfüllt. Sie müssen beide Teilbedingungen vollständig codieren, also `Variable Operator Konstante`. Es gibt **keine** abgekürzte Schreibweise im Sinne von '' Falls (Alter > = 18 UND < = 27) ''

Beim Programmieren empfiehlt es sich, die Teilbedingungen jeweils in Klammern zu setzen. Dadurch ist es einfacher den Überblick zu behalten.

Logische Operatoren

Logische Operationen benötigen wir für Bedingungen bei Selektionen und Iterationen. Sobald eine Bedingung aus zwei oder mehr Teilbedingungen besteht, müssen wir diese mit logischen Operatoren verknüpfen.

Mengenlehre

Siehe auch
https://de.wikibooks.org/wiki/Mathe_f%C3%BCr_Nicht-Freaks:_Verkn%C3%BCpfungen_zwischen_Mengen

Angelehnt an die Mengenlehre stellen wir das Resultat einer Verknüpfung grafisch dar. Jede Teilbedingung wird mit einem Oval dargestellt.

- Innerhalb des Ovals liegen alle Werte, für die die Bedingung erfüllt ist.
- Ausserhalb des Ovals sind Werte, für die die Bedingung nicht erfüllt ist.

Wenn wir mehrere Bedingungen kombinieren, können wir die logischen Operationen grafisch darstellen.

Übersicht der logischen Operatoren

Für die folgenden Beispiele verwenden wir eine Gruppe von Personen:

Name	Alter	Geschlecht
Anna	17	W
Beat	17	M
Carla	18	W
David	18	M
Emma	19	W
Frank	19	M

Ausserdem verwenden wir zwei Teilbedingungen:

- (alter > 18)
- (geschlecht = 'W')

Betrachten wir zunächst die Teilbedingungen einzeln, so ergibt sich folgendes Bild:

alter > 18	geschlecht = 'W'

Legen wir beide Abbildungen übereinander, so haben wir folgendes Bild:

AND

```
Falls ( (alter > 18) AND (geschlecht = 'W') )   // Pseudocode
if ( (alter > 18) && (geschlecht == 'W') ) {    // Java
```

Die Bedingung ist nur erfüllt, wenn beide Teilbedingungen erfüllt sind.

OR

```
Falls ( (alter > 18) OR (geschlecht = 'W') )
if ( (alter > 18) || (geschlecht == 'W') ) {    // Java
```

Die Bedingung ist erfüllt, wenn eine oder beide Teilbedingungen erfüllt sind.



Auswerten von verknüpften Bedingungen

Der Computer muss jede Bedingung auf einen bool'schen [1] Wert reduzieren.

Er kann streng genommen nur Bedingungen in der Form `Falls TRUE` bzw. `Falls FALSE` überhaupt verarbeiten. Um eine verknüpfte Bedingung auszuwerten muss der Computer daher diese Bedingung schrittweise vereinfachen:

1. Die Teilbedingung in den innersten Klammern werden ausgewertet. Dies ergibt für jede Teilbedingung den Wert TRUE oder FALSE.
2. Das Ergebnis der Auswertung in Schritt 1 werden verknüpft und ebenfalls ausgewertet. Dadurch entsteht die gewünschte, verinfachte Form der Bedingung.

Die folgende Grafik zeigt dieses Vorgehen beim Vereinfachen einer Bedingung. Das Beispiel geht davon aus, dass das `Alter = 25` ist.



Verknüpfungsoperatoren

Wir betrachten vorerst zwei logische Operatoren zur Verknüpfung von Teilbedingungen:

UND	Beide Teilbedingungen müssen `true` sein
ODER	Mindestens eine Teilbedingung muss `true` sein

Diese Tabelle zeigt, welchen Einfluss der Verknüpfungsoperator auf die Bedingung hat.

Variablen		**Teilbedingungen**		**Verknüpfung**	
zahl1=	**zahl2=**	**zahl1 >= 18**	**zahl2 < 50**	**(zahl1 >= 18) && (zahl2 < 50)**	**(zahl1 >= 18) || (zahl2 < 50)**
32	40	true	true	true	true
32	76	true	false	false	true
4	40	false	true	false	true
4	76	false	false	false	false

 © Marcel Suter

1)

Ein boolscher Wert kennt nur die beiden Zustände TRUE oder FALSE

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

Permanent link:
https://wiki.bzz.ch/modul/archiv/m319/learningunits/lu04/lu04e-verknuepftebedingungen

Last update: **2025/06/23 07:45**