## **LU01.L07 - Rekursiver GGT und Trace Table**

## **Algorithmus:**

```
def ggt(a, b):
    if b == 0:
        return a
    else:
        return ggt(b, a % b)

# Beispielwerte
a = 48
b = 18
ergebnis = ggt(a, b)
print(f'GGT von {a} und {b} ist: {ergebnis}')
```

## **Trace Table:**

Schritt	а	b	a % b	Rekursiver Aufruf (ggt(b, a % b))	Rückgabewert
1	48	18	12	ggt(18, 12)	-
2	18	12	6	ggt(12, 6)	-
3	12	6	0	ggt(6, 0)	-
4	6	0	-	-	6

## **Erklärung des Trace Tables:**

- **Schritt 1**: Der Algorithmus beginnt mit den Werten a = 48 und b = 18. Da b nicht 0 ist, erfolgt ein rekursiver Aufruf mit den neuen Werten ggt (18, 12).
- **Schritt 2**: Der Algorithmus wird nun mit a = 18 und b = 12 ausgeführt. Der Rest von 18 % 12 ist 6, daher wird ggt (12, 6) rekursiv aufgerufen.
- **Schritt 3**: Jetzt sind die Werte a = 12 und b = 6. Der Rest von 12 % 6 ist 0, und der rekursive Aufruf erfolgt mit ggt (6, 0).
- **Schritt 4**: In diesem Schritt ist b = 0, daher gibt der Algorithmus den Wert von a zurück, der 6 ist. Dies beendet die Rekursion und der GGT ist 6.



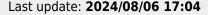
NC SA © Kevin Maurizi

From:

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

Permanent link:

https://wiki.bzz.ch/modul/m323/learningunits/lu01/loesungen/tracetable2





Last

update:
2024/08/06
2024/08/06
17:04
modul:m323:learningunits:lu01:loesungen:tracetable2 https://wiki.bzz.ch/modul/m323/learningunits/lu01/loesungen/tracetable2

https://wiki.bzz.ch/ Printed on 2025/11/13 09:45