



Vorlesung „Grundlagen der Informatik und Programmierung 1“

MODULARISIERUNG

Rekursion

Prof. Dr. Tom Vierjahn

Visual Computing (<https://vc.w-hs.de>)

Fachbereich Wirtschaft und Informationstechnik – Campus Bocholt



Wintersemester 2020/21



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

$n! =$

n	Berechnung	$n!$
0		1
1		1
2		2
3		6
4		24
5		120
6		720
7		5040
8		40 320
9		362 880
10		3 628 800

$$n! =$$

$$=$$

Definition: Rekursion

Bei der **Rekursion** ruft eine Funktion sich selbst unmittelbar oder mittelbar (über eine andere Funktion) auf. Rekursion führt ein Problem auf ein oder mehrere strukturell gleiche, aber „kleinere“ Probleme zurück, bis am Ende ein leicht zu lösendes Problem steht.

Was passiert?

Fakultät von 4:

```
1 #include <stdio.h>
2
3 int factorial(int n) {
4     if (n <= 1) {
5         return 1;
6     }
7     return n * factorial(n - 1);
8 }
9
10 int main() {
11     int f = factorial(4);
12     printf("%d\n", f);
13     return 0;
14 }
```


Ablauf:

Fakultät von 4:

```
1 #include <stdio.h>
2
3 int factorial(int n) {
4     if (n <= 1) {
5         return 1;
6     }
7     return n * factorial(n - 1);
8 }
9
10 int main() {
11     int f = factorial(4);
12     printf("%d\n", f);
13     return 0;
14 }
```

- ▶ Fakultätsfunktion
- ▶ Ablauf der rekursiven Implementierung
- ▶ Stack

Prof. Dr. Tom Vierjahn

▶  tom.vierjahn@w-hs.de

Visual Computing

▶  <https://vc.w-hs.de>

▶  VisualComputingWH

▶  Visual Computing WH

▶  @VisComputingWH

Westfälische Hochschule

Fachbereich Wirtschaft und Informationstechnik

Campus Bocholt



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)