

Rekursion Einleitung

Einleitung:

Unter Rekursion in der Softwareentwicklung versteht man, dass eine Funktion sich direkt oder indirekt selbst wieder aufruft.

```
private void doThis(int param) {  
    ..  
    doThis(var);    //Rekursiver Aufruf  
    ..  
}
```



Aufgabe: Film


- Schauen Sie sich den Film über Rekursion auf YouTube an.

<https://www.youtube.com/watch?v=weTpjhDnLnc>

Aufgabe: Wikipedia

Lesen Sie was Wikipedia zu Rekursionen schreibt:

Rekursion

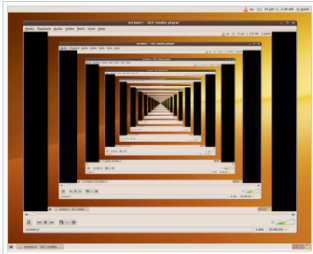
 Dieser Artikel erläutert unter anderem die Technik der rekursiven Definition in der Mathematik; zum Begriff *rekursive Menge* siehe jedoch das Stichwort [Entscheidbar](#).

Als **Rekursion** (*lateinisch* *recurere* ‚zurücklaufen‘) bezeichnet man den abstrakten Vorgang, dass Regeln auf ein Produkt, das sie selbst erzeugt haben, von neuem angewandt werden. Hierdurch entstehen potenziell unendliche Schleifen. Regeln bzw. Regelsysteme heißen *rekursiv*, wenn sie die Eigenschaft haben, Rekursion im Prinzip zuzulassen.

Rekursion ist ein zentraler Begriff in [Mathematik](#) und [Informatik](#) und hat vielfältige Anwendungen darüber hinaus; diese reichen bis in die Kunst, wo das Phänomen auch als *mise en abyme* bezeichnet worden ist.

Rekursion ist auch eine Problemlösungsstrategie. Komplexe Sachverhalte können oft mit rekursiv formulierten Regeln sehr elegant erfasst werden. Das Grundprinzip ist dabei dann das Zurückführen einer allgemeinen Aufgabe auf eine einfachere Aufgabe derselben Klasse. Beispielsweise ist die [rekursive Programmierung](#) Bestandteil vieler Programmiersprachen. [Prozeduren](#) oder [Funktionen](#) können sich dabei selbst aufrufen. Rekursion und [Iteration](#) sind im Wesentlichen gleichmächtige Sprachmittel.

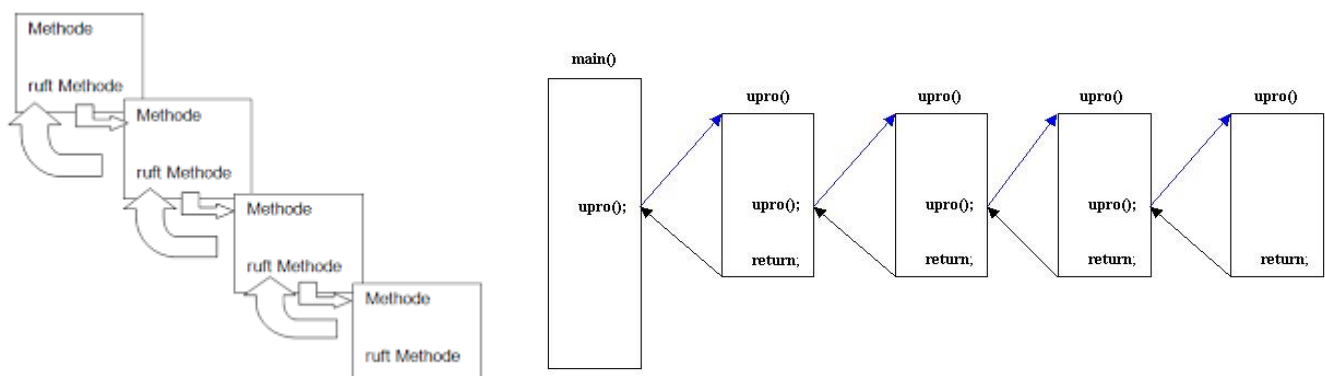
In Mathematik und Informatik erscheint Rekursion spezieller in der Form, dass eine [Funktion](#) in ihrer [Definition](#) selbst nochmals aufgerufen wird (*rekursive Definition*). Wenn man mehrere Funktionen durch wechselseitige Verwendung voneinander definiert, spricht man von *wechselseitiger Rekursion*. Nicht jede rekursive Definition ist eine Definition im eigentlichen Sinn, denn die zu definierende Funktion braucht nicht [wohldefiniert](#) zu sein. Jeder Aufruf der rekursiven Funktion muss sich durch Entfalten der rekursiven Definition in endlich vielen Schritten auflösen lassen. Ist dies nicht erfüllt, so spricht man von einem [infiniten Regress](#) (in der [Informatik](#) auch als [Endlosschleife](#) bezeichnet).



Ein Beispiel von Rekursion: [Rückkopplung](#) im [VLC media player](#) bei Anzeigen des eigenen Bildschirms.

Aufgabe: Erstes Verstehen

- Verstehen Sie damit diese Skizze?
Erklären Sie mit Ihrem Lernpartner was hier dargestellt ist.



Sind Rekursionen effizient?

Effizienz [[Bearbeiten](#) | [Quelltext bearbeiten](#)]

Rekursive Programme haben in der Regel keine gute [Performance](#). Durch die wiederholten Funktionsaufrufe (Inkarnationen) wird immer wieder derselbe Methodeneintrittscode bearbeitet und bei jeder Inkarnation der [Kontext](#) gesichert, was zu zusätzlichem Programmcode und höherem [Arbeitsspeicherverbrauch](#) führt. Alle rekursiven [Algorithmen](#) lassen sich jedoch auch durch [iterative Programmierung](#) implementieren (und umgekehrt). Man hätte die Fakultät auch so implementieren können: