# L2\_1 Algorithmik

**1 Begriff Algorithmus**

Algorithmen begegnen uns ständig im Alltag. Beim Onlinebanking, bei der Nutzung eines Routenplaners, beim Online-Shopping, bei der Suche im Internet und vielem mehr vertrauen wir auf Algorithmen. Sie und die ausführenden Computer beeinflussen unser Leben in starkem Maße.  
Algorithmen sind jedoch nicht nur in der Informatik vorzufinden. Sie werden auch in normaler Sprache formuliert bzw. niedergeschrieben. Als Beispiele hierfür können Bedienungsanleitungen für technische Geräte, Spielregeln für ein Gesellschaftsspiel oder Anwendungshinweise für Arzneimittel genannt werden.

Letztendlich handelt es sich bei einem Algorithmus immer um Handlungsanweisungen, die eindeutig den Weg zur Lösung eines gegebenen Problems beschreiben.

**2 Eigenschaften von Algorithmen**

Die Handlungsanweisungen eines Algorithmus sind allgemeingültig und lösen eine Vielzahl von Problemen der gleichen Art.   
Ein Algorithmus zur Ermittlung der Primzahlen des Wertebereichs 3 bis 100 muss also auch für alle anderen Wertebereiche gelten.

Die Handlungsanweisungen eines Algorithmus dürfen sich nicht widersprechen. Jede Anweisung muss immer eindeutig formuliert sein.   
Die Anweisungen eines Kochrezepts 'Eine Zwiebel in Würfel schneiden' – 'Butter und Zwiebelringe in die Pfanne geben' erfüllen diese Eigenschaft nicht, da zum einen nicht eindeutig festgelegt wird, wie warm die Pfanne sein soll und zum anderen die Zwiebel einerseits in Würfel geschnitten und andererseits in Ringen in die Pfanne gegeben werden soll.

Jede einzelne Handlungsanweisung eines Algorithmus muss für den Ausführenden des Algorithmus (Mensch oder Computer) verständlich und ausführbar sein.  
Die Anweisungen eines Kochrezepts 'Wasser auf 120° Celsius erhitzen' widerspricht dieser Eigenschaft, da sie nicht ausführbar ist.

Ein Algorithmus muss aus einer endlichen Anzahl von Handlungsanweisungen bestehen. Die Anweisungen müssen also aus einer begrenzten Anzahl von Zeichen bestehen. Diese Eigenschaft wird auch als Finitheit bezeichnet.  
Ein Algorithmus, der diese Eigenschaft nicht erfüllt ist beispielsweise:  
 1. Starte mit der Zahl 1  
 2. Addiere zu dieser Zahl die Zahl 1  
 3. Addiere zu dieser Zahl die Zahl 1  
 4. Addiere zu dieser Zahl die Zahl 1  
 5. …

Die Handlungsanweisungen eines Algorithmus können nach endlich vielen Schritten zu einem Ende kommen und ein Ergebnis liefern. Man spricht in diesem Fall von der Terminiertheit eines Algorithmus.

Der folgende Algorithmus zur Ermittlung der Zweierpotenzen zeigt Handlungsanweisungen, die nicht zu einem Ende führen. Dieser Algorithmus wäre somit nicht terminiert.

|  |  |
| --- | --- |
| Deklaration und Initialisierung: zahl = 2 als Ganzzahl | |
| Wiederhole solange zahl > 0 | |
|  | zahl = zahl \* 2 |
|  | Ausgabe: zahl |

Die Handlungsanweisungen eines Algorithmus müssen bei gleichen Voraussetzungen immer das gleiche Ergebnis liefern. Diese Eigenschaft wird auch als Determiniertheit bezeichnet. Ein Algorithmus ist somit determiniert, wenn die wiederholte Anwendung des Algorithmus mit denselben Eingabe­daten immer wieder dieselben Ausgabedaten liefert. Ein Algorithmus, der die Anweisung "*Bearbeiten der Hausaufgaben*" enthält, erfüllt diese Eigenschaft nicht.

Bei der Ausführung der Handlungsanweisungen eines Algorithmus darf es zu jedem Zeitpunkt der Ausführung höchstens eine Möglichkeit der Fortsetzung geben. Der Folgeschritt einer Anweisung muss eindeutig definiert sein. Diese Eigenschaft wird als Determinismus bezeichnet.  
Wenn an einer oder mehreren Stellen des Algorithmus alternative Handlungsanweisungen erfolgen, ohne die Vorgabe, welche zu wählen ist, spricht man von einem nichtdeterministischen Algorithmus.

Die Anweisungen eines Kochrezepts, bei dem beispielsweise mehrere Varianten des Würzens beschrieben sind, widersprechen dieser Eigenschaft und stellen einen nichtdeterministischen Algorithmus dar.