Algorithmisches Problemlösen

Eine Problemspezifikation ist eine vollständige und eindeutige Beschreibung des Ausgangszustands („Zustand vorher“) und Zielzustandes („Zustand nachher“).

Eine Algorithmus ist eine endliche Folge eindeutig ausführbarer Anweisungen zur Lösung eines Problems.
Ziel ist es, einen komplizierten Ablauf / Vorgang so zu beschreiben, dass er von einem „Prozessor“ (Mensch oder Maschine, der bzw. die für die Ausführung zuständig ist) ausgeführt werden wird.

Anforderungen an Algorithmen:
- Endlichkeit: Die Anweisungsfolge ist durch einen endlichen Text beschrieben.
- Ausführbarkeit: Die Anweisungen sind für den „Prozessor“ (Mensch oder Maschine) verständlich formuliert und ausführbar.
- Eindeutigkeit: An jeder Stelle ist der Ablauf der Anweisungen eindeutig festgelegt.
- Allgemeinheit: Die Anweisungen besitzen Gültigkeit für die Lösung einer ganzen Problemklasse, nicht nur für ein Einzelproblem.
(s. Gasper, Leiß, Spengler, Stimm: Technische und theoretische Informatik)

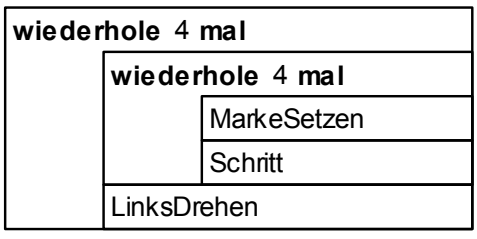
Ein Algorithmus ist korrekt bzgl. einer Spezifikation, wenn er jeden möglichen Ausgangszustand tatsächlich in den festgelegten Zielzustand überführt.

Die Sprache, in der Algorithmen für einen „Prozessor“ formuliert werden, ist (normalerweise) genau festgelegt. Den Aufbau der in dieser Sprache korrekt formulierten „Programme“ wird durch sog. Syntaxregeln beschrieben. Verstößt man gegen eine dieser Syntaxregeln, so kommt es zu einem Syntaxfehler (Fehlermeldung z. B.: unbekannte Anweisung).

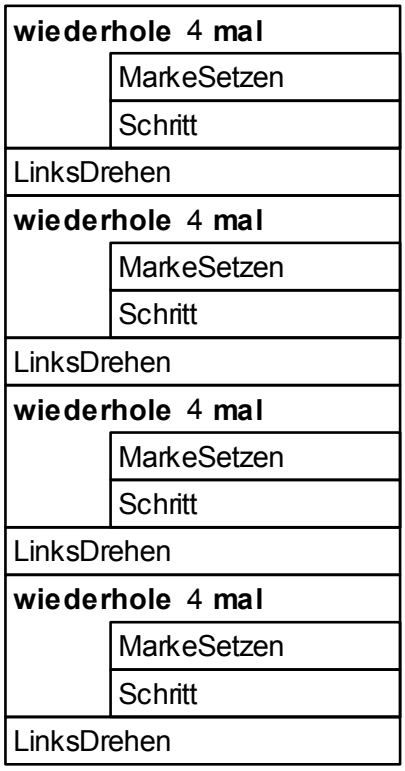
Wenn das Programm nicht korrekt ist, dann liegt ein logischer Fehler vor. Beachte, dass man in der Regel in einem solchen Fall keine Fehlermeldung erhält. Logische Fehler kann man durch Austesten des Programms feststellen.

Wiederholungen

Soll eine Sequenz (von Anweisungen) mehrfach ausgeführt werden, wobei die Anzahl der Wiederholungen von Anfang an feststeht, so benutzt man zur Beschreibung eine Wiederholung mit fester Anzahl.

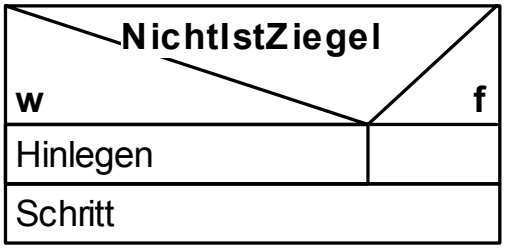


Wiederholungsanweisungen kann man auch ineinander schachteln.

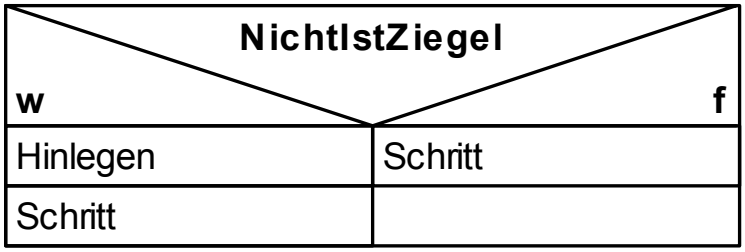


Fallunterscheidungen

einseitige Fallunterscheidung

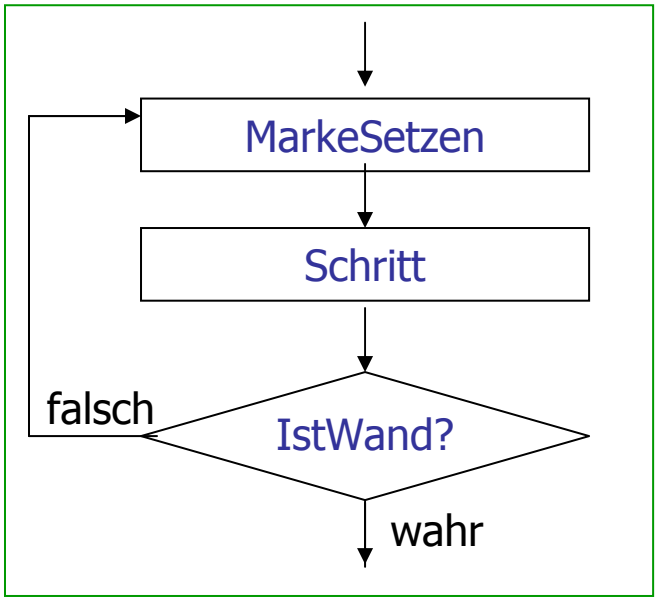


zweiseitige Fallunterscheidung

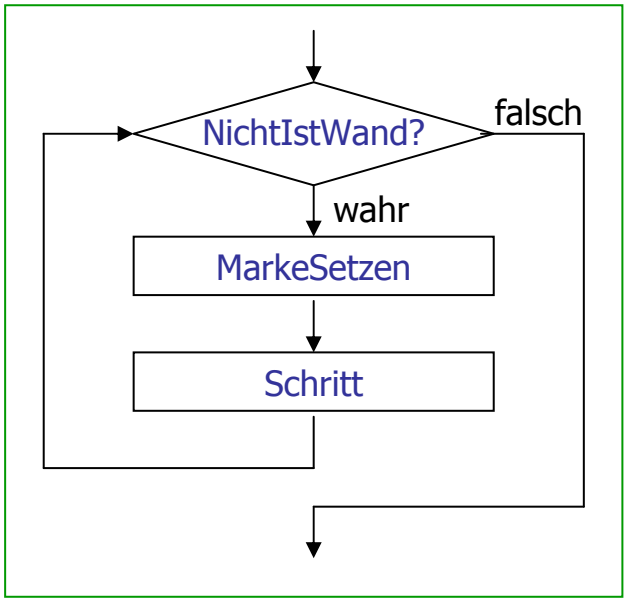


Wiederholungen mit Abbruchbedingung

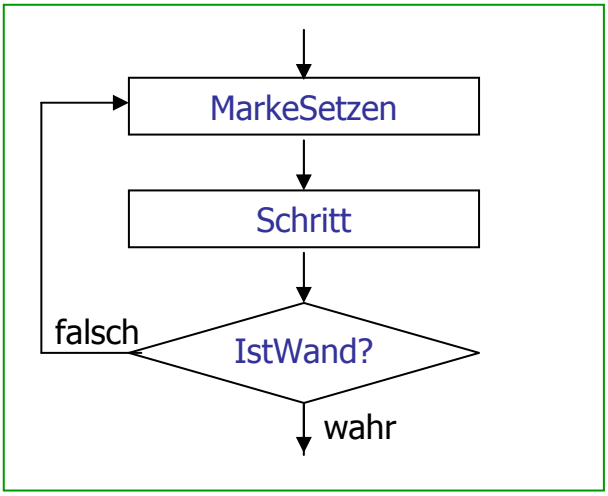
Austrittsbedingung

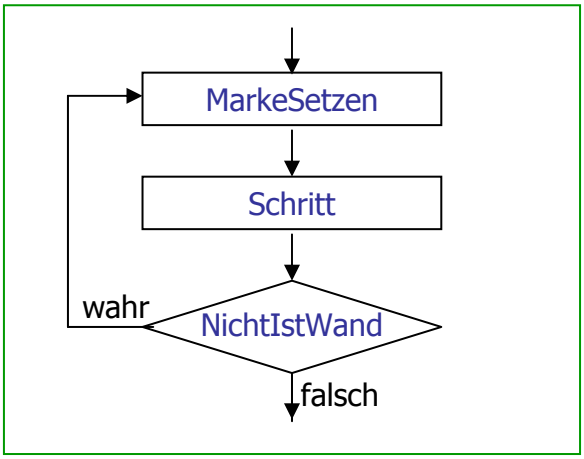


Eintrittsbedingung



Bei einer Wiederholung mit Austrittsbedingung wird die Sequenz (von Anweisungen) mindestens einmal ausgeführt, bei einer Wiederholung mit Eintrittsbedingung kann die Sequenz (von Anweisungen) auch überhaupt nicht ausgeführt werden.

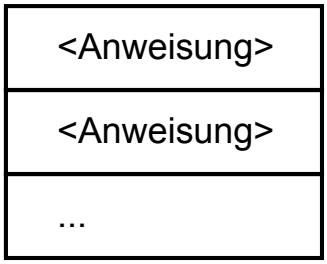


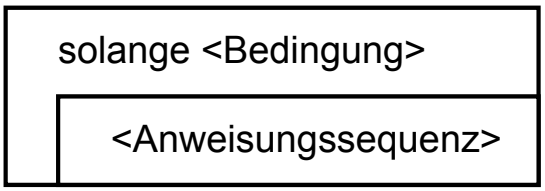


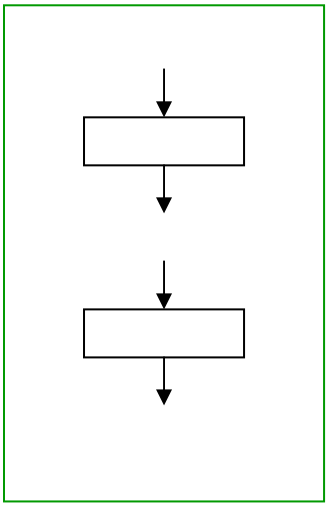
Bei einer Wiederholung mit Austrittsbedingung kann überprüft werden, ob die Austrittsbedingung erfüllt bzw. nicht erfüllt ist. (Analog gilt das für Eintrittsbedingung)

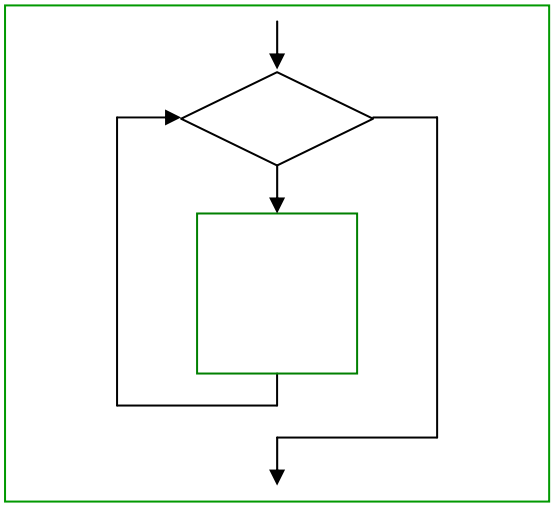
Bei einer Wiederholung mit Abbruchbedingung kann es vorkommen, dass es zu keinem regulären Abbruch der Wiederholungen kommt. Das Programm (bzw. der Algorithmus) gerät dann in eine Endlosschleife.

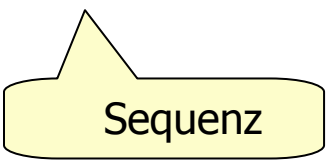
Kontrollstrukturen

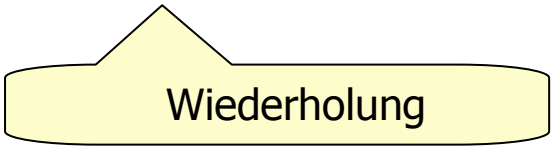


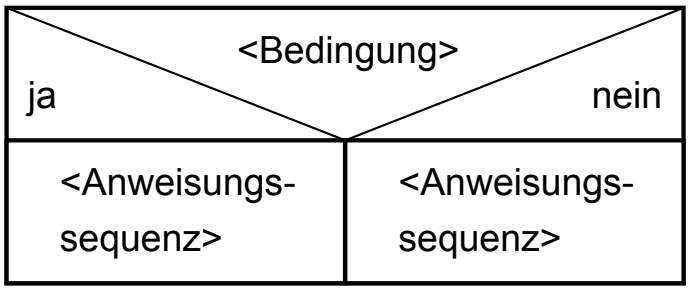


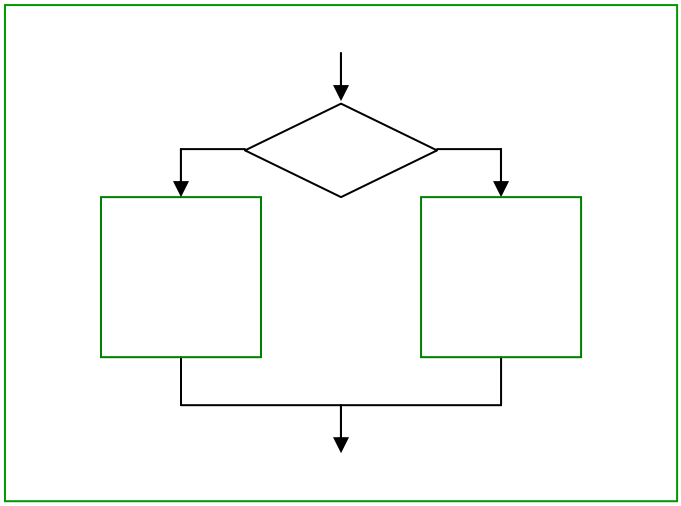












Fallunterscheidung

Kontrollstrukturen dienen dazu, den Ablauf der Verarbeitung festzulegen.