

Algorithmen

Mittwoch, 8. Februar 2023 05:42

Definition Algorithmus	<p>-Ein Algorithmus ist ein Verfahren mit einer präzisen endlichen Beschreibung unter Verwendung effektiver Verarbeitungsschritte</p> <ul style="list-style-type: none">•Verfahren: - besteht aus elementaren Verarbeitungsschritten<ul style="list-style-type: none">- die einer kontextbedingten Interpretation bedürfen- muss daher am ausführenden Objekt und dessen Möglichkeiten orientiert sein•präzise: - in einer genau festgelegten Sprache abgefasst, die für das ausführende Objekt verständlich ist<ul style="list-style-type: none">- es muss für jeden Verarbeitungsschritt klar sein, welche Tätigkeit auszuführen ist und wann sie durchgeführt werden muss•endlich: -> statisch: die Beschreibung muss endlich sein endlich viele Verarbeitungsschritte<ul style="list-style-type: none">-> dynamisch: die Ausführung muss endlich sein- der Algorithmus terminiert- man spricht aber auch von nichtterminierenden „Algorithmen“•Verarbeitungsschritte: - die einzelnen Verarbeitungsschritte müssen tatsächlich ausgeführt werden können <p>=>Ein Algorithmus ist ein Verfahren mit einer präzisen (d.h. in einer eindeutigen Sprache abgefassten) endlichen Beschreibung unter Verwendung effektiver (im Sinne von tatsächlich ausführbarer) Verarbeitungsschritte.</p> <p>->Löst typischerweise nicht genau eine Aufgabe, sondern eine Klasse von Aufgaben</p> <ul style="list-style-type: none">-Addiere 3 und 4 vs. Addiere zwei natürliche Zahlen-Welche Aufgabe bearbeitet werden soll, wird durch Parameter bestimmt.-Parameter dienen als Eingabe-Algorithmus liefert für geeignete Eingabe in der Regel ein Ergebnis. <ul style="list-style-type: none">•Funktionale Sicht: - Was soll der Algorithmus machen?•Operationale Sicht: - Wie wird die Aufgabe bewältigt?<ul style="list-style-type: none">- Dazu gehören: -> Elementare Verarbeitungsschritte-> Beschreibung der Auswahl der einzelnen auszuführenden Schritte (Kontrollfluss)-> Daten und Parameter, die bearbeitet werden
Algorithmische Sprachelemente	<p>SortiereDurchEinfuegen(A)</p> <pre>1 for i:=2 to n do (betrachte die Bücher an der Stelle i=2 bis n) j := Ai /* Boolesche Ausdruck 2 while j >= 2 and A[j-1] > A[j] do 3 (solange die Bücher A[j] und A[j-1] in der falschen Reihenfolge stehen...) 4 vertausche die Einträge A[j] und A[j#1] 5 (...vertausche Bücher und...) 6 j := j-arithmetischer Ausdruck 7 (...gehe zum Bücherpaar eins weiter links) endwhile endfor</pre> <p>Brauchen: * Verarbeitungsschritt → Variablen zur Zuweisung → Konstanten wie n → Konstrukte für Kontrollfluss (Schleifen Δ)</p> <ul style="list-style-type: none">•Datenspeicher und Zuweisungen: -Variablen und Zuweisungen zum Speichern von berechneten (Zwischen-)Ergebnissen<ul style="list-style-type: none">-Konstanten zum Bezeichnen fester Werte-Wichtig: sinnvolle Bezeichnerwahl!•Ausdrücke: -Boolesche Ausdrücke<ul style="list-style-type: none">-Arithmetische Ausdrücke•Sequenzierungsanweisungen: -Blöcke zum Gruppieren von Daten und Anweisungen<ul style="list-style-type: none">-Funktionsaufrufe zum mehrmaligen Wiederverwenden einmal definierter Algorithmen-Bedingte Anweisungen zum Verzweigen im Programmfluss-Iterationsanweisungen zur Realisierung von Schleif•Hilfskonstrukte: -Kommentare
Sprünge VS. Strukturierte Verzweigungen	<p>Verzweigung mit Sprüngen:</p> <pre>IF Bedingung THEN GOTO marke /* Anweisungen, die ausgeführt werden, falls Bedingung nicht erfüllt GOTO marke2 marke: /* Anweisungen, falls Bedingung erfüllt marke2: /* Ende der Verzweigung</pre> <p>Strukturierte Verzweigung :</p> <pre>IF Bedingung THEN /* Anweisungen, wenn Bedingung wahr ELSE /* Anweisungen, wenn Bedingung falsch ENDIF</pre> <p>Schleife mit Sprüngen:</p> <pre>loop: IF NOT(Bedingung) THEN GOTO ende /* Anweisungen, solange Bedingung wahr GOTO loop ende:/* Weiter im Programmfluss...</pre> <p>Strukturierte Schleifen</p> <pre>WHILE Bedingung DO /* Anweisungen, solange Bedingung wahr END WHILE /* Weiter im Programmfluss...</pre>
Modularisierung	<p>= Zerlegung in Teilprobleme</p> <ul style="list-style-type: none">•Einzelteile idealerweise wiederverwendbar<ul style="list-style-type: none">-> an mehreren Stellen innerhalb eines Programms-> in anderen Programmen

