

# Programmierlogik

## Algorithmus

Nach dem arabischen Mathematiker Al Chwarismi (ca. 820 n. Chr.) gebildetes Wort für Rechenanweisung.

Der Algorithmus nennt die Folge von Schritten, die benötigt werden, um Aufgaben über logische oder mathematische Probleme zu lösen.

Ein Algorithmus ist also ein allgemeines und eindeutiges Verfahren zur Lösung gleichartiger Probleme. Er besteht aus elementaren Anweisungen, die nach einer endlichen Zahl von Schritten das gewünschte Ergebnis ermitteln.

Drei wichtige Grundgedanken:

- die Determiniertheit: eine Algorithmus muss die Durchführung einer Aufgabenlösung auf genau eine Weise gestatten
- die Allgemeinheit: ein Algorithmus muss die Lösung einer Klasse von Aufgaben ermöglichen
- die Endlichkeit: ein Algorithmus muss sich in endlich vielen Schritten zu Ende führen lassen.

Ein Algorithmus ist ein System von Grundoperationen

Eine endliche Folge von eindeutig bestimmten Anweisungen zur Lösung von Aufgaben einer Aufgabenklasse. Für die Folge der Grundoperationen sind zwei Fälle möglich:

- die Reihenfolge ist ein für allemal festgelegt
- die Reihenfolge hängt z. T. von den Ergebnissen vorangegangener Operationen ab

Algorithmen sollen so formuliert sein, dass sie

- möglichst allgemein sind
- inhaltlich eindeutig sind (sprachliche Einschränkung)
- minimale Vorkenntnisse erfordern (kleinste Arbeitsschritte)
- leicht zu überprüfen sind (durch normalisierte Darstellung)

## Darstellungsmittel für Algorithmen

- Sie müssen die Darstellung von kleinen, eindeutigen Arbeitsschritten ermöglichen
- Sie müssen spracharm sein.
- Sie müssen leicht erkennbar sein.
- Sie müssen übersichtlich sein.
- Sie müssen als Kommunikationsgrundlage zwischen Entwickler des Lösungsweges und Anwender geeignet sein.

## Darstellungstechniken

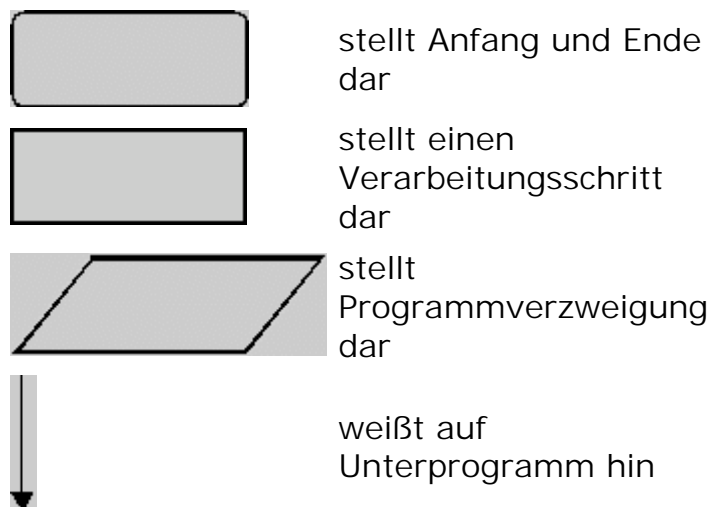
- Programmablaufplan
- Struktogramme (Nassi-Shneiderman Diagramm)

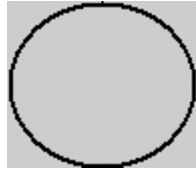
## Programmablaufplan (PAP)

Hilfsmittel zur grafischen Zerlegung eines Problems in Arbeitsschritte, die der Computerlogik entsprechen. Er zeigt in grafischer Form alle Operationen, die das Programm durchführen soll, sowie alle Programmverzweigungen.

Der PAP mag dem Laien als die verständlichste Darstellung eines logischen Ablaufs erscheinen. Dies trifft jedoch nur auf relativ kurze Abläufe zu. Je umfangreicher und stärker vernetzt die Abläufe dargestellt werden, desto undurchsichtiger wird ein Ablaufplan und um so schwieriger wird es, nachträglich Änderungen vorzunehmen.

Symbole des PAP:





stellt die Reihenfolge  
(Verbindung) der  
einzelnen Schritte dar

stellt  
Ablaufunterbrechung  
wegen z. B.  
Platzmangel dar

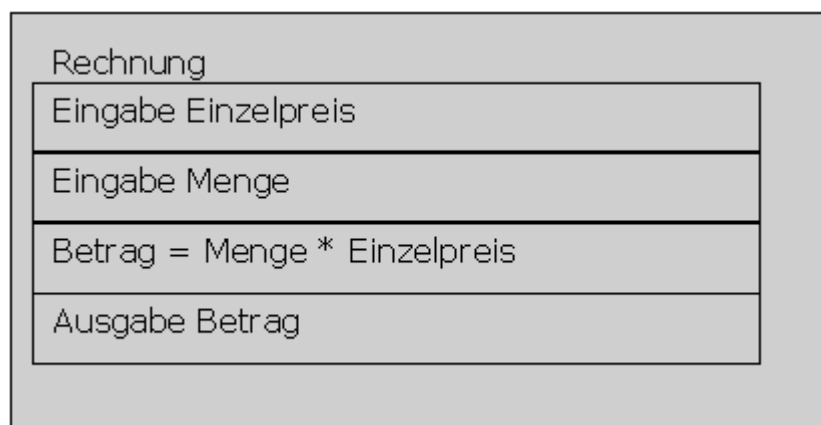
## Struktogramme

Diese Darstellungsform, entwickelt von J. Nassi und B. Shneiderman, befolgt die wesentlichen Grundsätze der strukturierten Programmierung. Daher auch der Name Struktogramm oder auch Nassi-Shneiderman-Diagramm.

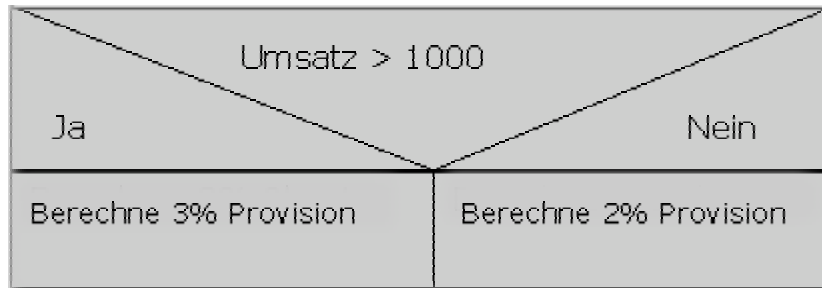
Es gibt keine Ablaflinien in Form von Verbindungslinien für den Programmablauf. Bei der Entwicklung kann das Problem erst als ein einziger Strukturblock dargestellt werden, der dann schrittweise in immer detailliertere weitere Strukturblöcke unterteilt wird.

Darstellungsform ist ein Rechteck. Es kann jede beliebige Operation beschrieben werden.

Werden mehrere Verarbeitungsschritte innerhalb eines Programms öfters benötigt, können sie unter anderem Namen zu einem Block zusammengefasst und jederzeit von anderen Stellen des Struktogramms aufgerufen werden.



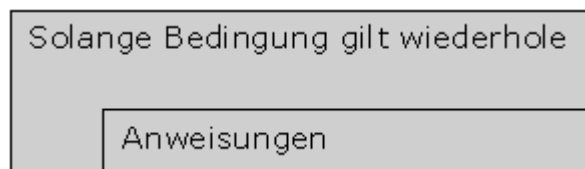
Um eine Verzweigung darzustellen, wird in dem mittleren oberen Bereich die zu prüfende Bedingung geschrieben. Der linke Bereich stellt den Ja-Zweig dar, der rechte den Nein - Zweig. Unter diesen Zweigen kann wieder jeder beliebige Strukturblock erscheinen.



## Strukturbausteine für Schleifen (Wiederholungen)

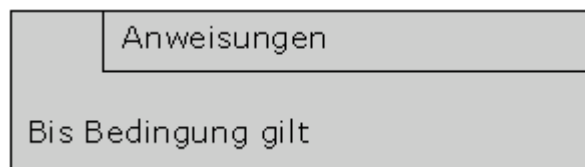
### kopfgesteuert Schleife:

Die zu wiederholenden Anweisungen werden ausgeführt, wenn die Bedingung im Schleifenkopf wahr ist. Diese Bedingung wird bereits vor dem ersten Eintritt in den Schleifenrumpf geprüft, so dass dieser möglicherweise nie durchlaufen wird. Ist die Bedingung falsch, wird mit der auf die Schleife folgenden Anweisung fortgefahren. Die Schleifenbedingung heißt hier Eintrittsbedingung.



### fußgesteuerte Schleife:

Der Schleifenrumpf einer fußgesteuerten Schleife wird auf jeden Fall einmal durchlaufen. Er wird im folgenden immer dann durchlaufen, wenn die Bedingung im Schleifenfuß wahr ist. Ist die Bedingung falsch, wird die auf die Schleife folgende Anweisung durchgeführt. Die Schleifenbedingung heißt hier Abbruchbedingung.



## Daten und Datentypen

Als Daten bezeichnet man Informationen, die vom Computern verarbeitet werden können.

Je nach dem, ob die Verarbeitung den Inhalt der Informationen verändert oder nicht, unterscheidet man zwischen Variablen und Konstanten.

Bezüglich ihrer Eigenschaften werden Daten in Datentypen klassifiziert. Ein Datentyp legt dann fest

- welche Werte ein Objekt des Datentyps annehmen kann
- welche Operationen mit einem Objekt des Datentyps durchgeführt werden können.

Datentyp	
einfacher Datentyp	strukturierter Datentyp
Integer	Tabelle
Real	Satz (Record)
Character	Datei
Aufzählungstypen	Liste
Boolean	Baum

## Einfache Datentypen

- Integer:
  - Bezeichnet den Bereich der ganzen Zahlen innerhalb eines vorgegebenen Intervalls. Man ordnet in der Regel einer Variablen den Typ Integer zu, wenn man weiß, dass nur ganzzahlige Ergebnisse zu erwarten sind (Zählvorgänge).
  - Beispiele: 123; 1; 0; -; -234 usw
  - Grundoperationen: Addition +, Subtraktion -, Multiplikation \*, ganzzahlige Division /(ohne Rest), Vergleichsoperatoren = < >
- Real:
  - Daten vom Typ Real sind eine Teilmenge der reellen Zahlen; meistens Dezimalzahlen. Man wählt diesen Datentyp, wenn die erwarteten Werte ganze Zahlen oder Kommazahlen sind.
  - Beispiele: 1,3; 3; 4; 4,5; -1; -3,5 usw.
  - Grundoperationen: Addition +, Subtraktion -, Multiplikation \*, Division /, Vergleichsoperatoren = < >
- Character:
  - Daten vom Typ Character sind Elemente einer endlichen, geordneten Menge von Zeichen. Ein Zeichen ist ein:
    - numerisches Zeichen = Ziffer 0; 1; 2
    - alphabetisches Zeichen = A B C D E a b c d e f
    - Sonderzeichen = , . : ; \_ # & / ? § usw.
  - Werden mehrere Zeichen zu einer Zeichenkette zusammengefasst, spricht man auch von einem String:
  - Grundoperationen: Vergleichsoperatoren = < >
- Boolean:

- Daten vom Typ Boolean bezeichnen den Bereich der logischen Werte, bestehend aus zwei Elementen: WAHR und FALSCH
- Grundoperationen: logisches UND, logisches ODER, logisches NICHT
- Prioritäten der logischen Operatoren: NICHT geht vor UND geht vor ODER

Die bisher beschriebenen Datentypen sind nicht weiter unterteilbar Einheiten. Für diese Datentypen ist eine Ordnung definiert, d. h. die Daten eines Datentyps sind untereinander vergleichbar ( $= < >$ ). Daher nennt man sie auch skalare Datentypen

Fasst man mehrere skalare Daten (möglicherweise unterschiedlichen Datentyps) zu einer Einheit zusammen, die unter einem Namen angesprochen werden kann, so spricht man von strukturierten Datentypen.

## Strukturierte Datentypen

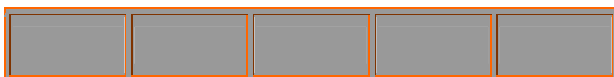
### Tabellen

Eine Tabelle (Array) fasst mehrere Daten gleichen Typs zu einem Datenfeld zusammen. Die einzelnen Bestandteile dieses Datenfeldes heißen Tabellenelemente. Ein bestimmtes Tabellenelement kann über einen Index angesprochen werden. Die Indexvariable muss ein skalarer Datentyp sein.

Es gilt:

- Jede einzelne Komponente einer Tabelle ist explizit ansprechbar und benennbar.
- Die gesamte Tabelle ist ein Datenfeld, das einen eigenen Namen besitzt.
- Die Anzahl der Tabellenelemente sowie der Datentyp der Elemente werden bei der Definition der Tabelle festgelegt und bleiben danach unverändert. Deshalb wird eine Tabelle auch als statische Struktur bezeichnet.
- Der Zugriff erfolgt über den Tabellennamen und Index.
- Eine Tabelle kann ein- oder mehrdimensional sein.

### 1-dimensionale Tabelle



## 2-dimensionale Tabelle


## Sätze (Record)

Ein Record faßt Daten verschiedenen Typs zu einer Einheit zusammen. Ein Satz wird durch die Aufzählung seiner Bestandteile definiert, wobei jeder einzelne Bestandteil einen Namen erhält. Über diese Namen erfolgt der Zugriff auf die einzelnen Felder eines Satzes.

Da der Satzaufbau während der Verarbeitung nicht geändert werden kann, liegt wie bei der Tabelle ein statischer Strukturtyp vor.

## Datei

Eine Datei (File) ist eine Zusammenfassung von mehreren Datensätzen zu einer Einheit unter einem eigenen Namen. In der EDV sind Dateien meist auf speziellen Medien gespeichert.

In Dateien kann man lesen, auf Dateien kann man schreiben.

Bei der Dateiverarbeitung kann festgelegt werden, wo die Datei zu Ende ist. Vor der Verarbeitung muß eine Datei geöffnet werden, nach der Verarbeitung muss sie geschlossen werden.

## Dateiorganisationsformen

- Sequentiell  
Die Daten stehen nacheinander in der Reihenfolge, in der sie geschrieben werden in der Datei.
- Relativ  
Jeder Datensatz wird unter einer bestimmten Satznummer gespeichert.
- Index - sequentiell  
Die Datei ist unterteilt in Index- und Datenbereich.

Der Indexbereich enthält nur den Schlüssel und die Adresse eines Datensatzes.

Der Datenbereich enthält die eigentlichen Datensätze.

Ein Feld des Datensatzes ist als Schlüssel festgelegt, das eindeutig sein muss und nach dem die Datensätze sortiert sind.

Im Indexbereich stehen die Schlüssel immer aufsteigend sortiert, im Datenbereich müssen die Sätze nicht sortiert sein.

Der Zugriff erfolgt über den Schlüssel und den Indexbereich.

Quellen:  
InBit, Paderborn

IT-Ausbildung, Band3  
Entwickeln und Bereitstellen von Anwendungssystemen, Medien-Institut Bremen