

Programmlogik und Darstellungsmittel für Programmabläufe

Abstraktion der Wirklichkeit

- viele Problemstellungen sind umfangreich und kompliziert
- daher ist systematische Vorarbeit erforderlich, um den Lösungsweg zu beschreiben
- man beschränkt sich auf die Dinge, die für das Problem relevant sind
- ein Programm zur Verwaltung eines Bankkontos benötigt Name und Adresse der Person, andere Angaben wie Hobbies u. ä. dagegen nicht
- die Wirklichkeit wird reduziert auf die Dinge, die für das spezielle Problem relevant sind

Programmlogik und Programmablauf

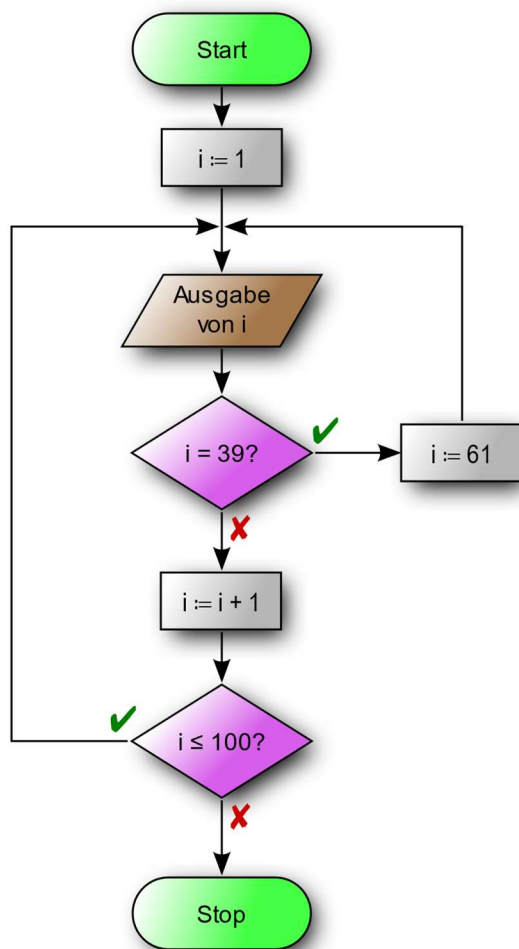
- ein Programmablauf oder Programmfluss beschreibt die Umsetzung des Algorithmus in einem Programm
- es wird eine Folge von Anweisungen (Operationen) ausgeführt
- diese Folge hängt von der Logik des Programms ab
 - sequenziell
 - iterativ
 - selektiv

Programmabläufe visualisieren

- strukturierte Programmierung verlangt eine gut durchdachte Vorbereitung
- Zerlegen der Aufgabenstellung in kleinere, überschaubare Einheiten, die man isoliert betrachte
- und später zu einer Gesamtlösung zusammenfügt

Programmablaufplan

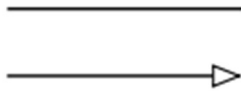
- grafische Darstellung mithilfe genormter Symbole (DIN 66001)



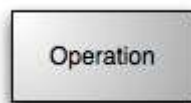
- Kreis; Oval / Rechteck mit gerundeten Ecken: Terminator



- Pfeil, Linie: Verbindung zum nächstfolgenden Element



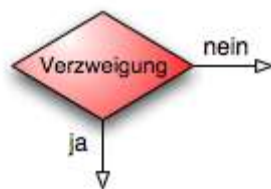
- Rechteck: Operation (Tätigkeit)



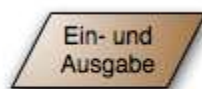
- Rechteck mit doppelten, vertikalen Linien: Unterprogramm ausführen



- Raute: Verzweigung / Entscheidungen

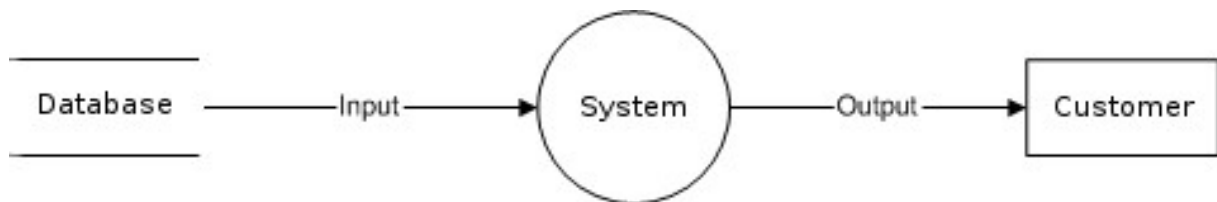


- Parallelogramm: Ein- und Ausgabe



Datenflussdiagramm

- beim Datenflussdiagramm werden vier Elementtypen mit folgender Semantik unterschieden:
 - Datenspeicher: dargestellt durch zwei parallele Linien, zwischen denen der Speichername steht
 - Datenfluss: dargestellt durch einen Pfeil mit Namen. Greift eine Funktion lesend und schreibend auf einen Datenspeicher zu, kann man dies entweder mit zwei getrennten Pfeilen oder mit einem Doppelpfeil darstellen.
 - Funktion (oder Prozess): dargestellt durch einen Kreis mit
 - Schnittstelle zur Umwelt: dargestellt durch ein Rechteck, das den Schnittstellen-namen enthält (externer Partner).
 - Schnittstellen, an denen Daten in das System einfließen, werden Datenquellen genannt.
 - Schnittstellen, an denen Daten aus dem System ausfließen, heißen Datensenken.



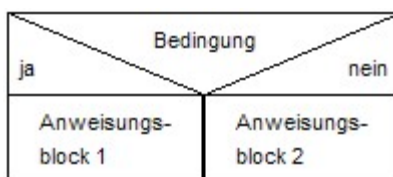
Struktogramme

- Verarbeitung / Folge (Sequenz)



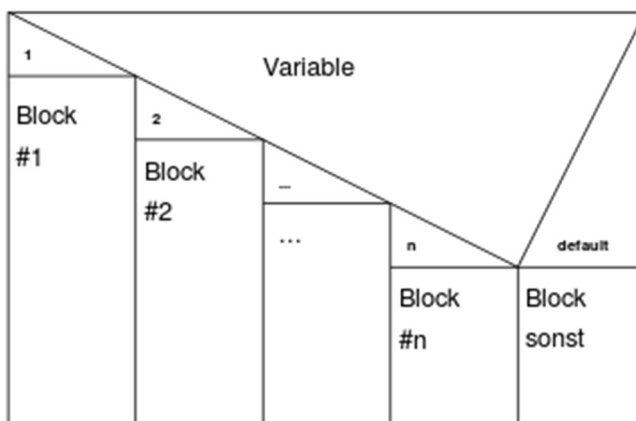
Jede Anweisung wird in einen rechteckigen Strukturblock geschrieben.
Die Strukturblocke werden nacheinander von oben nach unten durchlaufen.
Leere Strukturblocke sind nur in Verzweigungen zulässig.
Alternative Begriffe: Folge, Befehlsfolge, Anweisungsfolge, Anweisungsblock, Linearer Ablauf, Sequenz.

- Bedingte Anweisung und Verzweigung (Selektion)



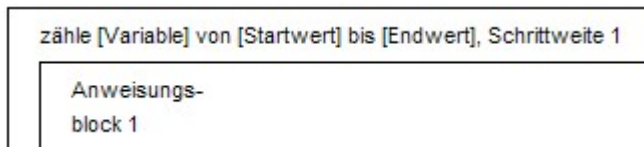
Wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen; trifft die Bedingung nicht zu (falsch), wird der Anweisungsblock 2 durchlaufen (if then else). Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Austritt unten nach Abarbeitung des jeweiligen Anweisungsblocks.
Alternative Begriffe: Einfache Alternative, Zweifache Auswahl, Alternative Verzweigung/Verarbeitung.

- Fallunterscheidung (Mehrfachverzweigung)



Der Wert von „Variable“ kann bedingt auf Gleichheit (Switch Case in Java) aber auch auf Bereiche (größer/kleiner bei Zahlen) geprüft werden. Der entsprechend zutreffende „Fall“ mit dem zugehörigen Anweisungsblock wird durchlaufen (switch, select). Eine Fallauswahl kann stets in eine verschachtelte Auswahl umgewandelt werden – etwa wenn die später eingesetzte Programmiersprache Fallauswahlen nicht kennt.

- Wiederholung oder Schleife (Iteration)



Wiederholungsstruktur, bei der die Anzahl der Durchläufe festgelegt ist (for). Als Bedingung muss eine Zählvariable angegeben und mit einem Startwert initialisiert werden. Ebenso muss ein Endwert und die (Zähl-)Schrittweite angegeben werden. Nach jedem Durchlauf des Schleifenkörpers (Anweisungsblock 1) wird die Zählvariable um die Schrittweite inkrementiert (bzw. bei negativer Schrittweite dekrementiert) und mit dem Endwert verglichen. Ist der Endwert überschritten bzw. unterschritten, wird die Schleife verlassen. Alternative Begriffe: Zählergesteuerte Schleife.

Pseudocode

- Halbformale, textuelle Beschreibung eines Programmablaufs
 - Für Kontrollstrukturen werden z.B. if ... then ... else ... end, while ... end verwendet
 - Blöcke werden mit begin und end markiert
 - es können Variablen- und Konstantendeklarationen enthalten sein
-
- **Aufgabenstellung:**
Zwei Zahlen sollen eingelesen, addiert und anschließend mit 3 multipliziert werden.
-
- **Pseudocode:**
Begin Rechnen
 Definiere zahl1
 Definiere zahl2
 Definiere ergebnis
 Eingabe(zahl1)
 Eingabe(zahl2)
 ergebnis = (zahl1 + zahl2) * 3
 Ausgabe(ergebnis)
End Rechnen

Programmbeispiel in Python

```
# Zahl1 und Zahl2 addieren und anschließend mit 3 multiplizieren
zahl1 = input("Zahl 1 eingeben:")
zahl2 = input("Zahl 1 eingeben:")
ergebnis = (zahl1 + zahl2) * 3
print (zahl1, "+", zahl2, "* 3 =", ergebnis)
```

Anweisung	Bedeutung
#	Kommentarzeile
input(prompt)	Eingabeaufforderung – gibt die Eingabe des Benutzers als String zurück Der Parameter <i>prompt</i> ist optional; hier kann ein String angegeben werden, der vor der Eingabeaufforderung ausgegeben wird
variablenname = ...	Definition einer Variablen durch Zuweisung eines Wertes mit = Zahl, Berechnung oder Zeichenkette (String) möglich Variablennamen beginnen mit einem Buchstaben
+ - * / ()	Addition, Subtraktion, Multiplikation, Division es gelten die Operatorrangfolgen der Mathematik, sofern keine Klammern gesetzt werden
print(...)	Ausgabe der Werte (Variablen und Konstanten) mehrere Werte werden durch Kommata getrennt angegeben