

Struktogramme

Hinweise¹

Stand: 09. 06. 2005

Dr. Reinhard Strangmeier
Lehrstuhl für Wirtschaftsinformatik Prof. Dr. H. Gehring
Fachbereich Wirtschaftswissenschaft
FernUniversität in Hagen

1 Einleitung

Die Ablaufstrukturen² eines strukturierten Programms lassen sich bekanntlich (Kurs 00008, KE 1, S. 63) als aus drei unterschiedlichen Elementarstrukturen zusammengesetzt denken und darstellen. Es sind

1. die Sequenz (Reihung, Abfolge von Anweisungen)
2. die Selektion (Auswahl, alternative Verzweigung auf Grund *einer* Bedingung)
3. die Iteration (Wiederholung, mit Abbruchkriterium)

Sie heißen auch *essentielle Strukturen*. Aus diesen drei Elementen lassen sich alle Ablaufstrukturen eines sequentiellen (d. h. nicht parallelen, nicht-objektorientierten) Programms konstruieren. Daher sind sie für das Verständnis informationstechnischer Systeme von besonderer Bedeutung – wie etwa die vier Aminosäuren, die die DNS bilden, oder das Periodensystem der Elemente in der Chemie.

Wir setzen dabei voraus, dass Eingabe und Ausgabe in geeigneter Weise geregelt sind. Dass also die Eingabedaten zur rechten Zeit am rechten Ort dem Programm zur Verfügung stehen und dass die Ausgabedaten gespeichert, sichtbar gemacht oder der weiteren Verarbeitung in geeigneter Weise zugeführt werden. Fragen der Initialisierung von Variablen oder der Ausgabe von Ergebnissen, die in Programmierkursen mit Recht akzentuiert behandelt werden, können in Struktogrammen berücksichtigt werden, müssen es aber nicht. Das gilt auch für die Fehlerbehandlung.

In der *erweiterten D-Struktur* (benannt nach dem niederländischen Informatiker Dijkstra) kommen ein viertes und ein fünftes Element hinzu:

4. die Auswahlanweisung (Mehrfachverzweigung auf Grund *mehrerer* Bedingungen)
5. die Rekursion.

¹ Dieser Text beantwortet in zusammenfassender, systematisierender Form einige Fragen, die zu Struktogrammen immer wieder gestellt werden. Er ersetzt nicht den Kurstext und darf nur im Zusammenhang mit diesem verwendet werden. Jede nichtauthorisierte Vervielfältigung oder Verwendung ist ausdrücklich untersagt. – Anregungen und Kritik sind stets willkommen.

² Man spricht auch vom "Kontrollfluss" und meint damit die de facto durchlaufene Folge der Bearbeitungsschritte durch ein Programm.

Beide sind jedoch nicht essentiell. Die Mehrfachverzweigung lässt sich beispielsweise als eine Sequenz von Selektionen konstruieren. Auch die Rekursion kann man in essentielle Strukturen auflösen – und muss dies oft sogar.

Weiter findet man in der Praxis *Variationen* der essentiellen Strukturen.

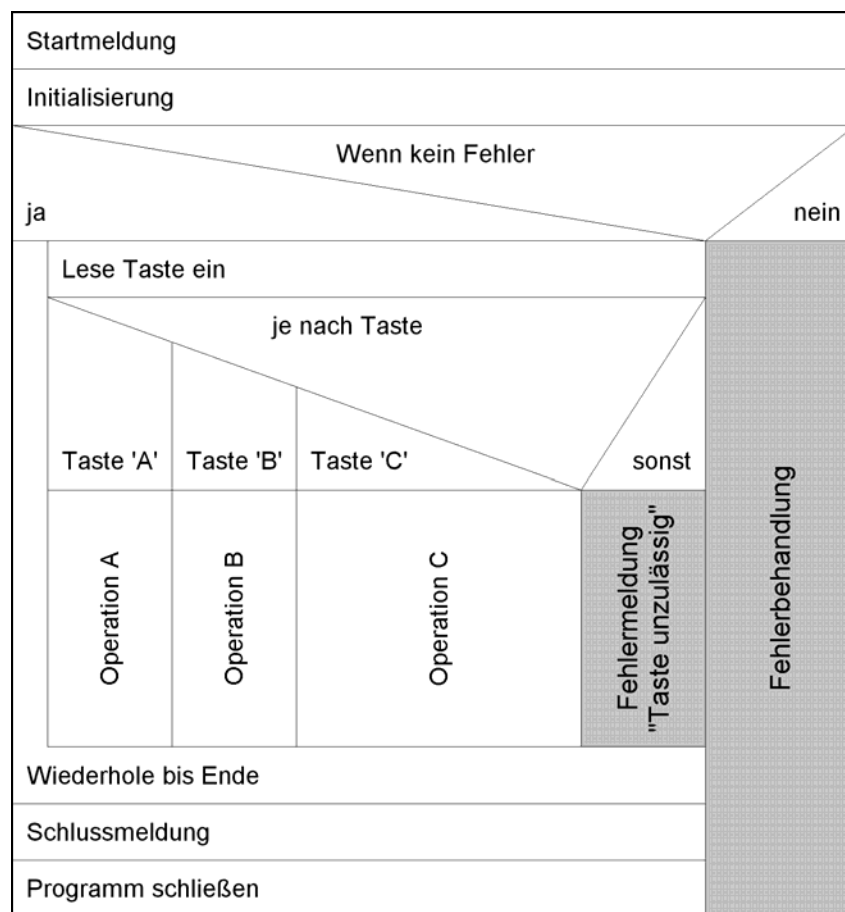
Ablaufstrukturen können schnell unübersichtlich werden. Um sie zu verdeutlichen, werden graphische Hilfsmittel verwendet, zum Beispiel Struktogramme. Struktogramme sollen durch eine entsprechende Visualisierung die menschliche Kommunikation über Ablaufstrukturen unterstützen. Sie sind nicht für eine maschinelle Auswertung ausgelegt.

2 Struktogramme

2.1 Einführungsbeispiel

Zunächst wird an einem Beispiel gezeigt, wie ein Struktogramm zu lesen ist. Anschließend werden die essentiellen Strukturen kurz vorgestellt.

Abb. 1. Struktogramm – Einführendes Beispiel



Quelle: erweiterte und modifizierte Fassung eines Beispiels aus der Dokumentation zu dem Programm WinGESy von M. Denzlein (www.sip-software.de)

Struktogramme sind rechteckig und werden von oben nach unten gelesen.

Am Anfang stehen hier die Anweisungen "Startmeldung" und "Initialisierung". Sie bilden eine Sequenz und werden durch Rechtecke symbolisiert. Innerhalb der Rechtecke werden die Anweisungen beschrieben. Dies kann umgangssprachlich (in so genannter natürlicher Sprache), in einer bestimmten Programmiersprache oder auch in einem geeigneten Pseudocode geschehen. Pseudocode ist eine teilformalisierte Sprache, in der aus Programmiersprachen bekannte Schlüsselworte mit natürlicher Sprache kombiniert werden.

In der dritten Zeile folgt eine Selektion, und zwar in der Variante Alternativanweisung. "Wenn kein Fehler" aufgetreten ist, folgen die unter dem mit "ja" gekennzeichneten Dreieck angeordneten Anweisungen. Wenn "nein" (also ein Fehler aufgetreten ist), folgt die unter dem mit "nein" gekennzeichneten Dreieck angeordnete Anweisung, die als "Fehlerbehandlung" beschrieben ist.

Die einzelnen Kontrollflusselemente beschreiben ebenfalls ein Rechteck. Unter dem "ja"-Dreieck der besprochenen Fehlerabfrage ist dies eine Iteration, und zwar in der Variante einer nichtabweisenden Wiederholungsanweisung, die mit "Wiederhole bis Ende" beschrieben wird. Die Abbruchbedingung "Ende" ist hier nicht näher beschrieben. Die Iteration wird durch das mit "Wiederholung bis Ende" gekennzeichnete winkelförmige Polygon symbolisiert. Die Anweisungen, die an die Innenseiten des Winkels grenzen, bilden den Anweisungsblock, der bei jedem Durchlauf der Iteration angesprochen wird.

Die Anweisung "Lese Taste ein" bedarf wohl keiner Erläuterung. Dann folgt (wieder eine Sequenz!) eine Selektion in der Variante einer Auswahlanweisung (Mehrfachverzweigung). Je nach Taste (Taste 'A', Taste 'B' oder Taste 'C') sollen die Operation A, die Operation B oder die Operation C ausgeführt werden. In allen anderen Fällen soll die Fehlermeldung "Taste unzulässig" ausgegeben werden.

Hinweis: Der "sonst"-Zweig der Auswahlanweisung ist in diesem Beispiel (wie auch sonst häufig) sinnvoll, kann aber entfallen, wenn die Logik des Programms dies nicht erfordert. Dies ist beispielsweise dann der Fall, wenn die Bedingung der Auswahlanweisung alle möglichen Programmezustände umfasst. (Im Kurs 008 ist diese einfachere Form der Auswahlanweisung abgebildet.)

Nach der Iteration folgen zwei weitere Anweisungen, "Schlussmeldung" und "Programm schließen".

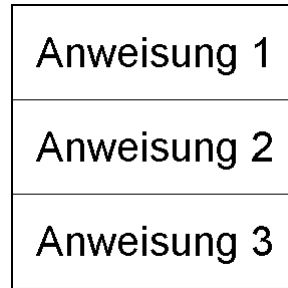
Es sei wiederholt, dass die Eingabe- und Ausgabeanweisungen sowie die Fehlerbehandlung, die in diesem Beispiel enthalten sind, nicht in jedem Struktogramm vorkommen müssen.

Das Beispiel kann auch verdeutlichen, dass die verbale Beschreibung dieser doch recht einfachen Programmstruktur nicht so einfach ist. Daher ist der Einsatz von Struktogrammen zur übersichtlichen und klaren Darstellung ein sinnvolles Hilfsmittel, insbesondere zur visuellen Unterstützung der fachlichen Kommunikation über kritische Ablaufstrukturen.

2.2 Die essentiellen Strukturen (mit einigen Variationen)

2.2.1 Die Sequenz

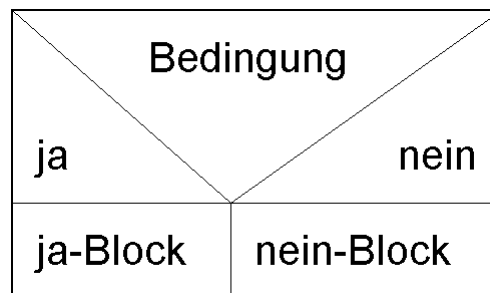
Abb. 2. Sequenz



Sequenzen von Anweisungen werden von oben nach unten abgearbeitet. Eine Menge von einer oder mehreren Anweisungen wird auch Block genannt.

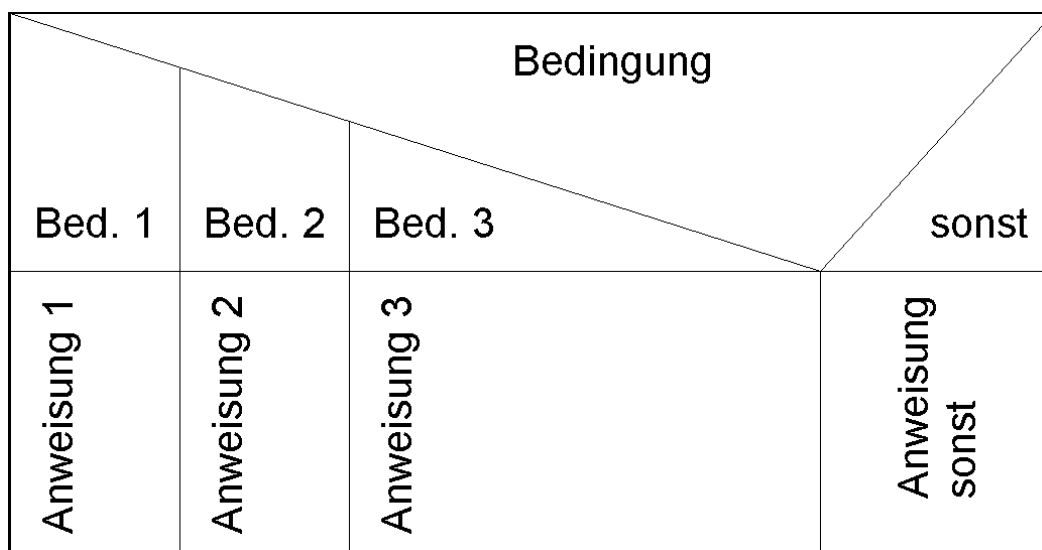
2.2.2 Selektionen

Abb. 3. Alternativanweisung



Falls die Bedingung zutrifft, wird der ja-Block durchgeführt, sonst der nein-Block.

Abb. 4. Auswahlanweisung



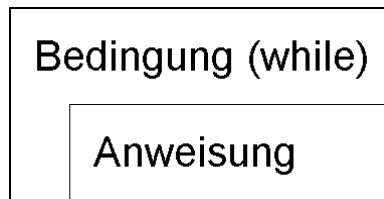
Hier wird je nach Ausprägung der Bedingung in mehrere Anweisungsblöcke bzw. Anweisungen verzweigt. Der rechte Zweig (sonst) wird durchlaufen, wenn keine der enumerierten Bedingungen auf der linken Seite zutreffen. Dieser Zweig ist optional.

Die Auswahlanweisung wird auch Mehrfachverzweigung oder Fallunterscheidung genannt. In vielen Programmiersprachen wird sie durch case-Konstrukte abgebildet.

2.2.3 Iterationen

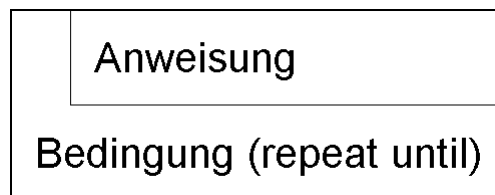
Bei Wiederholungen gibt es ebenfalls mehrere Varianten.

Abb. 5. Abweisende Wiederholungsanweisung
(d. h. mit vorangehender Bedingungsprüfung)



Die Anweisung wird solange wiederholt, wie die Bedingung gegeben ist.

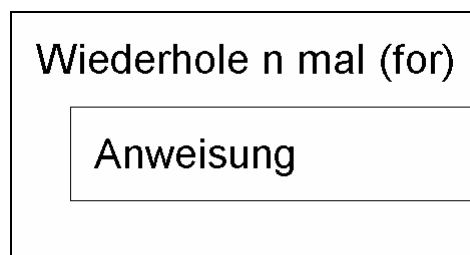
Abb. 6. Nichtabweisende Wiederholungsanweisung
(d. h. mit nachfolgender Bedingungsprüfung)



Die beiden Variationen unterscheiden sich nur darin, dass die Bedingung der abweisenden Wiederholungsanweisung vor Ausführung der Anweisung geprüft wird, während die Bedingung der nichtabweisenden Wiederholungsanweisung jeweils nach Durchlauf der Anweisung geprüft wird. Letztere wird also zumindest einmal ausgeführt.

Anweisungen, deren Zahl der Wiederholungen vor Eintritt in die Iteration festliegt, werden im Struktogramm wie folgt symbolisiert:

Abb. 7. Unbedingte Wiederholungsanweisung



Für unbedingte Wiederholungen gibt es in den meisten Programmiersprachen Konstrukte wie "for i = 1 to n do ...".

Das Symbol in Abb. 7 kann man sich aus den beiden anderen Iterationsvarianten (Abb. 5 und 6) zusammen gesetzt vorstellen.

---###---