

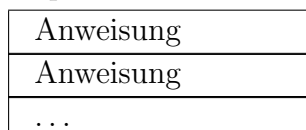
Graphische Darstellung von Programmabläufen

Jede berechenbare Funktion kann aus den drei elementaren Kontrollstrukturen *Sequenz*, *Alternative* und *Iteration* konstruiert werden (Böhm und Jacopini, 1966).

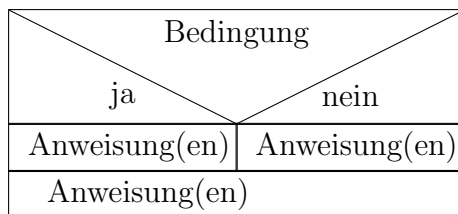
1. Struktogramme

Es gibt verschiedene Modelle, um diese Kontrollstrukturen graphisch darzustellen. Eine davon sind *Nassi-Shneidermann-Diagramme*. Sie werden zwar selten verwendet aber ihre Verwandtschaft mit den Programmbildern von Scratch machen sie für uns interessant.

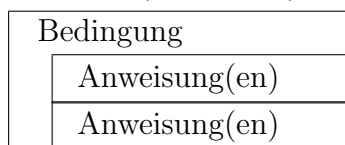
- Sequenz



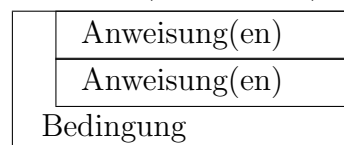
- Alternative



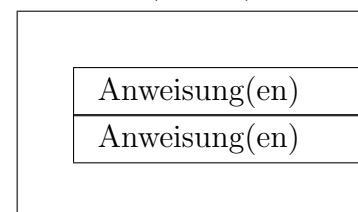
- Iteration (abweisend)



- Iteration (annehmend)



- Iteration (endlos)



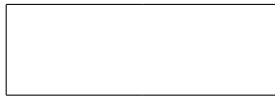
- Es gibt noch wenige zusätzliche Elemente, auf die hier aber nicht eingegangen wird.
- Die oben genannten Konstrukte können auch verschachtelt werden.

2. Programmablaufpläne

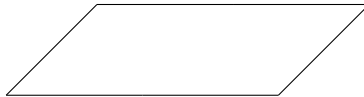
Zur graphischen Darstellung von Programmen können auch Programmablaufpläne (engl. *flowcharts*) verwendet werden.

Auch diese Art der Visualisierung wird meist für kleinere Programme und in der Ausbildung verwendet.

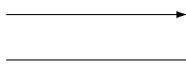
Zuordnungsanweisung:



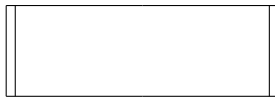
E/A-Anweisung:



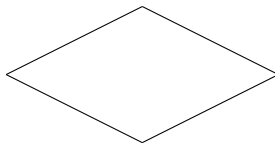
Pfeil/Linie :



Unterprogramm:



Verzweigung:



Start/Stop (Grenzstelle):



3. UML

Grosse bzw. professionelle Softwareprojekte werden heute mit Hilfe der UML (Unified Modelling Language) dargestellt. Mit Hilfe dieser „Modellierungssprache“ und geeigneter Software können aus den UML-Diagrammen sogar automatisch Programmteile erzeugt werden.

- C. Böhm and G. Jacopini. Flow diagrams, Turing machines and languages with only two formation rules. *Communications of the ACM*, pages 366–371, May 1966.