

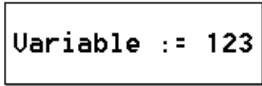
Struktogramme

Es ist sinnvoll sich einen Syntaxablauf vorher zu überlegen. Die Informatik hat dafür einen Diagrammtyp entwickelt, die Nassi-Schneidermann-Diagramme oder Struktogramme. Diese sind nur eine Form um einen Algorithmus darzustellen. Je nach Fragestellung gibt es auch andere Diagrammtypen, mit denen man Datenflüsse, Aktivitäten, Ereignisse oder Verbindungen zwischen Codeteilen angeben kann. Die Struktogramme sind heute in der Softwaretechnik eher unüblich, sie haben aber den Vorteil, dass man bei ihrer Erstellung sich den Algorithmus überlegt, anstatt am Computer so lange zu probieren bis man eine brauchbare Lösung. Ein Algorithmus kann dann 1:1 in Struktogramme umgesetzt werden.

Bestimmte Symbole stehen für bestimmte Syntaxelemente. Sie werden mit der Programmzeile ausgefüllt, die ausgeführt wird. Diese strukturierte Form erlaubt es auch, Algorithmen von einer Programmiersprache leicht in eine andere zu übertragen.

Jedes Struktogramm besteht aus einem grafischen Element und einer Beschriftung. Die Beschriftung kann abstrakt sein (umgangssprachliche Umschreibung oder Semicode, oder echter Programmcode).

Folgende Elemente gibt es:



Variable := 123

Die Zuweisung



Wert := Funktion()

Der Funktionsaufruf



für Variable := Startwert bis Endwert

Die FOR Schleife (Zähler gesteuerte Schleife)



während Bedingung

Die While Schleife (Abweisende, Kopfprüfende Schleife): Hinweis beide Schleifen haben das gleiche grafische Element, aber unterschiedliche Beschriftungen.



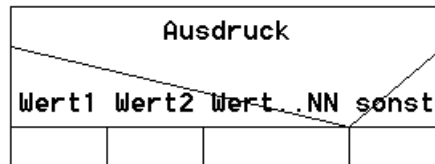
bis Bedingung

Repeat Schleife (Annehmende, nachprüfende Schleife)



Bedingung
wahr **falsch**

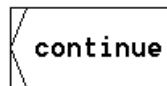
Verzweigung (IF/THEN)



Fallauswahl

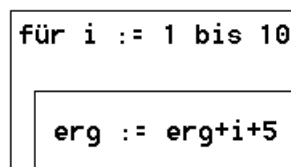


Schleifenabbruch



Schleifenfortsetzung

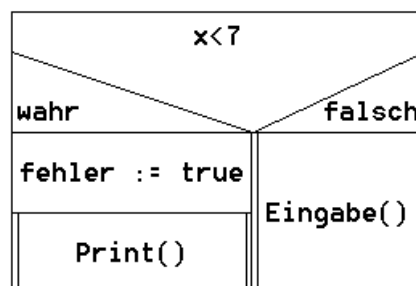
In die leeren Felder kommen dann jeweils eine oder mehrere Anweisungen. Die sie gegebenenfalls in Blöcken (begin end) setzen müssen. So entspricht folgendem Struktogramm



folgendem Code:

```
for i:=1 to 10 do
  erg:=erg+i+5;
```

und folgendes Struktogramm



ergibt folgenden Code:

```
if x<7 then
begin
  fehler:=true;
  Print;
end
else
eingabe;
```

Blockanweisungen werden also nicht aufgenommen, genau so wenig Deklarationen wie Variablendefinitionen.

Die Beschriftung der Symbole ist nicht standardisiert, nur die Symbole selbst. Gängig ist sowohl die Verwendung von Programmcode, wie auch umgangssprachlicher Beschreibung. (Sie werden auch zur Beschreibung betrieblicher Abläufe eingesetzt). **In unseren Übungen setzen wir den Programmcode direkt ein.**