## Programmentwicklung durch schrittweise Verfeinerung

Hierbei handelt es sich um die Lösung einer Programmieraufgabe durch Zerlegen in leichter überschaubare Teilproblem, die entweder elementar gelöst oder weiter zerlegt werden (top-down-Methode). Als Hilfsmittel kommt die Verwendung von Pseudo-Code und von Nassi-Shneiderman-Diagrammen in Betracht.

### Formulierung von Aufgabe 5 mit Pseudocode

Daten: $a = \sum_{i=0}^{m-1} 2^i a_i, \quad b = \sum_{j=0}^{n-1} 2^j b_j \quad (a_i, b_i \in \{0, 1\}) \qquad\qquad (a \cdot b = \sum_{j=0}^{n-1} 2^j a \cdot b_j)$

### 0. Problem

1. $a$, $b$ (hexadezimal) eingeben
2. `/*` Mult. auf Mult. mit Zweierpotenzen und Add. zurückführen `*/`

```
s = 0;
for (j = 0; j < n; ++j)
{
  if (b_j!=0)
  {
     s = s + 2^j a;
  }
}
```

3. $a$, $b$ und $a \cdot b$ (herkömml. Mult.) und nach obiger Methode berechnetes Produkt ausgeben

### 1. Verfeinerungsschritt (zu 2.)

2. $s = 0$; $\tilde{a} = a$; `/*` $\tilde{a}^{(j)} = 2^j a$ `*/`

```
for (j = 0; j < n; ++j)
{
  if (b_j!=0)
  {
     s = s + ã durch Von-Neumann-Addition
  }
  ã = 2 · ã durch Linksshift mit Überlaufbehandlung
}
```

*Problem:* Bestimmung von $b_j$, $n$ ?

*Idee:* $b_j$ schrittweise durch Rechtsshift von b (Bez. $\tilde{b}$) und Abfrage von $\tilde{b}$ `& 1` bestimmen.
Beendigung, falls $\tilde{b} = 0$

**1. Verfeinerungsschritt (zu 2.) - Modifikation**

2.1 $s = 0$; $\tilde{a} = a$; $\tilde{b} = b$; `/*` $\tilde{a}^{(j)} = 2^j a$; $\tilde{b}^{(j)} = b$`>>`$j$ `*/`
2.2 `while (`$\tilde{b}$`!=0)`
`{`
`if ((`$\tilde{b}$`&1) == 1)`
`{`
2.2.1 $s = s + \tilde{a}$ durch Von-Neumann-Addition
`}`
2.3 $\tilde{b}$`>>=1;`
2.4 Falls höchstes Bit in $\tilde{a}$ gesetzt und $\tilde{b} \neq 0$, Überlauf melden
2.5 $\tilde{a}$`<<=1;`
`}`

**2. Verfeinerungsschritt (zu 2.2.1)**

2.2.1 `/*` $s + \tilde{a}$ durch von-Neumann-Addition berechnen, Ergebnis auf $g$ `*/`
$g = s$; $h = \tilde{a}$;
`do`
`{`
*übertrag* $=$ $g$`&`$h$;
Falls höchstes Bit in *übertrag* gesetzt, Überlauf melden
*übertrag* `<<=1;`
*halbaddition* $=$ $g$`^`$h$;
$g =$*halbaddition*; $h =$*übertrag*;
`}`
`while (`*übertrag*`!=0);`

`/*` $s = s + \tilde{a}$ `*/`
$s = g$;

**3. Verfeinerungsschritt**

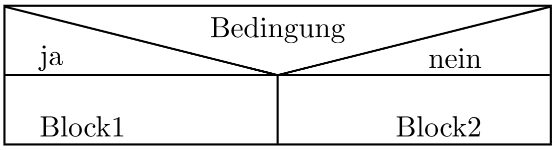
*Programm*

Wird die schrittweise Verfeinerung direkt in der Programmquelle vorgenommen, so empfiehlt es sich, die noch nicht ausgeführten Programmteile als Kommentare zu formulieren.

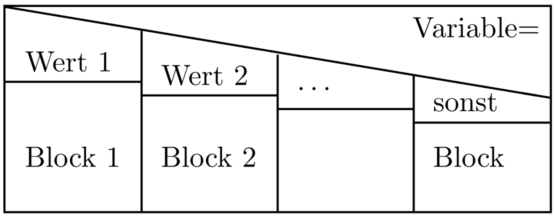
# Struktogramme (Nassi–Shneiderman–Diagramme) (1973)

**Elementare Strukturblöcke**

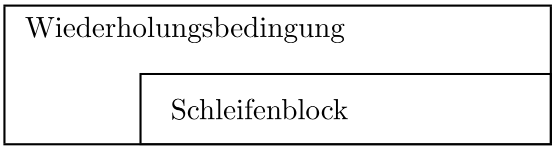
Einfacher **Anweisungs**–Strukturblock



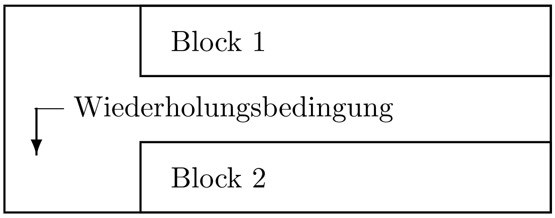

**Auswahl**–Strukturblock mit **Zweifach**verzweigung



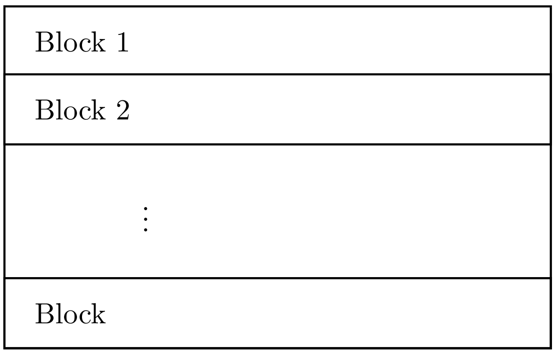

**Auswahl**–Strukturblock mit **Mehrfach**verzweigung




**Wiederholungs**–Strukturblock mit **Vorabtest**




**Wiederholungs**–Strukturblock mit Prüfung in der Schleife



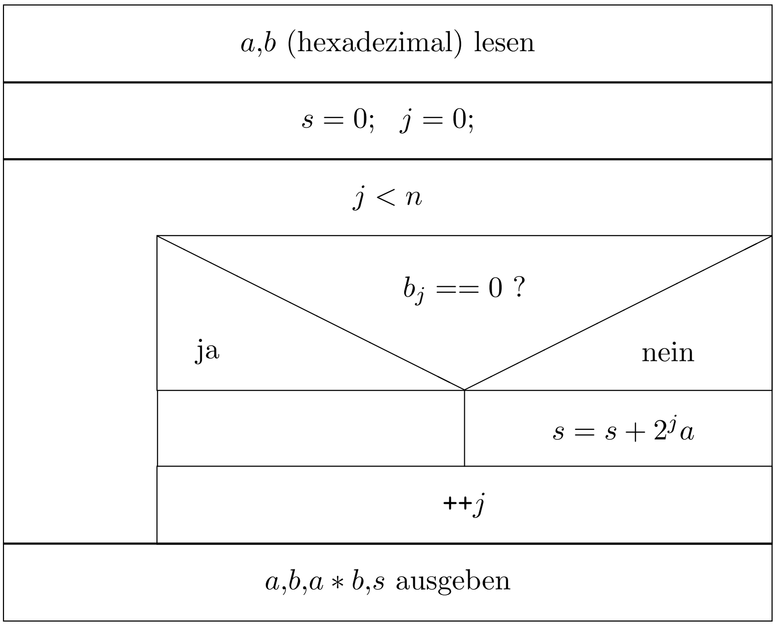

**Zusammengesetzter** Strukturblock

Jeden Unterblock eines elementaren Strukturblockes darf man durch einen beliebigen elementaren Strukturblock ersetzen.

**Formulierung mit Nassi-Shneiderman-Diagrammen**

Daten: $a = \sum_{i=0}^{m-1} 2^i a_i, \quad b = \sum_{j=0}^{n-1} 2^j b_j \quad (a_i, b_i \in \{0,1\})$

**0. Problem**




*usw.*