

Pseudocode Merkhzettel

Methoden:

- Funktions-Körper:
 function NAME(Param1, Param2,...)
 ...
 end function
- Aufruf:
 NAME(*a*, *b*,...)

Zuweisung:

- $a := b$ (Weist *a* den Wert *b* zu.)
- $a := b + c$ (Weist *a* die Summe von *b* und *c* zu.)
- $A := B$ (Weist der Menge *A* die Menge *B* zu.)

Schleifen:

- **for** *a* **in** *b*, ..., *c* **do**
 Führe Aktionen aus bis *a* den Wert *c* erreicht.
 Initial besitzt *a* den Wert *b*
 Variable *a* ist in diesem Block nutzbar.
 a wird am jede Iteration um 1 erhöht.
 end for
- **for** $a := b$ (**down**)**to** *c* **do**
 Führe Aktionen aus bis *a* den Wert *c* erreicht.
 Initial besitzt *a* den Wert *b*
 Variable *a* ist in diesem Block nutzbar.
 a wird am jede Iteration um 1 erhöht bzw. verringert.
 end for
- **while** Bedingung **do**
 Führe Aktionen aus, *solange* eine Bedingung gilt.
 end while
- **repeat**
 Führe Aktionen aus, *bis* eine Bedingung gilt.
 until Bedingung

Bedingte Anweisung:

```
if Bedingung then
    Aktionen, falls Bedingung wahr ist
else
    Aktionen, falls Bedingung falsch ist
end if
```

Ausgaben/Rückgaben:

- **print** *a* (*Gibt a aus und beendet die Funktion nicht*)
- **return** *a* (*Gibt a zurück und beendet die Funktion*)

Varianten

Der Stil von Pseudocode ist in der Regel nicht einheitlich und kommt auch in der Literatur in verschiedenen Varianten vor. Mögliche Varianten sind:

- Auf deutsch (*Für, Solange, Falls,...*)
- Mischung mit Prosatext („Sei *e* eine Kante aus *G...*“)
- Weglassen von *end function*, *end if*, etc. Dafür wird eine Einrückung von Blöcken umso wichtiger.