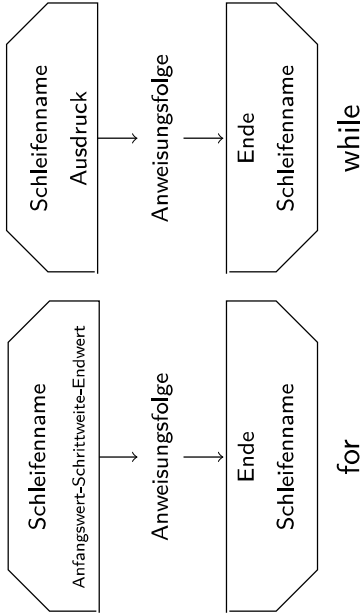


- Schleifen

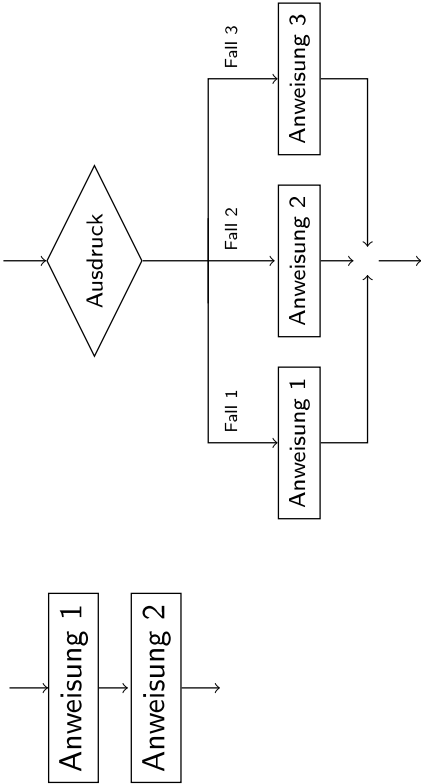


Z.B: Flussdiagramme, Ablaufdiagramme

- Endliche gerichtete knotenmarkierte Graphen
- Grafische Darstellung des Kontrollflusses zwischen Verarbeitungsschritten

Notation

- Sequenz
- Auswahl



Notation

- Algorithmus-anwendung
- Ein-/Ausgaben



Notation



Vorteile

- Übersichtlicher als Programmcode (Visuelle Klarheit)
- Bieten Möglichkeit zur allgemeinen Prozessbeschreibung
- Zwingen zu disziplinierter Gestaltung Programmablauf

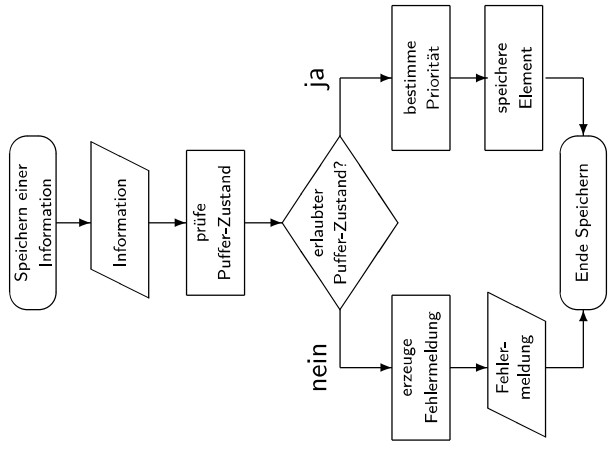
Nachteile

- Ausschließliche Ablauforientierung („Wie“ statt „Was“)
- Keine differenzierte Darstellung von Datenstrukturen
- Ungeeignet für große Systeme:  
Fehlende Abstraktions- und Modularisierungskonzepte, zu detaillierte Darstellung

- Entwurfsphase
- Ziele & Gliederung
- Systemarchitektur
- Komponentendiagramme
- Verteilungsdiagramme
- Paketdiagramme
- Klassendiagramme
- Objektdiagramme
- Funktionsbäume
- Entitäten und Beziehungen
- Datenkataloge & Syntaxdiagramme
- Datenflussdiagramme
- Programmablaufpläne
- Struktogramme
- Zusammenfassung

- Entwurfsphase
- Ziele & Gliederung
- Systemarchitektur
- Komponentendiagramme
- Verteilungsdiagramme
- Paketdiagramme
- Klassendiagramme
- Objektdiagramme
- Funktionsbäume
- Entitäten und Beziehungen
- Datenkataloge & Syntaxdiagramme
- Datenflussdiagramme
- Programmablaufpläne
- Struktogramme
- Zusammenfassung

Beispiel



Ent.12 Struktogramme

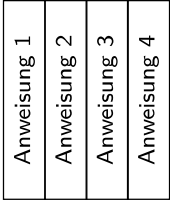
(auch: Nassi-Shneidermann-Diagramme)  
Grafische Darstellung von Kontrollstrukturen

Notation

- Anweisung
- Sequenz

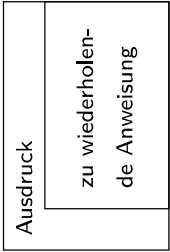


- Aufruf



Notation

- Kopfgesteuerte Schleife (`while`)
- Fußgesteuerte Schleife (`do while`)

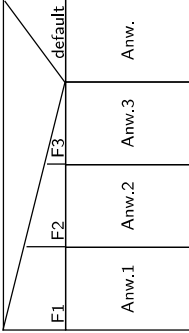
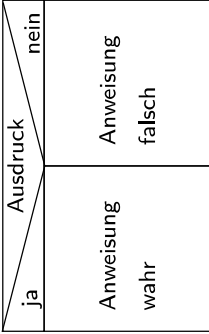


- Entwurfsphase
- Ziele & Gliederung
- Systemarchitektur
- Komponentendiagramme
- Verteilungsdiagramme
- Paketdiagramme
- Klassendiagramme
- Objektdiagramme
- Funktionsbäume
- Entitäten und Beziehungen
- Datenkataloge & Syntaxdiagramme
- Datenflussdiagramme
- Programmablaufpläne
- Struktogramme
- Zusammenfassung

- Entwurfsphase
- Ziele & Gliederung
- Systemarchitektur
- Komponentendiagramme
- Verteilungsdiagramme
- Paketdiagramme
- Klassendiagramme
- Objektdiagramme
- Funktionsbäume
- Entitäten und Beziehungen
- Datenkataloge & Syntaxdiagramme
- Datenflussdiagramme
- Programmablaufpläne
- Struktogramme
- Zusammenfassung

Notation

- Bedingte Anweisung
- Fallunterscheidung



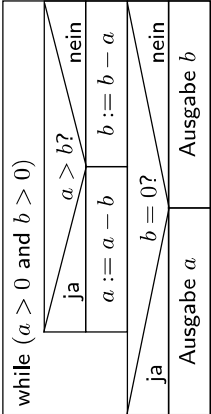
Grundelemente eines Struktogrammes

- Anweisung
- Bedingte Anweisung
- Fallunterscheidung
- Kopfgesteuerte Schleife
- Fußgesteuerte Schleife
- Aufruf

- Entwurfsphase
- Ziele & Gliederung
- Systemarchitektur
- Komponentendiagramme
- Verteilungsdiagramme
- Paketdiagramme
- Klassendiagramme
- Objektdiagramme
- Funktionsbäume
- Entitäten und Beziehungen
- Datenkataloge & Syntaxdiagramme
- Datenflussdiagramme
- Programmablaufpläne
- Struktogramme
- Zusammenfassung

- Entwurfsphase
- Ziele & Gliederung
- Systemarchitektur
- Komponentendiagramme
- Verteilungsdiagramme
- Paketdiagramme
- Klassendiagramme
- Objektdiagramme
- Funktionsbäume
- Entitäten und Beziehungen
- Datenkataloge & Syntaxdiagramme
- Datenflussdiagramme
- Programmablaufpläne
- Struktogramme
- Zusammenfassung

Beispiel



```
while (a>0 && b>0) {  
  if (a>b) {  
    a:=a-b;  
  } else {  
    b:=b-a;  
  }  
  if (b=0) {  
    System.out.print(a);  
  } else {  
    System.out.print(b);  
  }  
}
```

- Vorteile**
- Übersichtlicher als Programmcode (Visuelle Klarheit)
  - Gute grafische Darstellung linearer Kontrollstrukturen (Sprünge sind nicht vorgesehen)
  - Zwingen zu disziplinierter Gestaltung Programmablauf

**Nachteile**

- Manuell aufwendig zu zeichnen und zu ändern
- Nicht geeignet um komplexe Algorithmen darzustellen
- Abstrahieren kaum