



- Ablauf und Aufbau eines Programms wird vor der Programmierung mit Hilfe grafischer Symbole dargestellt
- unabhängig von der später zu verwendenden Programmiersprache
- Beschreibungsmittel:
 - Datenflusspläne
 - Programmablaufpläne
 - Struktogramme



Der Datenflussplan

- dient zur **Darstellung des Informationsflusses** in einer Anwendung
- zeigt
 - ⇒ die **Daten, die verarbeitet** werden,
 - ⇒ welche Daten von einer Verarbeitungsfunktion **eingelezen** und welche **ausgegeben** werden,
 - ⇒ die **Datenträger** und
 - ⇒ den **Typ der Daten**, die sie enthalten
- deutet die Verarbeitungsfunktion nur stichwortartig an



Beispiel für einen DFP

- 1) Fertiggestellte Kundenaufträge eingeben,
- 2) der Fakturierung melden
- 3) Zugriff auf die
 Stammdaten
 Kunden
 Teile
 und die Vormerkdaten
 Aufträge
- 4) Anfertigung der Kundenrechnung
- 5) Ausdruck der Kundenrechnung
- 6) Umsatzdaten der verkauften Teile in Transferspeicher ablegen
- 7) Berichtgenerierung durch Vertriebserfolgsrechnung:
 Auslesen der Informationen aus dem Transferspeicher
 Verknüpfung mit den Kundenstammdaten.



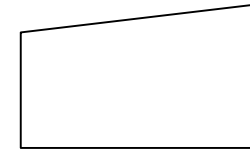
Bearbeiten



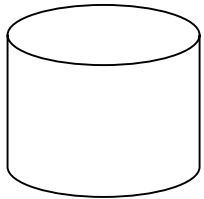
Anzeige



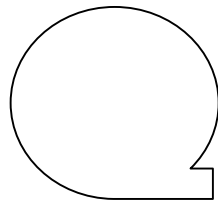
Schriftstück



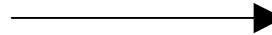
Eingabe von Hand



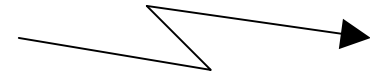
Plattenspeicher



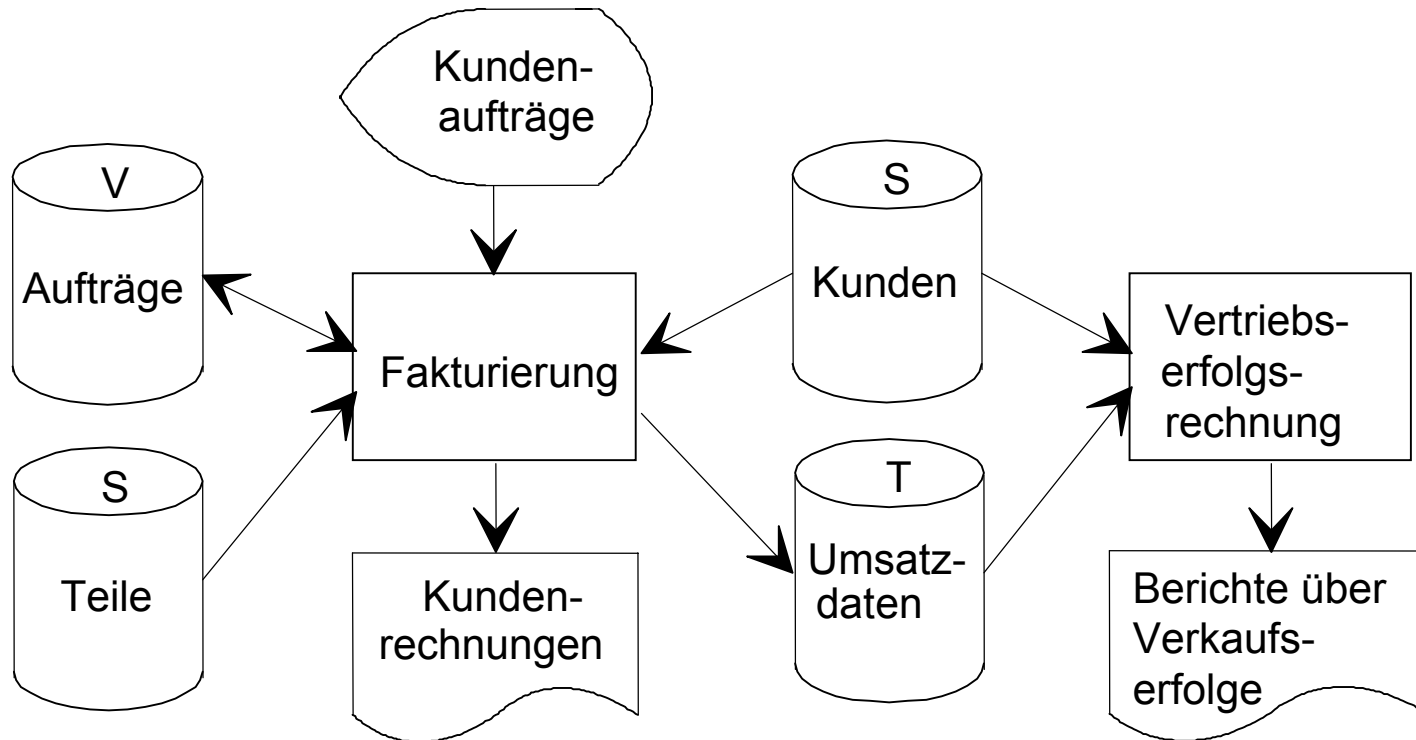
Magnetband



Flußlinie



Datenfernübertragung



Legende: S = Stammdatenspeicher V = Vormerkdatenspeicher T = Transferdatenspeicher

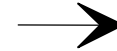
Bearbeiten

Eingabe/
Anzeige

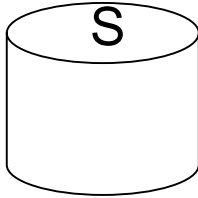
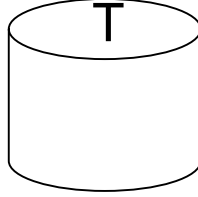
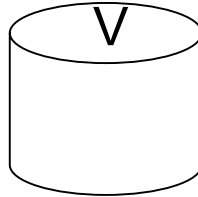
Schriftstück

Platten-
speicher

Flußrichtung





	Wesen	Beispiel	Symbol
Stamm- daten- speicher	<ul style="list-style-type: none">•Selten zu verändern•Zugriff: direkt & oft	<ul style="list-style-type: none">•Personaldaten•Kundendaten•Lieferantendaten	
Transfer- daten- speicher	<ul style="list-style-type: none">•Von einem Programm erzeugt•von einem anderen verarbeitet	<ul style="list-style-type: none">•Umsatzdaten•Stechuhr	
Vormerk- daten- speicher	<ul style="list-style-type: none">•Existieren, bis definiertes Ereignis eintritt	<ul style="list-style-type: none">•Offene Posten	

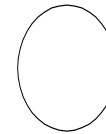
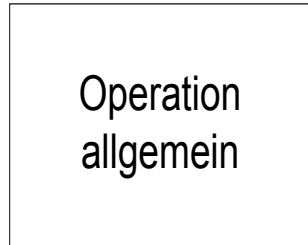


PAP

- Bringt Anweisungen in die benötigte Reihenfolge (zeitlich & logisch)
- beliebig detaillierbar

Struktogramm

- Nassi-Shneidermann-Diagramm
- Alternative zum PAP



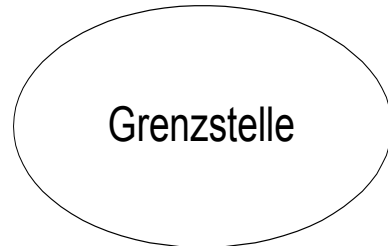
Übergangsstelle



Zusammenführung



Verzweigung



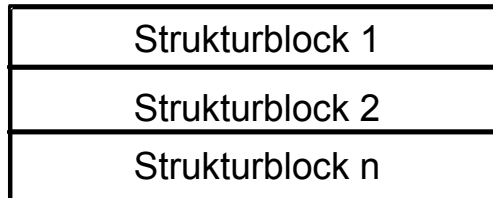
Grenzstelle



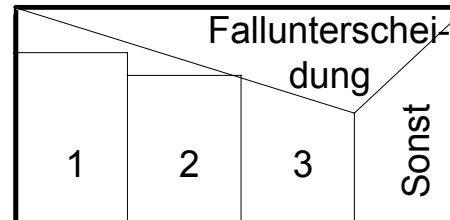
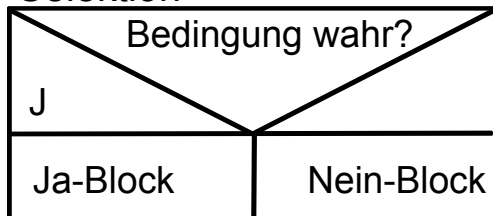
Ablauflinie

**Kontrollstrukturen:**

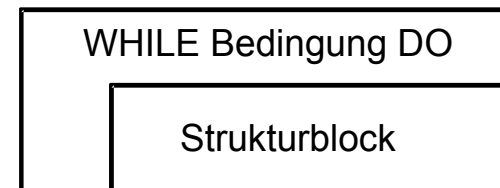
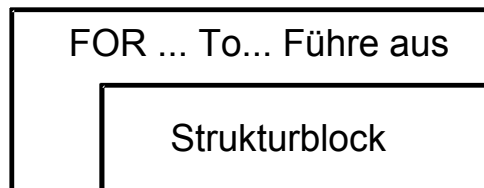
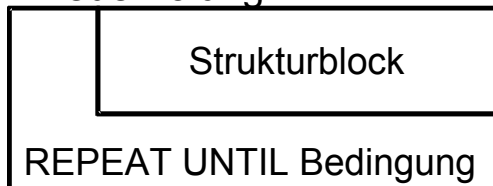
Sequenz



Selektion



Wiederholung





	DFP	PAP	Strukto
Vorteile	Zeigt detailliert Eingabe und Ausgabe, Speichermedien etc.	Beliebig detaillierbar	Beliebig detaillierbar nur 3 Konstrukte erzwingt strukturierte Programmierung
Nachteile	Eigentlicher Programmablauf wird nicht dargestellt („Back Box“)	Evtl. unübersichtliche Struktur.	Gewöhnungsbedürftige „Denke“