

# **Vorlesung Softwaretechnik - Definitionsphase, Datenorientierte, Algorithmisch- und regelbasierte Sicht -**

Prof. Klaus-Peter Fähnrich

Wintersemester 2008/2009

## Überblick LE 8

### LE 8: Datenorientierte Sicht

- Entity-Relationship-Modell
  - Vergleich ER-Konzepte vs. OO-Konzepte
  - Schlüssel, Tabellen und Dateien
  - Semantische Datenmodelle
  - Unternehmensdatenmodelle und Weltmodelle
- Modellierung multidimensionaler Datenstrukturen
  - Data Warehouse, Data Marts, OLAP und Hyperwürfel
  - Modellierungsansätze
  - Abbildung auf Tabellen
- Data Dictionary
- Zusammenfassung

### LE 9: Algorithmisch- und regelbasierte Sicht

## Lernziele

1. Erläuterung der Begriffe Entität, Entitätstyp, Entitätsmenge, Beziehung, Beziehungstyp, Attribut und Schlüssel;
2. Kardinalitäten;
3. Konzepte Aggregation und Vererbung;
4. Unterschiede zwischen ER- und OO-Modell;
5. Begriffe Data Warehouse, Data Mart, Hyperwürfel, multidimensionale Datenstruktur, OLAP und Data Mining;
6. Erstellen eines ER-Modells, einer Assoziationsmatrix für eine gegebene Aufgabenstellung;
7. Erstellen eines Hyperwürfels und seiner Darstellung durch einen ER-Modell;
8. Definition und hierarchische Anordnung von Datenstrukturen als Data-Dictionary-Einträge.

<p>Konzepte und Sichten</p> <p>← häufig verwendet</p> <p>→ selten verwendet</p>										
Funktionsbaum	Geschäftsprozess (1987)	Datenflussdiagramm (1966)	Data Dictionary (1979)	ER (Entity Relationship) (1976)	Klassendiagramm (1980/90)	Pseudocode	Regeln	Zustandsautomat (1954)	Petri-Netz (1962)	Sequenzdiagramm (1987)
Funktionale Hierarchie	Arbeitsablauf	Informationsfluss	Datenstrukturen	Entitätstypen & Beziehungen	Klassenstrukturen	Kontrollstrukturen	wenn-dann Strukturen	Endlicher Automat	Nebenläufige Strukturen	Interaktionsstrukturen
Funktionale Sicht			Datenorientierte Sicht		Objektorientierte Sicht	Algorithmische Sicht	Regelbasierte Sicht	Zustandsorientierte Sicht		Szenario-basierte Sicht
LE5			LE8		LE6-7	LE9	LE9-10	LE11-12		LE7

## Definition

- **Entity-Relationship-Modell (ER-Modell)**

- 1976 von P. Chen zur Datenmodellierung entwickelt

- **Ziel**

- Beschreibung der permanent gespeicherten Daten und ihre Beziehungen untereinander.
- Die Analyse der Information erfolgt aus fachlogischer Sicht.

- **Ergebnis**

- Konzeptionelles Modell, das gegen Veränderungen der Funktionalität weitgehend stabil ist.
- Semantische Datenmodellierung
- Erweiterung um Aggregation und Vererbung

## Schlüssel im ER-Modell

### Schlüssel

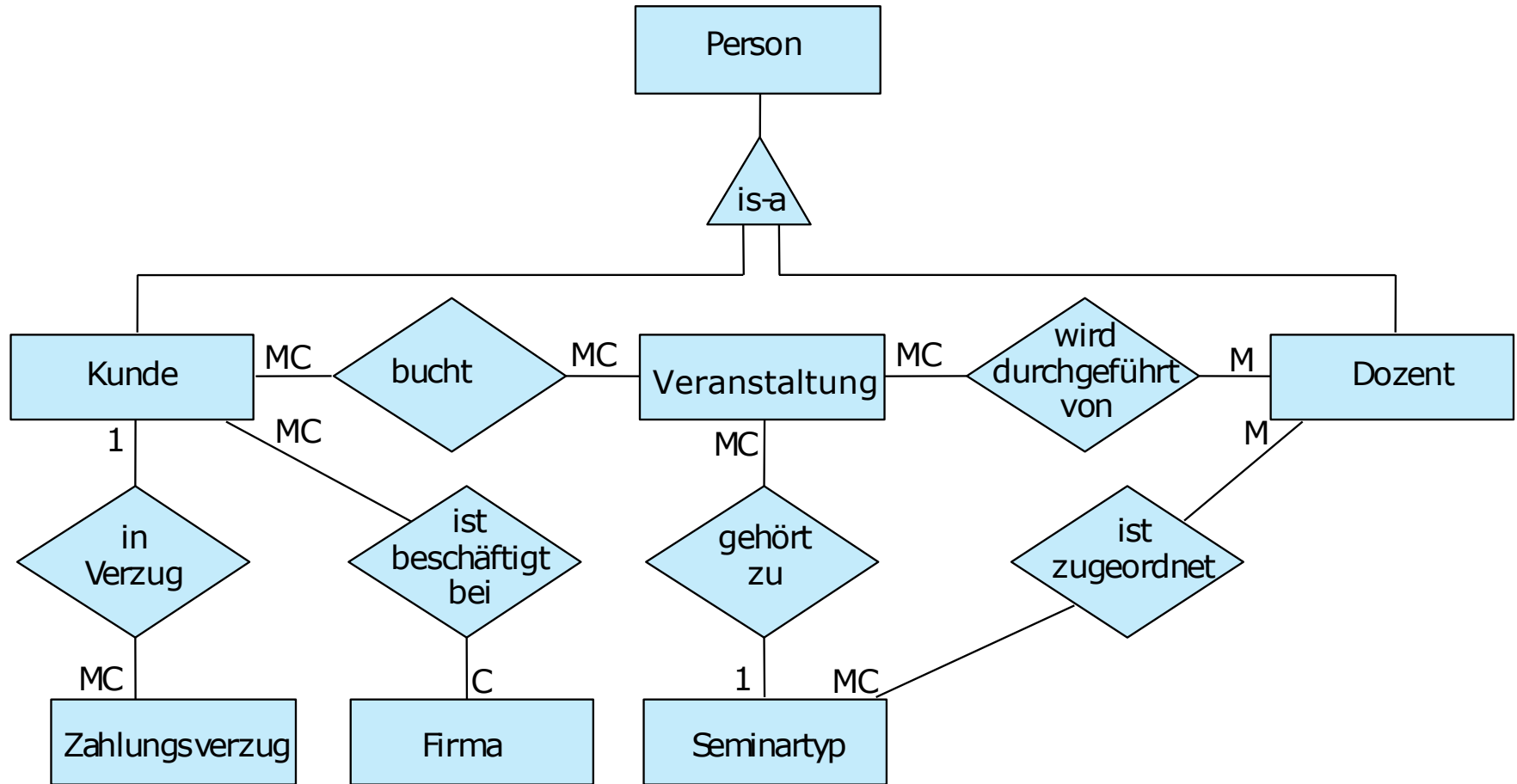
- Minimale, identifizierende Attributkombination;
- Schlüssel werden unterstrichen;
- Oft auch mehrere Schlüssel möglich;
- *Beispiel: Entitätsmenge: Stadt*  
*Attribute: PLZ, Staat, Einwohnerzahl, Vorwahl*  
*Schlüssel: PLZ und Staat oder Vorwahl;*
- In einem solchen Fall wird stets ein Schlüssel als Primärschlüssel ausgezeichnet;
- Ein Schlüssel K ist stets eine minimale, identifizierende Attributkombination;
- Jede echte Obermenge von K ist ein Schlüsselkandidat;
- *Beispiel: Für Stadt ist {Staat, Vorwahl} Schlüsselkandidat, Vorwahl ist Schlüssel, {PLZ, Staat} ist ebenfalls Schlüssel;*
- Minimal: Die eindeutige Identifizierbarkeit geht verloren, wenn ein Attribut entfernt wird.

# Entitäten im ER-Modell

## Entitäten

- Entitätstypen mit ihren Entitäten lassen sich durch Tabellen darstellen.
- Regel 1
  - Für jeden Entitätstyp wird ein Speicher bzw. eine Datei benötigt.
  - Jede Entität des entsprechenden Entitätstyps stellt einen Eintrag in diese Datei dar.
- Regel 2
  - Sind zwei Entitätstypen A und B durch einen 1:1- oder M:1-Beziehungstyp verbunden, dann wird der Primärschlüssel von B als sogenannter Fremdschlüssel in A eingetragen, d. h. als zusätzliches Attribut.
- Regel 3
  - Sind zwei Entitätstypen A und B durch einen M:N-Beziehungstyp verbunden, dann wird für den Beziehungstyp eine eigene Datei angelegt.
  - Als Attribute werden die Schlüssel der Entitätstypen verwendet, die der Beziehungstyp verbindet.
  - Beide Schlüsselattribute zusammen bilden den Primärschlüssel dieser Tabelle.
- Einem Beziehungstyp kann man ebenfalls Attribute zuordnen, wenn dies fachlich notwendig ist.

**Beispiel: Semantisches Datenmodell (MC-Notation)**





## Beispiel für semantische Datenmodelle

### Zusammenhang mit Funktionsbäumen

- Die in den Funktionsbäumen beschriebenen Funktionen benutzen die in ER-Diagrammen modellierten Datenstrukturen, um ihre Aufgabe zu erfüllen.
- Eine Möglichkeit den Zusammenhang zwischen beiden Sichten herzustellen, ist die Erstellung einer Assoziationsmatrix.

### Bewertung

- Vorteile
  - Im kaufmännischen Anwendungsbereich ist eine semantische Datenmodellierung ein absolutes Muss.
  - Auch in vielen technischen Bereichen ist die Komplexität der Daten so groß, dass ein semantisches Datenmodell erforderlich ist (Beispiel: Roboter-Modellierung).
  - Voraussetzung für einen relationalen Datenbankentwurf.
- Nachteile
  - Dynamische Aspekte können nicht dargestellt werden.
  - Erfordert ein höheres Abstraktionsniveau als die Konzepte zur Modellierung der funktionalen Sicht.
  - Semantische Datenmodelle können sehr umfangreich werden.
  - Es fehlt ein Verfeinerungsmechanismus um mehrere Abstraktionsebenen bilden zu können.

## Unternehmensdatenmodell

### **Probleme bei der Betrachtung nur eines Bereiches eines Unternehmens:**

- Mehrfachverwaltung der gleichen Datenbestände
- Brüche bei der Abwicklung von übergreifenden Geschäftsvorgängen
- Inkompatible Informationsflüsse

### **Ziel**

- Aufstellung eines umfassendes Unternehmensdatenmodells.

### **Unternehmensdatenmodell**

- Soll die Informationsstrukturen und Geschäftsprozesse aller Bereiche eines Unternehmens unter Berücksichtigung der Schnittstellen zueinander in einheitlicher und ganzheitlicher Form darstellen

### **Beschreibung von Unternehmensdatenmodellen**

- Semantische Datenmodelle
- Je nach Abstraktionsebene und Verdichtungsgrad können Datenmodelle unterschiedlicher Ebenen definiert werden.

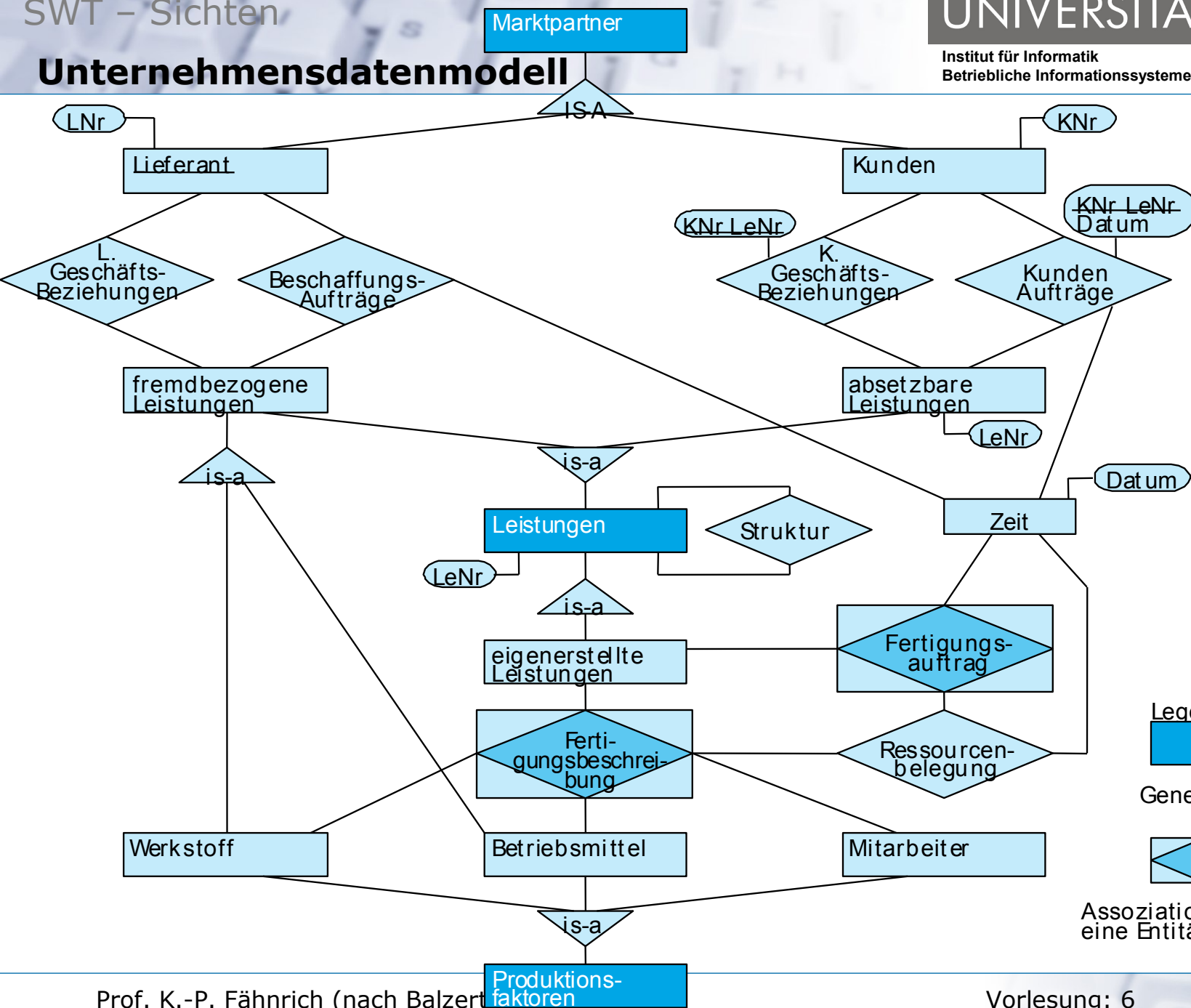
### **Strategisches Datenmodell**

- Enthält nur die wichtigsten Entitätstypen und Beziehungen
- Enthält nur grundsätzliche Branchenunterschiede
- Umfasst in der Regel rund 20 bis 30 Entitätstypen und Beziehungen

### **Nächste Abstraktionsebene**

- Zeigt bereits unterschiedliche Strukturen innerhalb einer Branche

# Unternehmensdatenmodell



# Multidimensionale Datenstrukturen

## Data Warehouse

- Themenorientierte, integrierte, zeitabhängige, nichtflüchtige Datensammlung zur Unterstützung von Managemententscheidungen.

## Data Marts

- Stellen funktionsbereichs- oder personengruppenspezifische Extrakte aus der Data Warehouse-Datenbasis zur Verfügung und
- erlauben dadurch dezentrale fachliche Sichten auf das zentrale Data Warehouse.

## Olap (On-line Analytical Processing)

- Beschreibt die Datenmodellierung in multidimensionalen Strukturen zur Datenanalyse.
- Bei multidimensionaler Aufbereitung können die Daten vom Benutzer schneller und aussagekräftiger zur Entscheidungsfindung analysiert werden.

## Data Mining

- Bezeichnet die automatische Suche nach unbekanntem Zusammenhängen in großen Datenbeständen.
- In einem umfassenden Datenbestand werden eine Vielfalt von Zusammenhängen entdeckt, die dann noch bezüglich ihrer Aussagekraft und Relevanz interpretiert werden müssen.

→ (siehe Prof. Rahm, *Datawarehousing*)

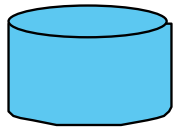
# Architekturstufen

- detailliert
- tagtäglich
- aktueller Wert
- hohe Zugriffswahrscheinlichkeit
- anwendungsorientiert

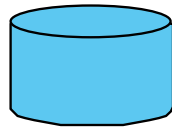
- nur Lese-Zugriff
- meist verdichtet
- zeitvariant
- integriert
- fachorientiert
- zusammenfassend

- beschränkt
- teils abgeleitet
- teils einfach

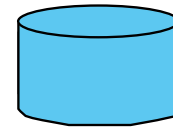
- temporär
- ad hoc
- heuristisch
- nicht wiederholend



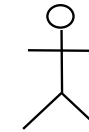
Operatives System



Data Warehouse



Data Mart



Kundensachbearbeiter

Name = Hans Schulz  
Umsatz = 6600,-

Name = Hans Schulz  
Umsatz = 2000,-  
Datum = 1.3.99

Name = Hans Schulz  
Umsatz = 4200,-  
Datum = 1.7.99

Name = Hans Schulz  
Umsatz = 6600,-  
Datum = 1.12.99

1.Q. 1999: 35000,-  
2.Q. 1999: 67000,-  
3.Q. 1999: 78000,-  
4.Q. 1999: 99000,-

Wie viele Kunden haben seit dem 1. Quartal 1999 ihre Umsätze pro Quartal um mehr als 10% erhöht?

Aktuell

Historisch

Periodisch

Temporär

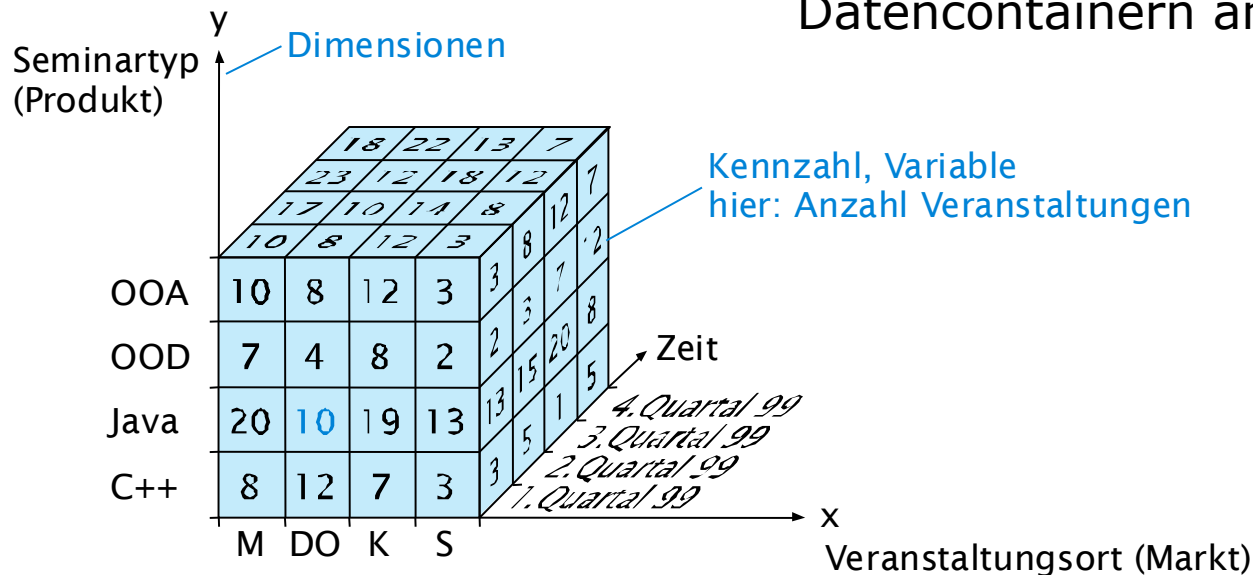
# OLAP und Hyperwürfel

OLAP-Systeme sollen...

- Fachabteilungen die Möglichkeit geben, schnelle, interaktive und komplexe Analysen durchzuführen;
- die gesammelten Daten aus möglichst vielen verschiedenen Perspektiven zeigen;
- die Analyseergebnisse in akzeptabler Zeit bereitstellen.

Hyperwürfel

- Quantitative Kennzahlen werden in mehrdimensionalen Datencontainern angeordnet.



## Data Dictionary

Verzeichnis, das Informationen über die Struktur von Daten, Ihre Eigenschaften sowie ihre Verwendung enthält. Wird zur Konsistenzüberwachung eines Datenbestandes benötigt.

### Erstellungsregeln

- Definition der Daten: top-down
- Ende der Verfeinerung, wenn ein Abstraktionsniveau erreicht ist, das für die Definitionsphase ausreicht.
- Wiederholungs- und Optionsklammern (Backus-Naur-Form) möglichst weit oben in der Hierarchie.
- Zirkuläre Definitionen sind nicht erlaubt.
- Alias-Namen sind erlaubt (für Sonderfälle).
- Problembezogene Namen wählen
- Bereits definierte Datenstrukturen wiederverwenden
- Nicht versuchen, Semantik durch Syntax auszudrücken
- Auswahl besteht aus mindestens zwei Alternativen

**Vorteil:** Aufbau und Verfeinerung von Datenstrukturen lassen sich formal und kompakt ähnlich wie in modernen Programmiersprachen beschreiben.

**Nachteil:** Da es sich um keine grafische Darstellung handelt, ist die Lesbarkeit eingeschränkt.

→ (siehe Vorlesung, *Digitale Informationsverarbeitung*)

## Zusammenfassung I

Das **ER-Modell** erlaubt die Modellierung von permanent zu speichernden Datenstrukturen und ihren Beziehungen zueinander. **Entitäten (entities)** werden zu **Entitätstypen (entity types)** zusammengefasst. Gleichrangige, fachliche Zusammenhänge zwischen Entitäten werden durch **Beziehungen (Assoziationen, relationships)** beschrieben, die zu **Beziehungstypen (relation types)** zusammengefasst werden. Durch **Rollen** wird angegeben, welche Funktion eine Entität in der Beziehung spielt. Die Eigenschaften von Entitäten und Beziehungen werden durch **Attribute** beschrieben. Jede Entität muss durch einen eindeutigen Schlüssel identifizierbar sein. Der Komplexitätsgrad einer Assoziation wird durch die **Kardinalität** angegeben (Muss- oder Kann-Beziehung, variable oder feste Obergrenze). Eine **rekursive Beziehung** liegt vor, wenn ein Entitätstyp mit sich selbst in Beziehung steht. Das Ergebnis einer ER-Modellierung bezeichnet man als **konzeptionelles Modell**.

Entitätstypen und Beziehungstypen können auf Tabellen abgebildet werden. Die Entitäten werden als Zeilen in die Tabellen eingetragen. Alle Entitäten einer Tabelle bilden eine Entitätsmenge (entity set).

Vereinfacht lässt sich sagen, dass ein Klassendiagramm ohne Operationen und Botschaften einem ER-Modell entspricht. Allerdings müssen in einem ER-Modell Schlüssel-Attribute für jeden Entitätstyp ergänzt werden.



## Zusammenfassung II

Während sich die ER-Konzepte gut dazu eignen, operativ genutzte Datenbestände zu modellieren, werden für Analysen auf Datenbestände **multidimensionale Datenstrukturen** in Form von **Hyperwürfeln** modelliert. Mit **OLAP**-Operationen kann der Benutzer in den Hyperwürfeln navigieren. **Data Mining** weist den Benutzer auf unbekannte Zusammenhänge hin. Die Datenbestände werden aus operativen Datenbestände in **Data Warehouses** importiert. **Data Marts** stellen Ausschnitte aus Data Warehouses für spezielle Themenbereiche zur Verfügung.

Aufgrund der besonderen Anforderungen an Data Warehouses eignet sich die ER-Modellierung nur bedingt für Hyperwürfel. OO-Ansätze befinden sich im Forschungsstadium.

Eine präzise Definition von Datenstrukturen wird durch **Data Dictionary-**Einträge (DD) ermöglicht. DD-Einträge verwenden dabei eine textuelle, modifizierte Backus-Naur-Form zur Definition.

Im Gegensatz zu ER- und OO-Modellen wird mit DD-Einträgen die Feinstruktur eines Attributs beschrieben, nicht aber der Zusammenhang zwischen Entitätstypen und Klassen.

## Literatur

- Balzert H., Lehrbuch der Softwaretechnik, Spektrum Akademischer Verlag, Heidelberg, 1999.
- 
- Chen P., The Entity-Relationship Model – Towards a Unified View of Data, in: ACM Transactions on Database Systems, Vol. 1, No.1, March 1976, S. 9-36
  - Scheer A.-W., Wirtschaftsinformatik – Informationssysteme im Industriebetrieb, 3. Auflage, Springer-Verlag, Berlin, 1990
  - Holthuis J., Der Aufbau von Data Warehouse-Systemen, 2. Auflage, Dt. Universitäts-Verlag, Wiesbaden, 1999

## Überblick LE 9

### LE 8: Datenorientierte Sicht

### LE 9: Algorithmisch und Regelbasierte Sicht

- Kontrollstrukturen
  - Sequenz, Auswahl, Wiederholung und Aufruf
  - Strukturierte Programmierung
- Entscheidungstabellen und Entscheidungsbäume
  - Erstellung einer Entscheidungstabelle
  - Darstellungsformen
  - Entscheidungstabellen-Verbunde
  - Erweiterte Entscheidungstabellen
  - Eintreffer- und Mehrtreffer-Entscheidungstabellen
- Zusammenfassung

## Lernziele

1. Syntax und Semantik linearer Kontrollstrukturen
2. Syntax und Semantik von Entscheidungstabellen
3. Schachtelung linearer Kontrollstrukturen
4. Anwendung gegebener Kontrollstrukturen
5. Erstellen von Kontrollstruktur und Entscheidungstabelle für eine gegebene Problemstellung

<p><b>Konzepte und Sichten</b></p> <p>→</p> <p>häufig verwendet</p> <p>selten verwendet</p>										
Funktionsbaum	Geschäftsprozess (1987)	Datenflussdiagramm (1966)	Data Dictionary (1979)	ER (Entity Relationship) (1976)	Klassendiagramm (1980/90)	Pseudocode	Regeln	Zustandsautomat (1954)	Petri-Netz (1962)	Sequenzdiagramm (1987)
Funktionale Hierarchie	Arbeitsablauf	Informationsfluss	Datenstrukturen	Entitätstypen & Beziehungen	Klassenstrukturen	Kontrollstrukturen	wenn-dann Strukturen	Endlicher Automat	Nebenläufige Strukturen	Interaktionsstrukturen
Funktionale Sicht			Datenorientierte Sicht		Objektorientierte Sicht	Algorithmische Sicht	Regelbasierte Sicht	Zustandsorientierte Sicht		Szenario-basierte Sicht
LE5			LE8		LE6-7	LE9	LE9-10	LE11-12		LE7

## Kontrollstrukturen

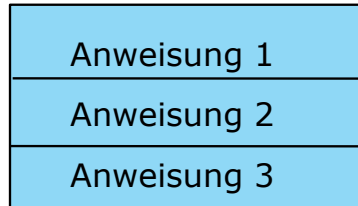
- Kontrollstrukturen dienen dazu, den Ablauf eines Algorithmus zu steuern.
- Gute Kontrollstrukturen
  - ermöglichen problemangepasste und natürlichförmige Beschreibung von Problemlösungen;
  - erlauben die Widerspiegelung der Problemstruktur im Algorithmus;
  - sollten leicht lesbar und verständlich sein;
  - erlauben eine leichte Zuordnung zwischen statischem Algorithmustext und dynamischem Algorithmuszustand;
  - decken mit minimalen, orthogonalen Konzepten ein breites Anwendungsspektrum ab;
  - erleichtern Korrektheitsbeweise von Algorithmen.
- Es gibt vier semantisch unterschiedliche Kontrollstrukturen:
  - Sequenz,
  - Auswahl,
  - Wiederholung und
  - Anwendung anderer Algorithmen; sowie
- drei verschiedene Notationen:
  - Struktogramm-Notation,
  - Pseudo-Code-Notation und
  - Programmablaufplan-Notation.

## Sequenz

## Sequenz

## allgemein

## Erläuterung

Struktur-  
gramm

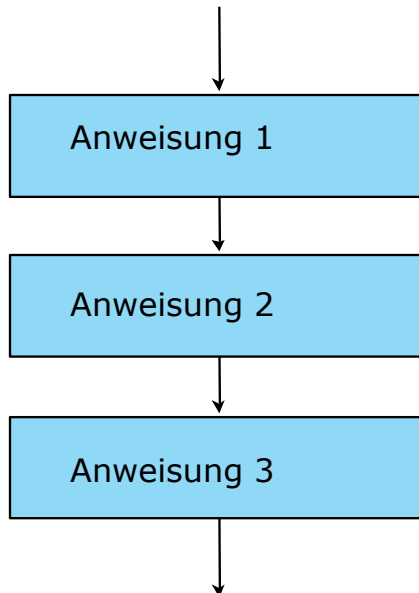
Ein beliebig groß gewähltes Viereck  
Wird nach jeder Anweisung mit  
einer  
horizontalen Linie geschlossen.

Pseudocode  
(Java)

Anweisung1;  
Anweisung2;  
Anweisung3;

Die einzelnen Anweisungen werden  
durch Semikolon voneinander  
getrennt.

PAP

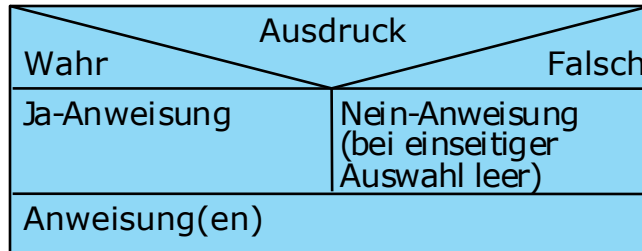


Einfache Anweisungen werden durch  
Rechtecke dargestellt, die wiederum  
durch Ablaflinien verbunden werden.

# Zweiseitige Auswahl

**Auswahl (ein-,zweifach) allgemein Erläuterung**

Strukto-  
gramm



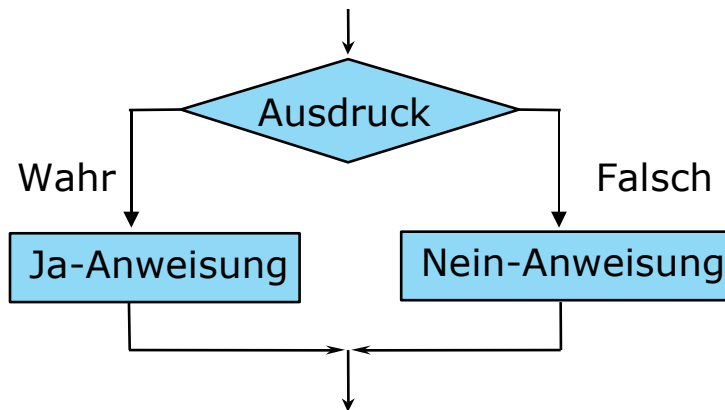
Ist der Ausdruck wahr, dann werden die Ja-Anweisungen, sonst die Nein-Anweisungen ausgeführt.

Pseudocode  
(Java)

```
if (Ausdruck)
    Ja-Anweisung;
else
    Nein-Anweisung;
Anweisung(en);
```

Semantik analog zum Strukto-  
gramm. Bei der einseitigen Aus-  
wahl entfällt „else“-Anweisung.  
Das Ergebnis des Ausdrucks muss  
vom Typ boolean sein.

PAP



Semantik analog zum Strukto-  
gramm. Bei der einseitigen  
Auswahl entfällt der Nein-  
Anweisungszweig.



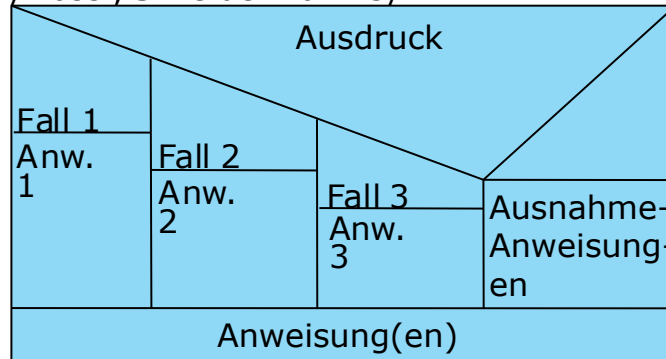
# Mehrfachauswahl

## Mehrfachauswahl allgemein

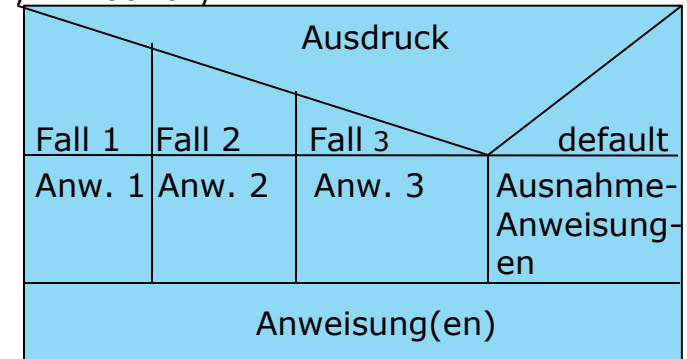
## Erläuterung

Struktogramm

/Nassi, Shneiderman 73/



/DIN 66261/

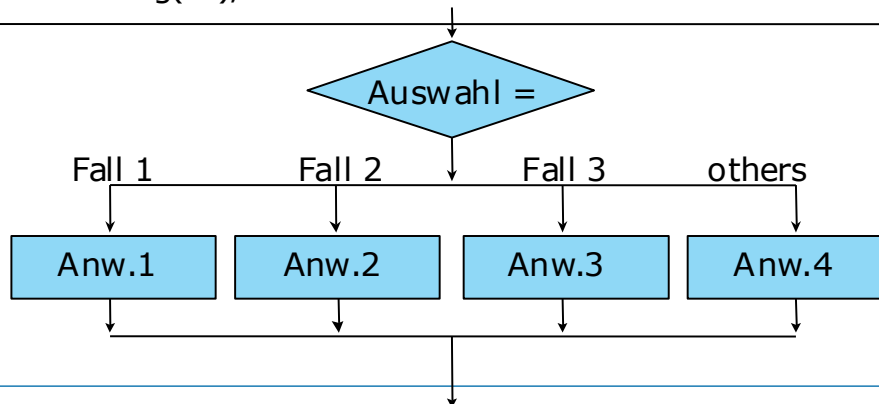


Pseudocode (Java)

```
switch (Ausdruck)
{
  case konstanterAusdruck1:
    Anweisung(en); break;
  case konstanterAusdruck2:
    Anweisung(en); break;
  ...
  default: Anweisung;
}
Anweisung(en);
```

Der Ausdruck dient als Selektor zum Auswählen der einzelnen Fälle. Ist ein entsprechender Fall nicht aufgeführt, wird die Anweisung hinter „default“ ausgeführt.

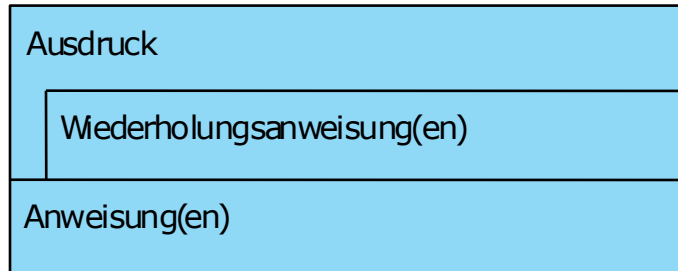
PAP



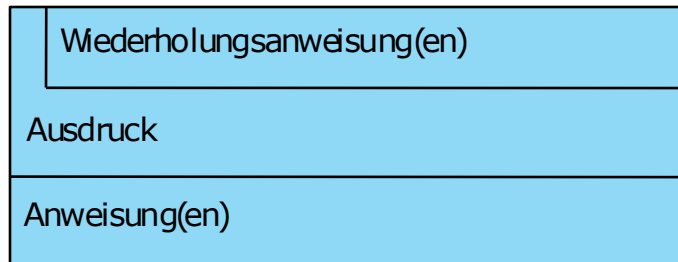
# Wiederholung I

Wiederholung	allgemein	Erläuterung
--------------	-----------	-------------

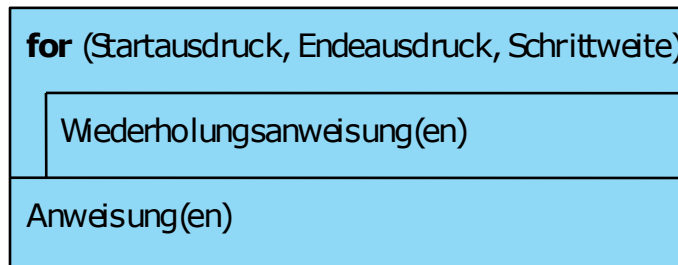
Strukto-  
gramm



Wiederholung mit Abfrage vor jedem Wiederholungsdurchlauf



Wiederholung mit Abfrage nach jedem Wiederholungsdurchlauf

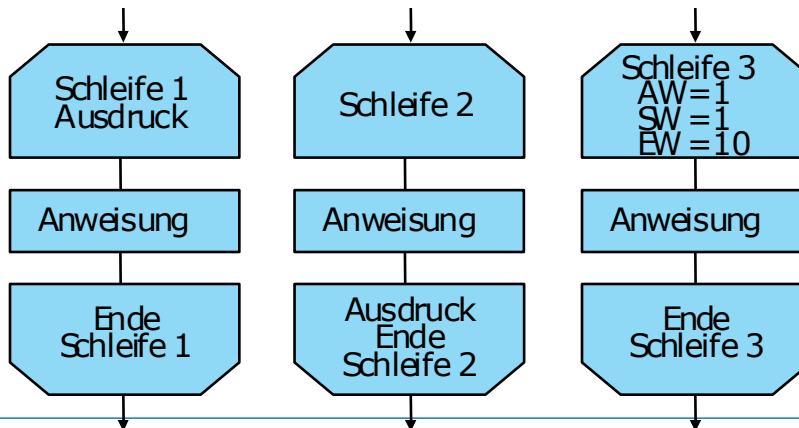


Wiederholung mit fester Wiederholungszahl (Zählschleife, Laufanweisung)

# Wiederholung II

Wiederholung	allgemein	Erläuterung
Pseudocode (Java)	<pre>while (Ausdruck) {   Wiederholungsanweisungen; } Anweisung(en);</pre>	Wiederholung mit Abfrage vor jedem Wiederholungsdurchlauf
	<pre>do {   Wiederholungsanweisungen; } while (Ausdruck); Anweisung(en);</pre>	Wiederholung mit Abfrage nach jedem Wiederholungsdurchlauf
	<pre>for (Startausdruck, Endausdruck, Schrittweite) {   Wiederholungsanweisungen; } Anweisung(en);</pre>	Wiederholung mit fester Wiederholungszahl (Zählschleife, Laufanweisung)

PAP



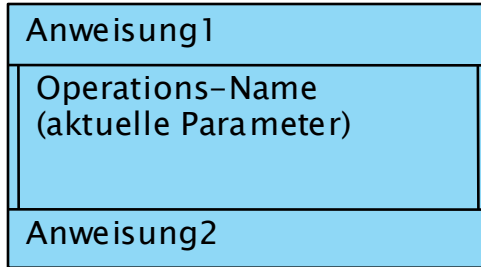
analog  
Struktogramm

AW = Anfangswert  
SW = Schrittweite  
EW = Endwert

# Aufruf

Aufruf	allgemein	Erläuterung
--------	-----------	-------------

Strukto-  
gramm



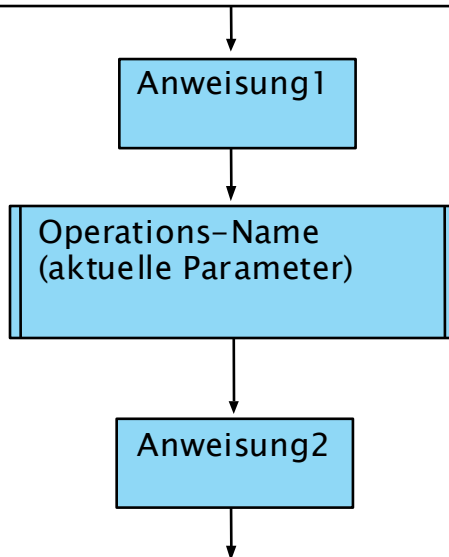
Nach Ausführung der aufgerufenen Operation wird die rufende Operation hinter der Aufrufstelle fortgesetzt.

Pseudocode  
(Java)

```
Anweisung1;
Operationsname (aktuelleParameter);
Anweisung2;
```

Ein Aufruf erfolgt durch Angabe des Operationsnamens, gefolgt von der Liste der aktuellen Parameter.

PAP



# Eigenschaften von Kontrollstrukturen

## Gemeinsames Kennzeichen der vorgestellten Kontrollstrukturen

- Besitzen genau einen Ein- und einen Ausgang
- Zwischen dem Eingang und dem Ausgang gilt das Lokalisierungsprinzip
  - Der Kontrollfluss verlässt den durch Eingang und Ausgang definierten Kontrollflussbereich nicht
  - Makroskopisch betrachtet verläuft der Kontrollfluss linear durch einen Algorithmus, daher: „lineare Kontrollstrukturen“

## Strukturiertes Programmieren im engeren Sinne

- Verwendung ausschließlich linearer Kontrollstrukturen
- Dijkstra subsumierte unter diesem Begriff verschiedene methodische Ansätze, die die Programmzuverlässigkeit verbessern
- Weiterentwicklung dieser Ansätze führte zu oft weit auseinander-liegenden Definitionen des Begriffs „Strukturierte Programmierung“.

## Sprung („goto-Anweisung“)

- Verstößt gegen die aufgestellten Anforderungen
- Verwischt den semantischen Unterschied zwischen einer Auswahl und einer Wiederholung
- $n + 1/2$ -Schleife
  - Innerhalb des Wiederholungsteils können ein oder mehrere Unterbrechungen oder Aussprünge programmiert werden.

# Eigenschaften von Kontrollstrukturen

## Struktogramme

- Optimale grafische Darstellung von linearen Kontrollstrukturen
- Der verfügbare Platz ermöglicht die Wahl aussagekräftiger Namen
- Sprünge lassen sich nicht darstellen

## Programmablaufplan

- Für grundlegende Kontrollstrukturen gibt es keine eigenen Symbole (Mehrfachauswahl, Wiederholungen)
- Vorgenommene Erweiterungen entsprechen nicht der ursprünglichen Intention
- Insgesamt bieten PAPA dem Benutzer zu große Freiheiten

## Vorteile von Lineare Kontrollstrukturen

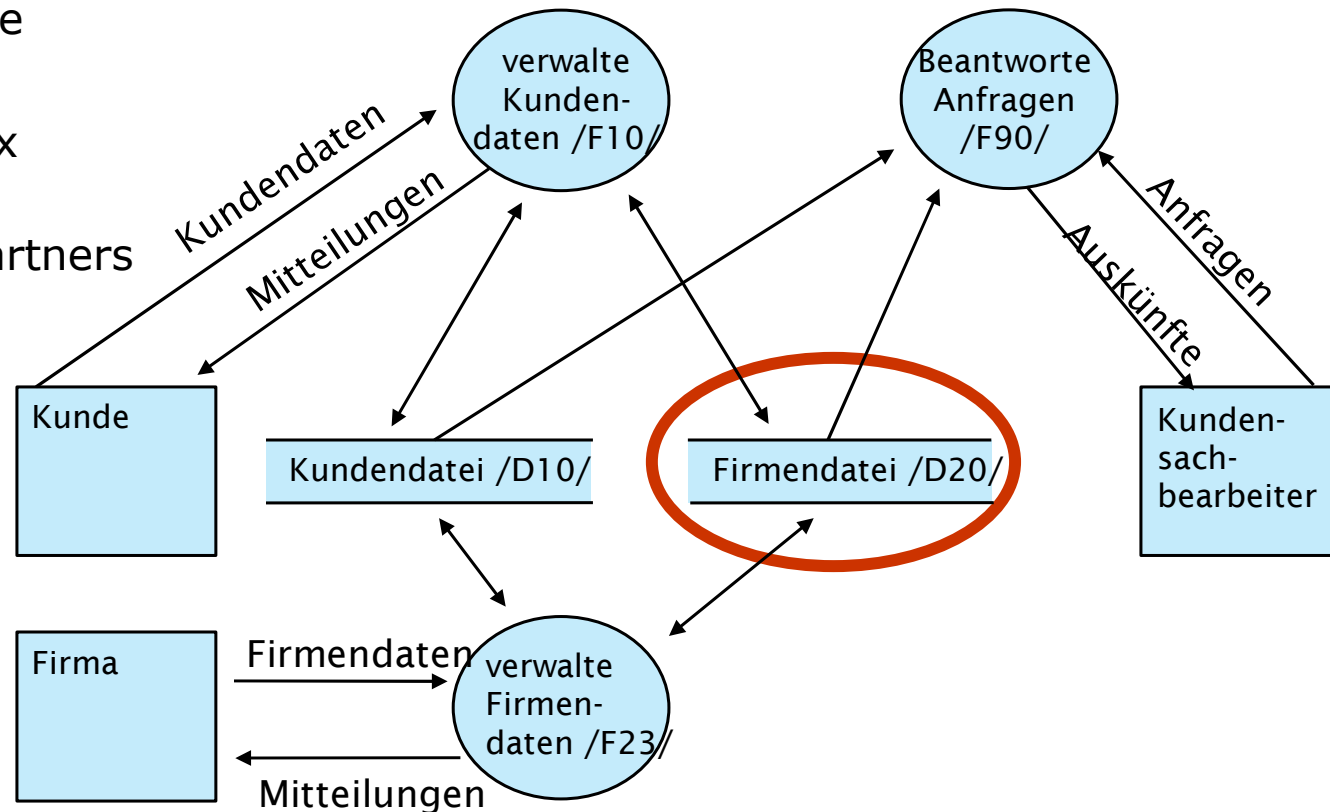
- Vereinheitlichung der Programmierstile, d.h. Standardisierung der Kontrollflüsse
- Übersichtliche, gut lesbare Anweisungsteile
- Leichte Überprüfbarkeit der Termination
- Für gleichartige Probleme entstehen gleichartige Kontrollfluss-Strukturen
- Statische Überprüfung der Korrektheit möglich
- Auswirkungen jeder Kontrollstruktur übersehbar

## Methodik

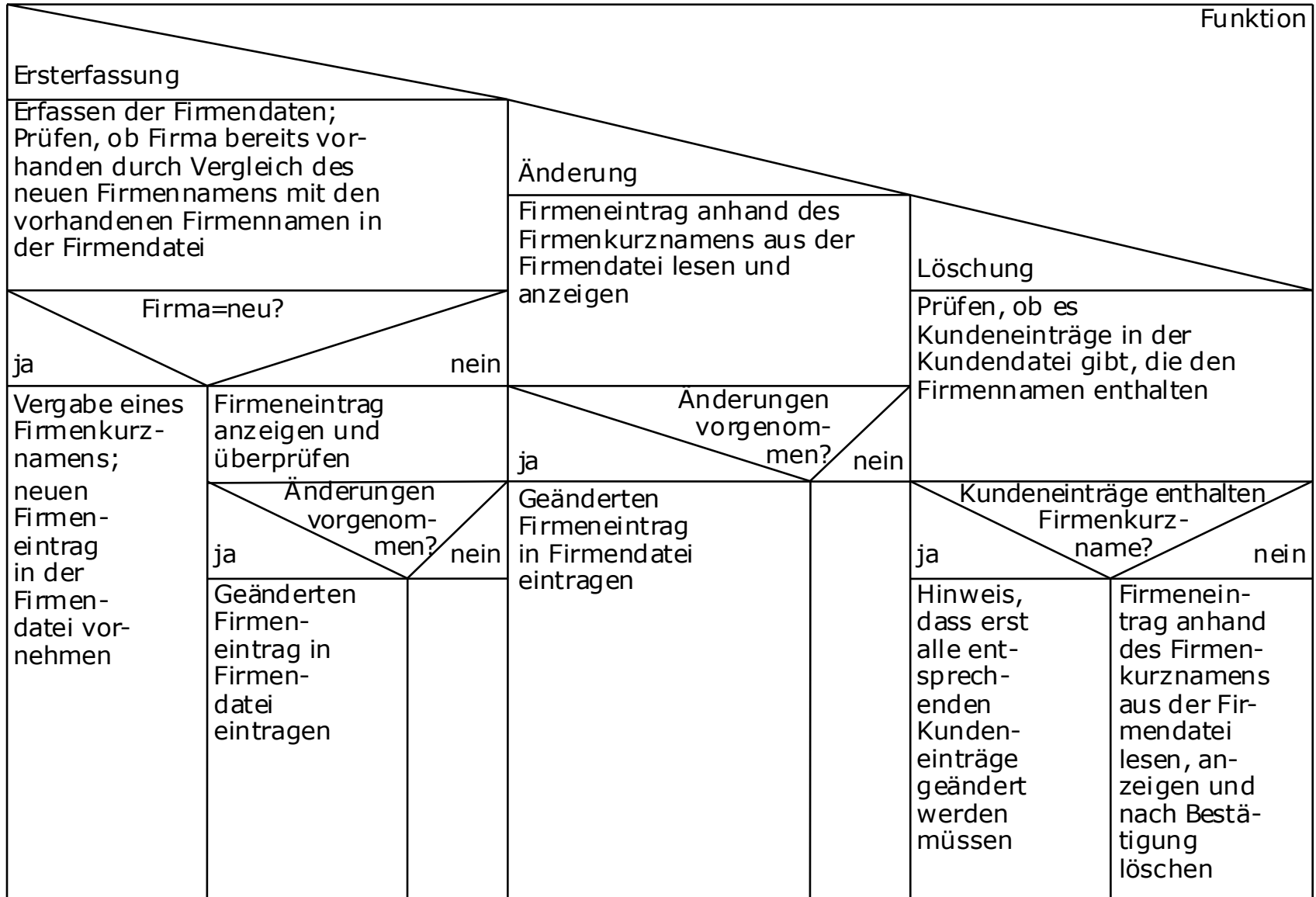
- Folgende Reihenfolge einhalten:
  1. Immer zuerst die Kontrollstrukturen entwerfen
  2. Erst dann die elementaren Anweisungen überlegen.

## Beispiel für ein Struktogramm

- Fallstudie Seminarorganisation:
  - /D20/: Gehört ein Kunde zu einer Firma, dann sind über die Firma folgende Daten zu speichern:
    - Firmenkurzname
    - Firmenname
    - Adresse
    - Telefon, Fax
    - Name des Ansprechpartners
    - Abteilung
    - Telefon
    - Umsatz.



# Beispiel für ein Struktogramm





## Entscheidungstabellen

- **Entscheidungstabellen (ET):** Vorzunehmende Aktionen oder Handlungen, die von der Erfüllung oder Nichterfüllung mehrerer Bedingungen abhängen, können kompakt und übersichtlich definiert werden.

- **Struktur einer Entscheidungstabelle**

wenn

dann

Name der ET	Regelnummern
Bedingungen	Bedingungsanzeiger
Aktionen	Aktionsanzeiger

- **5 Schritte**

1. Ermittlung der Aktionen
2. Ermittlung der Bedingungen
3. Eintrag der Bedingungen und Aktionen in die ET
4. Eintrag aller Bedingungskombinationen
5. Eintrag der Aktionsanzeiger.

## Beispiel für eine Entscheidungstabelle 1

- Ein Sachbearbeiter in einer Bank soll bei der Einlösung von Schecks folgende Regeln beachten:
  - /1/ Wenn die vereinbarte Kreditgrenze des Ausstellers eines Schecks überschritten wird, das bisherige Zahlungsverhalten aber einwandfrei war und der Überschreibungsbetrag kleiner als 500,- Euro ist, dann soll der Scheck eingelöst werden.
  - /2/ Wenn die Kreditgrenze überschritten wird, das bisherige Zahlungsverhalten einwandfrei war, aber der Überschreibungsbetrag über 500,- Euro liegt, dann soll der Scheck eingelöst und dem Kunden neue Konditionen vorgelegt werden.
  - /3/ War das Zahlungsverhalten nicht einwandfrei, wird der Scheck nicht eingelöst.
  - /4/ Der Scheck wird eingelöst, wenn der Kreditbetrag nicht überschritten ist.
- Folgende Aktionen sind möglich:
  - A1: Scheck einlösen /1/, /2/, /4/
  - A2: Scheck nicht einlösen /3/
  - A3: Neue Konditionen vorlegen /2/
- Folgende Bedingungen treten auf:
  - B1: Kreditgrenze überschritten? /1/, /2/, /4/
  - B2: Zahlungsverhalten einwandfrei? /1/, /3/
  - B3: Überschreibungsbetrag < 500,- Euro? /2/.

**Beispiel für eine Entscheidungstabelle 2**

- ET für die Scheckeinlösung
  - Aktionsteil wurde noch um eine zusätzliche »unlogisch« - Aktion erweitert

	ET1: Scheckeinlösung	R1	R2	R3	R4	R5	R6	R7	R8
B1	Kreditgrenze überschritten ?	J	J	J	J	N	N	N	N
B2	Zahlungsverhalten einwandfrei ?	J	J	N	N	J	J	N	N
B3	Überschreibungsbetrag < 500,- Euro	J	N	J	N	J	N	J	N
A1	Scheck einlösen	X	X			X		X	
A2	Scheck nicht einlösen			X	X				
A3	neue Konditionen vorlegen		X						
A4	unlogisch						X		X

ET1: Scheckeinlösung (optimiert)		R2	R3/4	Else
B1	Kreditgrenze überschritten ?	J	J	
B2	Zahlungsverhalten einwandfrei ?	J	N	
B3	Überschreibungsbetrag < 500,- Euro	N	-	
A1	Scheck einlösen	X		X
A2	Scheck nicht einlösen		X	
A3	neue Konditionen vorlegen	X		

**Alternative Darstellungsformen 1**

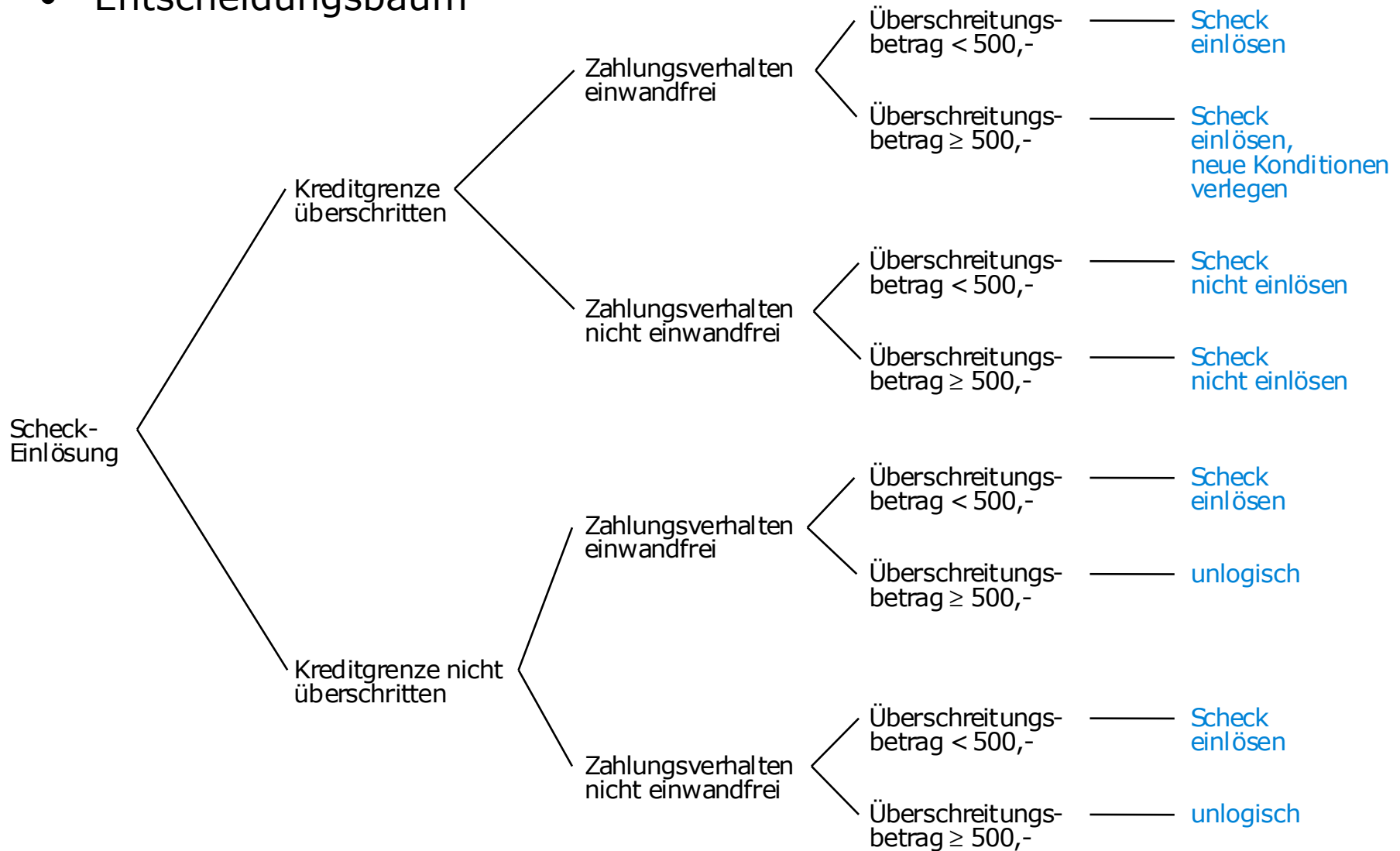
- Horizontale Anordnung der Regeln

Kreditgrenze überschritten	Zahlungsverhalten einwandfrei	Überschreitungsbeitrag < 500,- ?	Scheck einlösen	Scheck nicht einlösen	neue Konditionen vorlegen	unlogisch
J	J	J	X			
		N	X		X	
	N	J		X		
		N		X		
N	J	J	X			
		N				X
	N	J	X			
		N				X

Kreditgrenze überschritten	Zahlungsverhalten einwandfrei	Überschreitungsbeitrag < 500,- ?	Scheck einlösen	Scheck nicht einlösen	neue Konditionen vorlegen
J	J	N	X		X
	N	-		X	
Else			X		

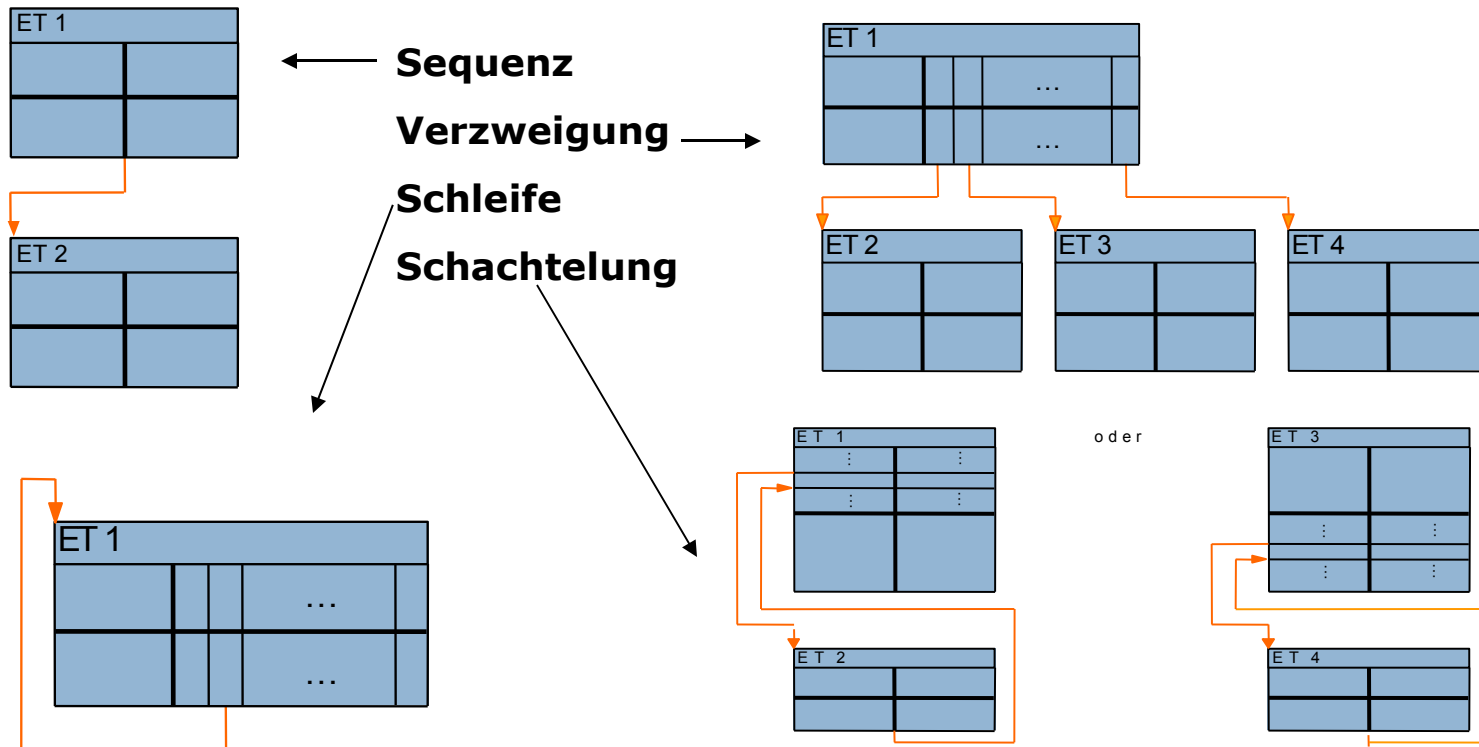
# Alternative Darstellungsformen 2

- Entscheidungsbaum



## Entscheidungstabellen-Verbunde

- Bei mehr als 5 Bedingungen können vollständige ET nicht mehr aufgestellt werden.
- Mögliche Verknüpfungsformen sind Sequenz, Verzweigung, Schleife und Schachtelung.



## Erweiterte Entscheidungstabellen

- Weitere Möglichkeit Entscheidungstabellen mit vielen Bedingungen kompakt darzustellen, besteht darin erweiterte Bedingungen und Aktionszeiger zu verwenden
- Anstelle von J,N,- und X kann ein beliebiger Text im Anzeigerteil verwendet werden.

ET: SoftWAir	R1	R2	R3	R4	R5	R6
Alter	>=18	>=18	>=18	>=18	>=2 u.<18	<2
Flugziel	Miami	N.Y.	Miami	Miami	-	-
Aufenthalt >=6 Tage	J	-	N	-	-	-
Abflug zw. 21. u. 30.12.	N	-	-	J	-	-
Preisnachlass in %	20	0	0	0	30	100

- Durch Anwenden der Else-Regel lassen sich Tabellen weiter optimieren

ET: SoftWAir	R1	R5	R6	ELSE
Alter	>=18	>=2 u. <18	<2	
Flugziel	Miami	-	-	
Aufenthalt >=6 Tage	J	-	-	
Abflug zw. 21. u. 31.12.	N	-	-	
Preisnachlass in %	20	30	100	0

# Mehrtreffer-Entscheidungstabellen

## Mehrtreffertabellen

- Nicht-disjunkte Bedingungsanzeigerteile sind erlaubt.
- Zwischen den Regeln können vier verschiedene Beziehungstypen bestehen:
  1. Gleichheit (Identität)
    - Sind die Bedingungsanzeiger zweier Regeln gleich, dann sind die Regeln in Abhängigkeit von den Aktionsanzeigern entweder redundant, widersprüchlich oder zusammenfassbar
  2. Ausschluss (Exklusion)
    - Ein gegenseitiger Ausschluss zweier Regeln liegt vor, wenn sie in mindestens einer Bedingungszeile unterschiedliche Bedingungsanzeiger haben
  3. Einschluss (Inklusion)
    - Liegt vor, wenn mindestens ein Bedingungsanzeiger B1 von R1 auch in R2 enthalten ist und alle übrigen Bedingungen gleich sind
  4. Überschneidung (Intersektion).
    - mindestens ein Bedingungsanzeiger B1 von R1 auch in R2 und ein weiterer Bedingungsanzeiger B2 von R2 auch in R1 enthalten ist und alle übrigen Bedingungsanzeiger gleich sind



## Vorteile von Entscheidungstabellen

- Erlauben eine übersichtliche und kompakte Darstellung von Aktionen, die von mehreren Bedingungen abhängig sind.
- Entscheidungstabellen können auf verschiedene Eigenschaften hin überprüft und optimiert werden.
- Werkzeuge unterstützen die Erstellung, Analyse und Optimierung von Entscheidungstabellen.
- Implementierungsphase:
  - Werkzeuge generieren den fertigen Quellcode.
- Definitionsphase:
  - Abarbeitungsreihenfolge wird nicht festgelegt.
  - Freiheitsspielräume bleiben erhalten, die bei Kontrollstrukturen nicht mehr vorhanden sind.

## Zusammenfassung

Kontrollstrukturen legen innerhalb eines Algorithmus fest, in welcher Reihenfolge, ob und wie oft Anweisungen ausgeführt werden sollen. Die **strukturierte Programmierung** erlaubt nur **lineare Kontrollstrukturen**. Es lassen sich vier verschiedene Typen unterscheiden: die **Sequenz**, die **Auswahl**, die **Wiederholung** und der **Aufruf**. Alle Typen lassen sich beliebig miteinander kombinieren und ineinander schachteln.

Eine textuelle Darstellungsform (**Pseudo-Code**) und zwei grafische (**Struktogramme** und **PAP**).

**Entscheidungstabellen** erlauben eine tabellarische oder grafische Darstellung (**Entscheidungsbäume**) von durchzuführenden Aktionen in Abhängigkeit von Bedingungen.

Tritt zu einer Zeit genau eine Bedingungskonstellation auf, dann handelt es sich um eine **Eintreffer-Tabelle**.

**Begrenzte Entscheidungstabelle** liegt vor, wenn als Bedingungsanzeiger nur J,N und als Aktionszeiger nur X verwendet werden.

Erfordert eine Problembeschreibung eine Kombination von Entscheidungstabellen (möglich sind Sequenz, Verzweigung, Schleife und Schachtelung), dann geschieht dies durch einen **Entscheidungstabellen-Verbund**.

## Literatur

- Balzert H., Lehrbuch der Softwaretechnik, 2. Auflage, Spektrum Akademischer Verlag, Heidelberg, 2000.
- 
- Strunz H., Entscheidungstabellentechnik, Carl Hanser Verlag, München, 1977
  - Programmkonstrukte und Regeln für ihre Anwendung, Beuth-Verlag, Berlin, November 1994
  - Sinnbilder und ihre Anwendung, Beuth-Verlag, Berlin, 1983