

# Zusammenfassung OODB Kapitel 6 + 7

## 6. Einführung in Begriff und Konzepte objektorientierter DBS

### Darstellung mittels Relationenmodell (Nachteile):

- relational Datenmodell ist unhandlich und schwerfällig
- Zusammengesetzte Attribute müssen zerschlagen werden.
- Mengenattribute verstoßen gegen 1NF
- Spezialisierung
- Künstliche Schlüssel müssen eingeführt werden.
- Aufwendige Formulierungen für Abfragen

→ diese Nachteile sind gleichzeitig die Ziele eines OO\_DBS

### Ziele:

- Struktur der Entity-Sets beliebig verschachtelbar
- Entity-Sets können mit spezifischem Verhalten versehen werden
- Struktur und Verhalten können vererbt werden.
- Einzelne Individuen nicht nur über Schlüsselattribute unterscheidbar

### Objekte: besitzen Daten (Attributwerte) und Verhalten (Methoden).

#### Manipulationen nur über

Methoden. Objekte identische Struktur und gleichem Verhalten bilden Klassen. Verfügen über eine Objektidentifikation.

### Vererbung:

Verfügbarkeit von Attributen und Methodeneiner übergeordneten Klasse in einer untergeordneten Klasse.

Vererbungsprinzip: gewährleistet Wiederverwendbarkeit von Datenstrukturen (Attribute) und Verhaltenseigenschaften (Methoden)

Einfache Vererbung:

- o eindeutiger Weg von einer untergeordneten Klassen zu übergeordneter Klasse.
- o Ist die Hierarchie als Baum darzustellen spricht man von einer Einfachvererbung.

Mehrfachvererbung: zu einer untergeordneten Klasse gehören mehrere übergeordnete Klassen

Generalisierung bottom up

Spezialisierung down to bottom

### Vererbungsstrukturen:

1. Disjunkte und vollständige Überdeckung einer Menge durch ihre Spezialisierungsmengen.
2. gegenseitig disjunkte Überdeckung, unvollständig durch...
3. Überlapende und vollständige Überdeckung durch...
4. Überlappende und unvollständige Mengen durch..

Überlappende Spezialisierungen: Schnittmengen als eigene Untermenge

Unvollständige Überdeckung: eigenständige Klassen für die außer acht gelassenen

Mengen

# Zusammenfassung OODB Kapitel 6 + 7

## Überschreiben:

Einer Nachricht kann in einer Unterklasse eine andere Methode zugeordnet werden, als es in einer oberen Klasse der Fall ist.

## Überladen:

Prozedurnamen können in unterschiedlichen Objekttypdefinitionen andere Implementierungen zugeordnet werden (Polymorphismus).

→ Ein Name ist nur innerhalb einer Klasse eindeutig.

→ Ererbte Eigenschaften und Methoden können redefiniert werden (Overriding).

## Statisches binden

## Dynamisches Binden:

Operationsaufruf wird erst zur Laufzeit eine Implementierung zugeordnet (Auswahlmöglichkeit).

## Prinzipien der Objektorientierung:

1. Komplexe Objekte
  2. Objektidentität
  3. Datenkapselung
  4. Typen und Klassen
  5. Vererbung
  6. Polymorphismus
  7. Vollständigkeit
  8. Erweiterbarkeit
- 
1. Komplexe Objekte können selbst Objekte als Attribute haben.  
Zugelassen sind elementare Datentypen, Mengen, Listen oder ganze Objekte.  
Objekterstellung mittels Konstruktoren [tupel (Tupelbindung), set (Mengenbildung), bag (Multimenge), list (geordnete Liste)].
  2. Jedes Objekt erhält systemunterstützten Identifikator (object identifier, OID)  
Zwei Objekte sind gleich, wenn sie gleiche Werte haben  
Zwei Objekte sind identisch, wenn der Identifikator übereinstimmt.
  3. Bedeutet, nach außen hin werden Implementierungsdetails verborgen. Nur über Nachrichten → Methode ansprechbar.
  4. Müssen unterstützt werden. Durch Typen wird Klassenstruktur beschrieben.  
Typen sind: Basistypen, eine Klasse oder ein zusammengesetzter Typ.  
Klassenkonzept ist dynamischer
  5. Klassenhierarchien drücken Spezialisierungen aus.  
Eine Unterklasse erbt Eigenschaften einer übergeordneten Klasse und kann eigene hinzufügen: `Leiter inherits Angestellte:[KFZ_ID: number(6)]`
  6. Dieselben Methodennamen können bei Objekten unterschiedlicher Klassen vielgestaltig (polymorph) angewendet werden durch eine abweichende Implementierung. (unterscheide statisches und spätes binden!)  
Spätes binden auch bei Überladen und Überschreiben möglich.
  7. Eine berechnungsvollständige Datenbankprogrammiersprache wird die Manipulation von Datenbankobjekten und die Implementierung von Methoden unterstützen.
  8. Die Menge der vordefinierten Datentypen, Klassen und Operationen muss durch den Benutzer erweiterbar sein.

# Zusammenfassung OODB Kapitel 6 + 7

## DB-Eigenschaften:

1. Dauerhaftigkeit
  2. Rekonstruierbarkeit
  3. Mehrbenutzerbetrieb
  4. Große Datenbestände
  5. Ad-hoc-Abfragemöglichkeiten
- 
1. Persistenz eines Datenbankzustandes muss über längeren Zeitabstand gesichert sein (über Sekundärspeicher). Evtl. transiente Objekte gewünscht (Lebensdauer auf Programmlaufzeit beschränkt).
  2. Fehlerbehandlung muss unvollständig ausgeführte Transaktionen zurücksetzen und noch einmal starten können (Recovery).
  3. leistungsfähiges Transaktionskonzept. Mehrere Benutzer müssen auf der Datenbank arbeiten können.
  4. Sekundärspeicherverwaltung. Verwaltungskomponenten für Sekundärspeicher müssen bereitgestellt werden, die die komplexen Objekte berücksichtigt.
  5. Deklarative Query-Sprache müssen deklarative Sprachkonstrukte zur Verfügung stellen, um Datenbank ad hoc auswerten zu können (zeichnen sich durch ihre Mächtigkeit aus).

**Umstritten:** Views, Integritätsbedingungen, Schemaevolution, DBA-Utilities.

**Zu einem OO-DBS gehören diese 8 + 5 oben genannten Eigenschaften.**

## Optional Bestandteile einer OO-Datenbank:

- Mehrfachvererbung
- Statische Typisierung und Typ-Interferenz
- Verteilung
- Entwurfstransaktionen
- Versionen
- Integrität
- Sichten
- Schemaevolution
- Zugriffskontrolle

## OODBMS ist System, welches:

- auf einem objektorientierten Datenmodell basiert
- eine objektorientierte Datenbankprogrammierungsumgebung anbietet
- (zumindest konzeptuell) erweiterbar ist
- mindestens die 5 Datenbankfunktionalität.

# Zusammenfassung OODB Kapitel 6 + 7

## Ebenen des objektorientierten Anwendungssystems und deren Objektorientierung:

1. Graphische Benutzerschnittstelle
  - konventionelle Programmierung
  - an Benutzerschnittstelle werden objektorientierte Konzepte verwendet.
2. Objektorientierte Programmiersprache
  - es wird objektorientiert programmiert
  - die Datenverwaltung wird mit relationaler Technologie realisiert
3. Objektorientiertes Datenbanksystem.
  - durchgängig objektorientierter Ansatz
  - Wiederverwendung von Daten und Verhalten
  - Unterstützung komplex strukturierter Datenbestände
  - Keine Kluft zwischen Programmiersprachen und Datenbanken

## 7. Modellierung objektorientierter Datenbanken

**Zielsetzung:** Analyse und Entwurf einer objektorientierten Datenbank

**Analyse:** nachvollziehbare und widerspruchsfreie Beschreibung des Problembereichs

**Entwurf:** nachvollziehbare und widerspruchsfreie Beschreibung des Problembereichs

Hierfür zur Verfügung stehende Beschreibungselemente:

1. Objekte
  2. Klassen mit Attributen und Methoden
  3. Beziehungen zwischen Klassen
  4. Vererbung
  5. Nachrichten
- 
1. Objekte wie oben:
    - individuelle Ausprägung von ganzheitlichen Dingen oder Konzepten des Problembereichs
    - durch Attribute gekennzeichnet
    - verfügt über Objektidentifikation
    - enthält die zugehörigen Methoden.
  2. Alle Objekte einer Klasse besitzen gleiche Attribute und Methoden
    - Intension: Beschreibung der Attribute und Methoden
    - Extension: Menge der Ausprägungen einer Klasse
  3. Beziehungen
    - Assoziationen (Generalisierung)
    - Zusammensetzung (Aggregation) von Objekten

# Zusammenfassung OODB Kapitel 6 + 7

## Nachrichten:

- ausgetauscht zwischen den einzelnen Objekten
- damit wird das dynamische Verhalten zwischen einzelnen Objekten verschiedener Klassen modelliert

## Methode:

Klar definierte Dienstfunktion, die Werteingaben, Wertänderungen und Wertselektionen der Objekte regelt.

## Objektstruktur:

- Objekt besitzt eine innere Struktur in Form von Attributen
- Aktuelle Belegung dieser Attribute macht den Objektzustand aus.

## Attributwerte:

- atomaren Werte (Zahlen, Zeichenketten)
- Referenzen auf andere Objekte

## Objektgleichheit:

Ausgangskonzepte:

- die Identifikatoren der Objekte
- die Werte der Objektattribute

*Wertegleichheit:*

Wie in Progsprachen, Notation  $w_1 = w_2$

*Objektgleichheit:*

- wenn gleichen Objektidentifikator  $\rightarrow$  gleiches Objekt  $w_1 = w_2$

Objektgleichheit erster Stufe:  $o_1 =_1 o_2$

- ihre Attribute übereinstimmen (gleiche innere Strukturierung)
- für deren Werte bzw. Referenzen Werte- bzw. Objektgleichheit gilt (gleiche Attributbelegung)

## Surrogate:

- eindeutige Identifikationsschlüssel für Objekte
  - unveränderbar
  - werden unabhängig von Objekteigenschaften systemweit vergeben
  - unabhängig gegenüber Änderungen in der Objektstruktur
  - unabhängig vom geographischen Speicherort und von der Speicherungsform vergeben.
- $\rightarrow$  Eindeutigkeit, Unveränderbarkeit, Strukturunabhängig und Ortsunabhängig

## Eigenschaften von Klassen:

- Klasse beschreibt eine Menge von Objekten mit denselben Attributen und Methoden.
- Klassenattribute werden nur einmal pro Klasse angelegt
- Klassenmethoden können definiert werden.
- Verfügt über vordefinierte Attribute und Methoden (Attribut: Objektidentifikation; Methoden: erzeugen und löschen)
- Metaklassen: Objekte selbst sind Klassen

# Zusammenfassung OODB Kapitel 6 + 7

## Beziehungskonzepte

### Abstraktionskonzepte:

1. Klassifikation
  2. Aggregation (Zusammensetzung)
  3. Verallgemeinerung bzw. Spezialisierung
- 
1. Klassifikation:
    - mathematisch Mengenbildung
    - dient der Definition bestimmter Konzepte als Klassen von Objekten mit gemeinsamen Eigenschaften
  2. Aggregation (Zusammensetzung)
    - kartesisches Produkt
    - definiert eine neue Klasse aus bereits existierenden Klassen (die dann zu Komponenten werden)
  3. Verallgemeinerung und Spezialisierung
    - mathematisch Potenzmengenkonstruktionen bzw. disjunkte Zerlegung einer Klasse in Unterklassen (Teilmengen)
    - besitzen Vererbungseigenschaften: alle Attribute der allgemeineren Klasse werden an jede Teilmenge vererbt

Verallgemeinerung(Generalisierung): definiert Teilmengenbeziehung von Klassen (von unten nach oben)

Spezialisierung definiert Teilmengenbeziehung von Klassen (von oben nach unten)

### Assoziationen:

- modellmäßige Beziehungen zwischen Klassen mit zwei Assoziationen:
  - die Bedeutung
  - die Mächtigkeit
- Jede Assoziation hat eine inverse Assoziation
- Assoziationstypen: einfach (1), konditionelle(0..1), mehrfache(1..\*) und konditionell mehrfach (0..\*)

### Beziehungsklassen:

- Klasse die von mehr als einer Klasse abhängig ist
- Hat eigene Attribute, die diese Beziehung näher beschreiben
- Kann Methoden haben, die Zugriffsfunktionen für die Objekte der Beziehungsklasse darstellen
- Wird durch gestrichelt Linie an der entsprechenden Beziehung angehängt

### Aggregation:

- ermöglicht das Kombinieren von Teilobjekten zu zusammengesetzten Objekten.
- Aggregationsbeziehungen bauen auf Assoziationen auf, um eine Komponentenstruktur auszudrücken.
- Die Struktur wird durch Assziationen von Komponenteklassen zur übergeordneten Aggregationsklasse visualisiert. (Bsp. Vergaser, Zylinder, Kolben → Motor)

# Zusammenfassung OODB Kapitel 6 + 7

Vererbung siehe oben

## Dynamische Verhalten:

### **Nachrichten:**

Aufforderung eines Objekts an ein anderes, eine bestimmte Methode auszuführen. Sämtliche Interaktionen laufen über diese Interaktion.

### **Protokoll:**

Menge aller öffentlichen Meldungsnachrichten, die einer Klasse zur Verfügung stehen

Instanzenmethoden: Objektorientiert. Z.B. Attribut eines Objektes verändern(LKW-Nr.).

Klassenmethode: Klassenorientiert. Z.B. Anzahl der LKW verändern

### **Dynamisches Verhalten ist wichtig:**

- Reihenfolge der einzelnen Interaktionen (Nachrichten)
- Das zeitliche Verhalten unter Berücksichtigung von Wartezeiten und Ausführungszeiten.

### **Zustandsübergangsdiagramme:**

- Zeigen Ablauffolge der Zustände vom Start bis zum Endzustand eines Systems
- Lösen Zustandsänderung durch Ereignisse (oder Nachrichten) aus
- Auf unterschiedlichen Abstraktionsebenen einsetzbar
- Als Graphen darstellbar
- Knoten als Zustand, gerichtete Kanten als Zustandsänderungen