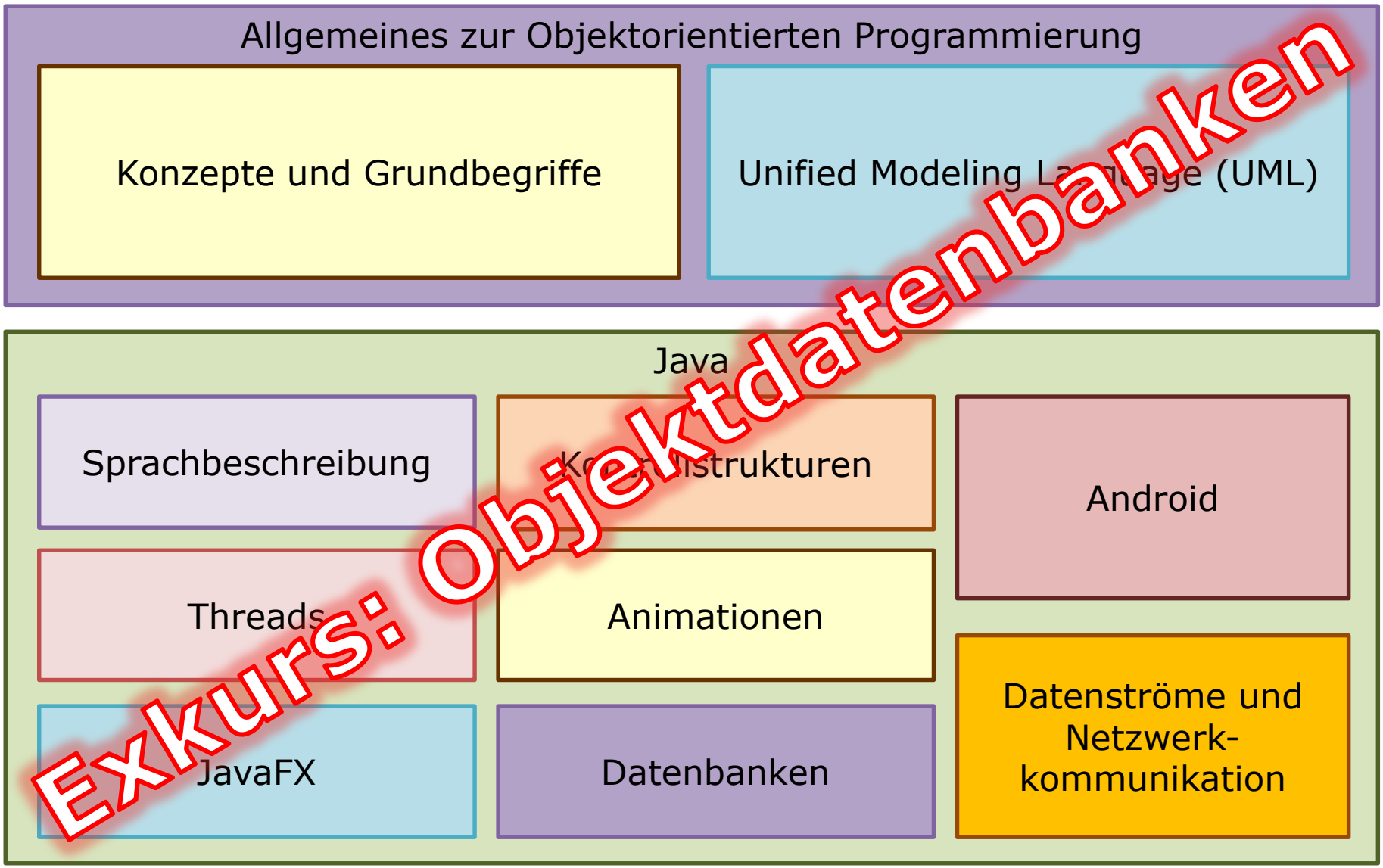
BERUFSAKADEMIE SACHSEN
STAATLICHE STUDIENAKADEMIE
DRESDEN
UNIVERSITY OF COOPERATIVE EDUCATION

Objektorientierte Programmierung

Objektdatenbanken

Prof. Dr.-Ing. Tenshi Hara
tenshi.hara@ba-sachsen.de

Aufbau der Lehrveranstaltung



Definition

Eine Objektdatenbank speichert Daten als Objekte im Sinne der Objektorientierung (anders als z.B. relationale Datenbanken).

Objektdatenbank und Objektdatenbank-Management-System bilden gemeinsam das Objektdatenbanksystem.

Jedes Objekt modelliert eine Dateninstanz und enthält in der gleichen Struktur ihre Attribute und die zum Zugriff notwendigen Methoden.

Anforderungen

Neben den Anforderungen klassischer DBMS

- komplexe Objekte: Objekte setzen sich aus beliebigen anderen Datentypen zusammen
- Sicherstellung der Objektidentität: systemweit eindeutige OID
- volle Kapselung: Zugriff auf das Objekt findet nur über Methoden statt
- Objektklassen unterliegen einer Klassenhierarchie
- späte Bindung: bei vererbten Objekten werden immer zuerst die überladenen Methoden verwendet
- Turing-vollständige Datenmanipulationssprache (DML)

Funktionsweise

1. Datensatz wird mit Attributen und Methoden in einem Objekt zusammengefasst

2. Objekte werden wiederum in abstrakten Klassen zusammengefasst
   a) jede Klasse wird aus verschiedenen Einheiten an Objekten gebildet
   b) innerhalb eines Klassenkonstrukts wird geordnet
      → Hierarchie: Vererbung von Eigenschaften, sowie Objektvernetzung

3. jede Objekteinheit erhält automatisch ihre einmalige OID
   a) einfache Referenz
   b) einfacher Abruf

Vorteile

- überbrücken logische Konversionslücke zwischen objektorientiertem Programm und (relationaler) Datenhaltung (Object-Relational Impedance Mismatch)
  - direkte Kommunikation von Programm und Datenbank
  - objektrelationale Abbildungen werden nicht benötigt
- keine Joins notwendig → Objekte werden über Beziehungen abgefragt
- semantische Zusammenhänge zwischen Objekten und Datenbanksystem sind bekannt: Ergebnis einer Anfrage ist nicht Menge von Datensätzen, sondern einzelnes Objekte.
- keine Surrogate Keys (künstlich erzeugte Primär- oder Sekundärschlüssel) nötig
  - jedes Objekt hat seine eindeutige OID
  - OIDs werden im ODBMS automatisch intern verwaltet

Nachteile

- geringe Verbreitung
  - keine standardisierte Schnittstelle
  - aufwändige Migration
- keine Skalierung wie bei objektrelationalen Abbildungen
- bestimmte Anfragen haben exponentielle Komplexität
  (bspw. exklusive Schreibsperre bei multiplen Zugriffspfaden zu Objekten (Vererbung und Assoziation))
  → deutliche Performance-Einbußen

Beispiel

Speichern des ersten Objekts

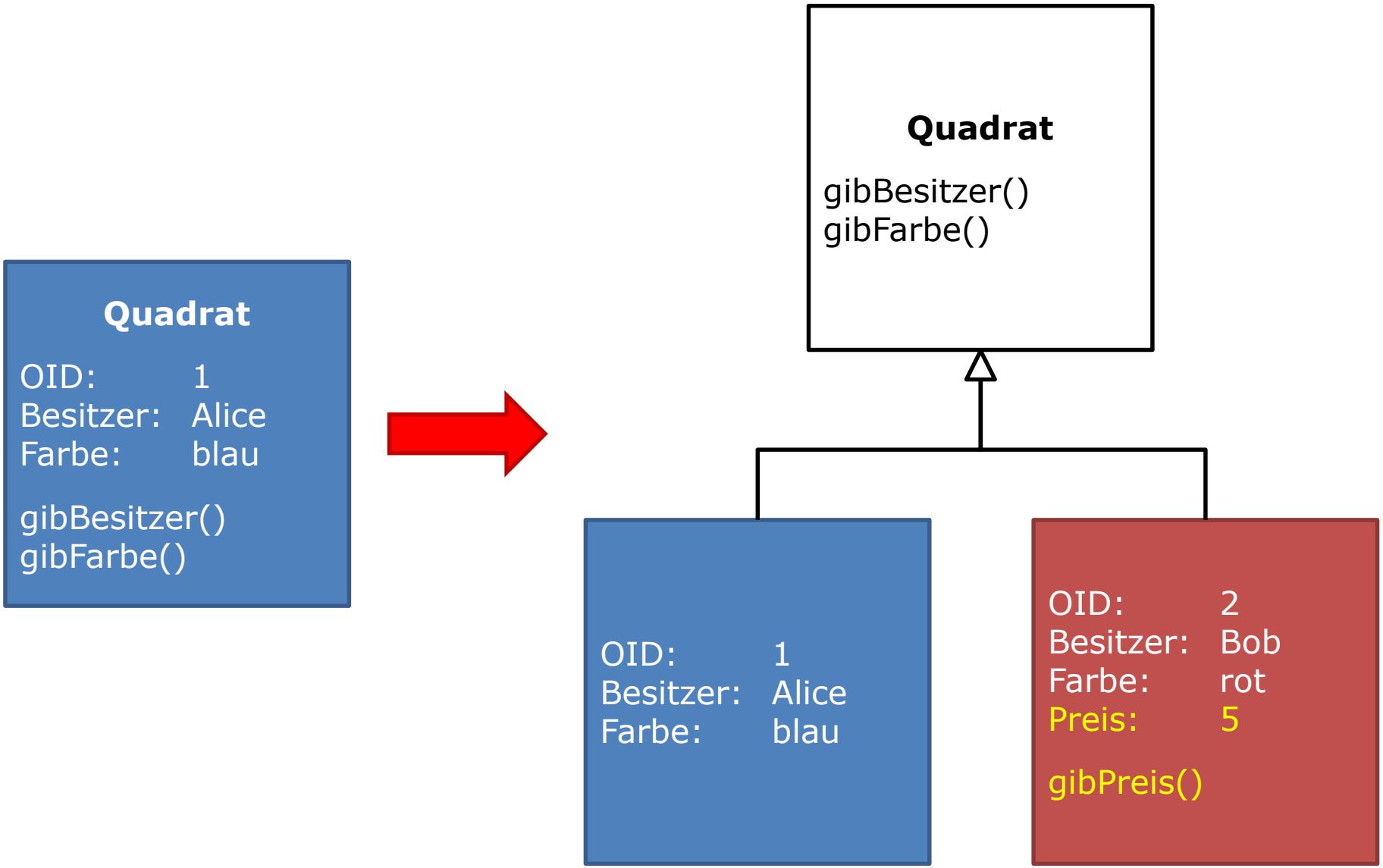
ε

Quadrat

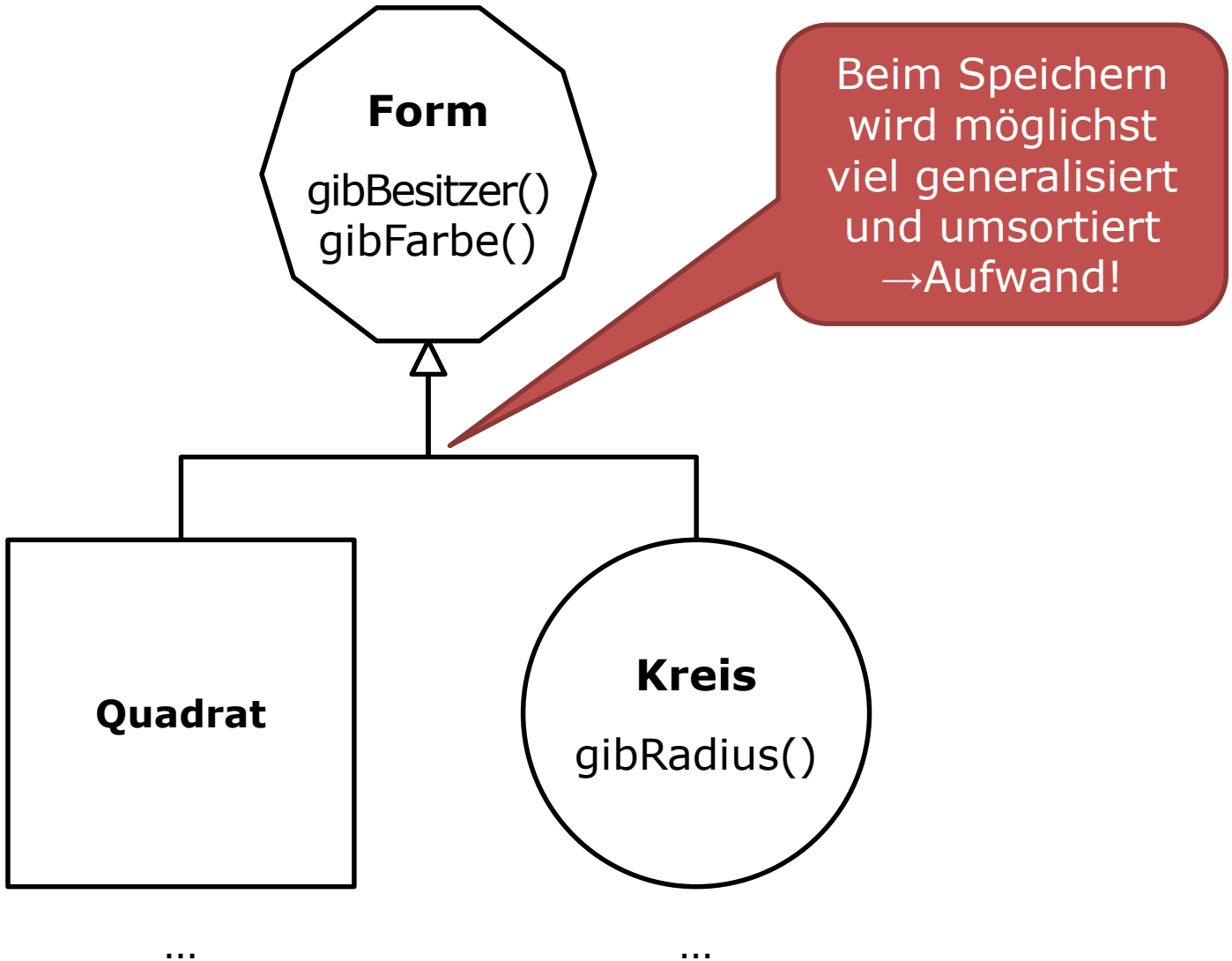
OID: 1
Besitzer: Alice
Farbe: blau

gibBesitzer()
gibFarbe()

Speichern des zweiten Objekts



Speichern eines anderen Objekts



Ausblick

Mehr zu Datenbanken und
Datenbank-Management-Systemen
in der Lehrveranstaltung

Grundlagen Datenbanken

(Pflichtmodul 3IM-GLDB-30)