

# ***Objektorientierte Datenbanken***

Vorlesung 4  
Sebastian Iwanowski  
FH Wedel

# JDO: 2. Teil

## **Persistenzkonzept von JDO**

**→ siehe OODB 3 Slide 14 - 20**

## **Transaktionskonzepte von JDO**

# Wdh.: Transaktionseigenschaften

**A**tomicity

Alle Aktionen auf einmal beim commit()

**C**onsistency

hängt von der Sorgfalt des Programmiers ab

**I**solation

Eine Transaktion darf nicht die Daten für eine andere ändern !

**D**urability

wird durch die Persistenz der Datenbank erreicht



## Lösungsalternativen:

- **Nur eine Transaktion zur selben Zeit**
- **Sperrung der von einer Transaktion benutzten Daten für alle anderen**

# Isolationsproblem bei Transaktionen

## 4 Isolationsniveaus (nach ISO):

- **Lesen aus nicht abgeschlossenen Transaktionen (Uncommitted Read)**

überhaupt keine Gewähr der Konsistenz

- **Lesen nur aus der Datenbank (Committed Read)**

keine gegenseitigen Überschreibungen von gleichzeitig laufenden Transaktionen  
wiederholtes Lesen aus Datenbank kann unterschiedliche Resultate liefern

- **Wiederholungssicheres Lesen (Repeatably Read)**

wiederholtes Lesen aus Datenbank liefert immer dieselben Resultate  
wiederholte Anfragen an die Datenbank kann unterschiedliche Resultate liefern

- **Lesen nur von Daten mit Schreibsperre (Completely Isolated)**

wiederholtes Lesen aus Datenbank liefert immer dieselben Resultate  
wiederholte Anfragen an die Datenbank liefern immer dieselben Resultate

# Isolationsproblem bei Transaktionen

## 2 Sperrmöglichkeiten

- **Lese- und Schreibsperre**

Exklusive Sperre: Eine Transaktion blockiert alle anderen

- **nur Schreibsperre**

Gemeinsame Sperre: Keine Transaktion darf schreiben, aber alle dürfen lesen

Exklusive Sperre: Nur eine Transaktion darf schreiben, aber alle dürfen lesen

# JDO-Transaktionskonzept

## 3 Transaktionsstrategien:

- **Normale (pessimistische, datastore) Transaktionen**

Bei Zugriff wird eine Sperre auf das betreffende Datenbankobjekt gelegt.  
Erst bei Transaktionsende wird die Sperre aufgehoben.

- **Optimistische Transaktionen**

Es gibt während der Transaktion die meiste Zeit keine Sperre.

Es wird vor dem Commit nachgeprüft, ob die benutzten Daten sich während der Transaktion geändert haben (dabei gibt es eine Sperre).

Falls sich die Daten während der Transaktion geändert haben, erfolgt eine Nachricht an die Transaktion als Exception und kein Commit.

- **Datenbankzugriff außerhalb von Transaktionen**

für Objekte in bestimmten Zuständen (wird später besprochen)

# JDO-Transaktionskonzept

## Technische Realisierung:

- **in jedem PersistenceManager nur eine Transaktion gleichzeitig**  
erhältlich durch `PersistenceManager.currentTransaction()`  
Verschachtelte Transaktionen sind verboten (anderenfalls `JDOUserException`).
- **verschiedene PersistenceManager in verschiedenen Threads**  
Damit sind parallele Transaktionen möglich.
- **individuelle Einstellung des Transaktionstyps**  
durch `Transaction.setOptimistic(bool)`
- **Aktualisierungsmöglichkeit bei zwischenzeitlichen Änderungen**  
durch `PersistenceManager.refresh(obj)`  
oder `PersistenceManager.refreshAll()`