

# Übung OM: Objektrelationales Mapping

Gruppe 8: Lukas Arnold, Patrick Bucher, Christopher James Christensen, Jonas Kaiser, Melvin Werthmüller

## 1. Selbststudium

### Frage 1

Was ist der Unterschied zwischen objektorientierten und objektrelationalen Datenbanken?

- In einer objektorientierten Datenbank werden Objekte in der selben Struktur abgelegt, wie sie in einer objektorientierten Anwendung bestehen.
- In einer objektrelationalen Datenbank werden Objekte auf Tabellen abgebildet und umgekehrt.

### Frage 2

Was ist ein Surrogat?

- Ein Surrogat ist ein Schlüsselwert, der im Gegensatz zu einem natürlichen Schlüssel nicht aus den Informationen eines Tupels besteht, sondern künstlich generiert und unveränderlich ist.

### Frage 3

Was ist ein NF<sup>2</sup>-Modell?

- NF<sup>2</sup> bedeutet *Non First Normal Form* und beschreibt ein Modell, das die erste Normalform aufgibt und stattdessen komplexe Merkmale wie Tabellen zulässt.

### Frage 4

Was ist der Zusammenhang von NF<sup>2</sup> mit der Objektorientierung?

- Bei der objektorientierten Programmierung können PART\_OF-Beziehungen mittels Komposition umgesetzt werden.
- Bei NF<sup>2</sup> können Objekte Unterobjekte als Merkmale aufnehmen.

## Frage 5

Was heisst objektrelationales Mapping?

- Beim objektrelationalen Mapping wird die objektorientierte Struktur einer Anwendung auf Relationen einer Datenbank umgesetzt.
- Dabei wird nicht jede Tabelle zu einem Objekt bzw. jedes Objekt zu einer Tabelle, da Beziehungstabellen als Objektreferenzen umgesetzt werden.

## 3. Fragen

1. Schauen Sie sich die generierte Datei persistence.xml an. Wie heisst die Persistence Unit, welche bei Ihnen generiert worden ist?
  - ORMPU
2. Schauen Sie sich die generierte Datei Professoren.java an. Wie bzw. mit welchen Annotationen wurden die Primär- und Fremdschlüssel spezifiziert?
  - Primärschlüssel: @Id
  - Fremdschlüssel: @OneToMany
3. Schauen Sie die generierte Datei Studenten.java an. Mit welchen Annotationen wurde die many-to-many-Relationship umgesetzt? Wie sieht es diesbezüglich bei Vorlesungen.java aus?
  - Studenten.java: @ManyToMany
  - Vorlesungen.java: @ManyToMany
4. Wie wurde NOT NULL in JPA-Annotationen umgesetzt?
  - @Basic(optional = false)

## 4. Programmierung

// A.

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory("ORMPU");
EntityManager em = emf.createEntityManager();
```

// B.

```
Query findAllProfessoren = em.createNamedQuery("Professoren.findAll");
List<Professoren> professoren = findAllProfessoren.getResultList();
for (Professoren professor : professoren) {
    System.out.println(professor.getName());
    for (Vorlesungen vorlesung : professor.getVorlesungenCollection()) {
        System.out.println("- " + vorlesung.getTitel());
    }
    System.out.println();
}
```

```

// C.
Query sokratesVorlesungen = em.createQuery(
    "SELECT v FROM Vorlesungen v " +
    "JOIN v.gelesenVon p " +
    "WHERE p.name = 'Sokrates'",
    Vorlesungen.class);
List<Vorlesungen> vorlesungen = sokratesVorlesungen.getResultList();
for (Vorlesungen vorlesung : vorlesungen) {
    System.out.println("Sokrates liest: " + vorlesung.getTitel());
}

// D.
Professoren sokrates = em.createQuery(
    "SELECT p FROM Professoren p " +
    "WHERE p.name = 'Sokrates'",
    Professoren.class).getResultList().get(0);
Query sokratesRaum = em.createQuery("UPDATE Professoren p " +
    "SET p.raum = 1234 " +
    "WHERE p.name = 'Sokrates'");
EntityTransaction et = em.getTransaction();
et.begin();
em.merge(sokrates);
sokratesRaum.executeUpdate();
et.commit();

// E.
Professoren precht = new Professoren(Integer.MAX_VALUE, "Precht");
et.begin();
em.persist(precht);
et.commit();

// F.
Vorlesungen postmoderne = new Vorlesungen(Integer.MAX_VALUE);
postmoderne.setGelesenVon(precht);
postmoderne.setTitel("Postmoderne");
et.begin();
em.persist(postmoderne);
et.commit();

```