

Konzepte objekt-orientierter Zugriffstechniken

a)

Das Konzept der Speicherresidenten Datenbankstruktur in ADO.NET ist, dass es ein objektorientiertes Modell einer generischen Datenbank gibt. Dabei gibt es die Klasse DataSet, die die Datenbank darstellt. Diese Datenbank hat wiederum Tabellen, die von der Klasse DataTable repräsentiert wird und diese Tabelle hat Spalten, die von der Klasse DataColumn repräsentiert werden. Aber natürlich haben die Tabellen auch Zeilen, für die die Klasse DataRow steht und in den Zeilen können noch Werte stehen, die von der Klasse Item repräsentiert werden.

Es wird also deutlich, dass es bei ADO.NET keine spezifischen Klassen gibt, die nur auf eine spezielle Datenbank zugeschnitten sind, sondern es ist möglich, mit diesen Klassen eine generische Datenbank abzubilden.

b)

Das grundlegende Konzept der Java Persistence API besteht darin, dass Klassen Entitätstypen sind, die in ihren Attributen bestimmte Werte enthalten. Die Objekte dieser Klassen wiederum sollen persistent sein. Um eine Verbindung (objektrelationales Mapping) zwischen Datenbank und der entsprechenden Klasse herzustellen, werden Annotationen und ein Entitymanager verwendet. Wenn es also eine Tabelle Buch mit den Spalten Autor, Buchtitel und Preis in einer Datenbank geben würde, würde dies wie folgt aussehen:

```
@Entity
public class Buch{
    @id
    String Autor;
    String Buchtitel;
    int Preis;
    ...
}
```

Diese Klasse Buch würde in diesem Beispiel also eine spezifische Tabelle in einer Datenbank verkörpern.

Über bestimmte Funktionen wird es wiederum ermöglicht, auf die Attribute der Klasse zuzugreifen.

In der Java Persistence API werden also Klassen einer Tabelle zugeordnet.

c)

Der grundlegende Unterschied der beiden Konzepte besteht darin, dass ADO.NET das objektorientierte Modell einer generischen Datenbank verwendet und die Java Persistence API das objektorientierte Modell einer spezifischen Datenbank verwendet.