

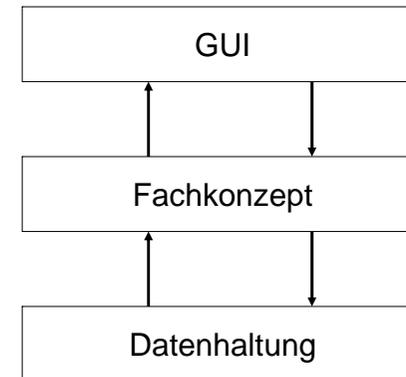
Informatik B - Objektorientierte Programmierung in Java

Vorlesung 17: 3-Schichten-Architektur 2 Fachkonzept - Datenhaltung

© SS 2005 Prof. Dr. F.M. Thiesing, FH Dortmund

Drei-Schichten-Architektur

- Zur Erinnerung: Aufteilung der Schichten



© Prof. Dr. Thiesing, FH Dortmund

Inhalt

Drei-Schichten-Architektur (3-Tier-Architecture)

- Anbindung FK – DB
 - Beispiel: Schritt4

© Prof. Dr. Thiesing, FH Dortmund

Drei-Schichten-Architektur

- Im ersten Teil dieser Vorlesung wurde gezeigt, wie eine Fachkonzeptklasse (`Buch`) an eine grafische Benutzungsoberfläche angebunden wird, so dass die Fachkonzeptklasse keinerlei Wissen über die Darstellung Ihrer Objekte hat.
- Im folgenden Teil der Vorlesung wird dies nun für die persistente (dauerhafte) Speicherung der Objekte durchgeführt. Auch hier hat anschließend die Fachkonzeptklasse keinerlei Wissen über die Art und Weise, wie und wo die Objekte abgespeichert werden.

© Prof. Dr. Thiesing, FH Dortmund

Drei-Schichten-Architektur

- Somit liegt also eine klare Trennung zwischen den einzelnen Schichten vor.
- Der Vorteil dieser allgemeinen Vorgehensweise liegt eindeutig darin, dass die Art der Speicherung einfach austauschbar wird.
- Eine Speicherung in eine flache Datei ist genauso möglich, wie die Speicherung in eine relationale oder objektorientierte Datenbank.

Anbindung FK - DB

- In der Tabelle, die für jede Fachkonzeptklasse angelegt werden muss, werden die Attributwerte der Objekte abgespeichert.
- Damit Objekte unterschieden werden können, ist ein eindeutiger Schlüssel hinzuzufügen.
- Dazu ist die Klasse `Buch` um das Attribut `Schlüssel` und die entsprechenden `get-` und `set-` Operationen zu erweitern.
- Ein neuer Konstruktor, durch den alle drei Attribute initialisiert werden, ist ebenfalls eingefügt.

Anbindung FK - DB

- Wir wollen nun unsere Fachkonzeptklasse `Buch` an eine relationale Datenbank anbinden, der in der industriellen Praxis wohl häufigste Fall.
- Grundsätzlich gilt beim Neuaufbau einer DB:
 - Jedes Objekt einer Fachkonzeptklasse wird in einer relationalen Tabelle abgespeichert. Ein einzelnes Objekt wird zu einer Tabellenzeile.
 - Die Attributnamen der Fachkonzeptklasse bilden die Spaltenüberschriften

Tabelle: Buch

Schlüssel	Titel	Preis
1	Goto Java 2	99,90
2	Swing Tutorial	49,95
...		

Anbindung FK - DB

```
public class Buch
{
    //Buch Attribute
    private String name;
    private float preis;
    private int schluessel;
    public Buch(int schluessel, String name, float preis)
    {
        this.schluessel = schluessel;
        this.name = name;
        this.preis = preis;
        ...
    } // Konstruktor
    ...

    int getSchluessel(){return schluessel;}
    void setSchluessel(int schluessel) {this.schluessel =
        schluessel;}
    ...
}
```

Anbindung FK - DB

VL 17
9

- Zum grundsätzlichen Umgang, d.h. zum Verbinden und Schließen einer relationalen Datenbank, wird die Klasse `DBZugriff` erstellt.
- Dort befindet sich der Name unserer Buchdatenbank und Operationen zum Verbinden und Schließen der Datenbank.

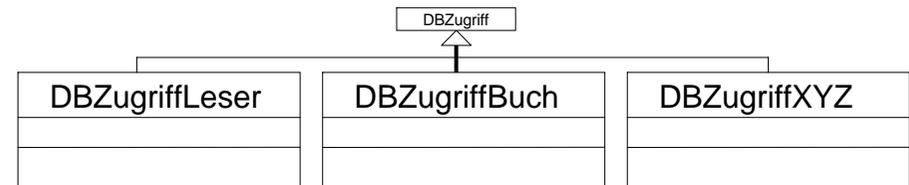


© Prof. Dr. Thiesing, FH Dortmund

Anbindung FK - DB

VL 17
11

- Sollen mehrere Fachkonzeptklassen in dieser Datenbank abgespeichert werden, so ist es offensichtlich, dass für jede dieser Fachkonzeptklassen eine eigene `DBZugriff`-FK-Klasse existieren muss, in der Operationen für die spezifischen Attribute der Klassen enthalten sein müssen.

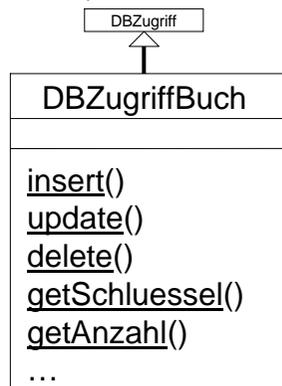


© Prof. Dr. Thiesing, FH Dortmund

Anbindung FK - DB

VL 17
10

- Aus dieser allgemeinen Klasse wird die spezielle Klasse `DBZugriffBuch` abgeleitet, die die speziellen Operationen zum Speichern, Löschen und Ändern von Büchern enthält.
- Außerdem befinden sich in dieser Klasse Operationen zur Verwaltung der Schlüssel und zur Abfrage der der DB-Tabelle.



© Prof. Dr. Thiesing, FH Dortmund

Anbindung FK - DB

VL 17
12

- In der Klasse `BuecherVerwaltung` müssen nun die bisherige Objektverwaltung und die persistente Speicherung der Daten miteinander verknüpft werden.
 - Wird ein neues Buchobjekt erzeugt, wird es direkt in der Datenbank abgespeichert.
 - Wird ein bestehendes Buchobjekt verändert, werden die Änderungen ebenfalls direkt abgespeichert.
 - Wird ein Buchobjekt gelöscht, wird dies ebenfalls direkt an die Datenhaltung weiter gegeben.

© Prof. Dr. Thiesing, FH Dortmund

Anbindung FK - DB

■ Die Klasse BuecherVerwaltung:

```
public class BuecherVerwaltung
{
    static public void store(Buch buch)
    {
        buch.setSchluessel(DBZugriffBuch.getSchluessel()+1);
        DBZugriffBuch.insert(buch.getSchluessel(),
            buch.getName(), buch.getPreis());
    }
    static public void update(Buch buch)
    {
        DBZugriffBuch.update(buch.getSchluessel(),
            buch.getName(), buch.getPreis());
    }
    static public void delete(Buch buch)
    {
        DBZugriffBuch.delete(buch.getSchluessel());
    }
}
```

© Prof. Dr. Thiesing, FH Dortmund

Anbindung FK - DB

■ Zum Aufbau einer relationalen Datenbank unter Windows ist wie folgt vorzugehen:

- 1.) ODBC Datenquelle einrichten
 - ◆ Start → Systemsteuerung
 - ◆ Öffnen von :
 - W2k/NT/XP : Verwaltung→Datenquelle (ODBC)
 - 95,98, ME: ODBC-32 Datenquelle (oder ähnlich)
 - ◆ Hinzufügen (Reiter „Benutzer-DSN“)
 - Microsoft Access-Datenbank *hinzufügen*
 - Microsoft-Access-Driver (*.mdb) *auswählen*
 - Fertig stellen →
 - ◆ Datenquellname: BuchDB
(Bitte auf die richtige Eingabe achten!!!)
 - ◆ Beschreibung: BuchprogrammDB

© Prof. Dr. Thiesing, FH Dortmund

Anbindung FK - DB

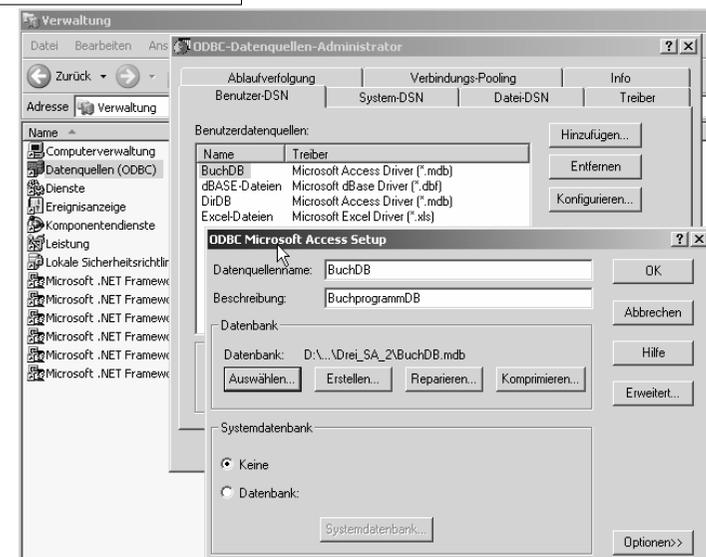
■ Die Klasse BuecherVerwaltung:

```
static public Buch[] getAll() {
    Buch hilfsbuch[] =
        new Buch[DBZugriffBuch.getAnzahl()];
    DBZugriffBuch.StartAbfrage();

    for(int i=0; i < hilfsbuch.length; i++){
        hilfsbuch[i] = DBZugriffBuch.getNext();
    }
    DBZugriffBuch.schliessen();//in StartAbfrage geöffnet
    return hilfsbuch;
}
} // BuecherVerwaltung
```

© Prof. Dr. Thiesing, FH Dortmund

Anbindung FK - DB



ng, FH Dortmund

Anbindung FK - DB

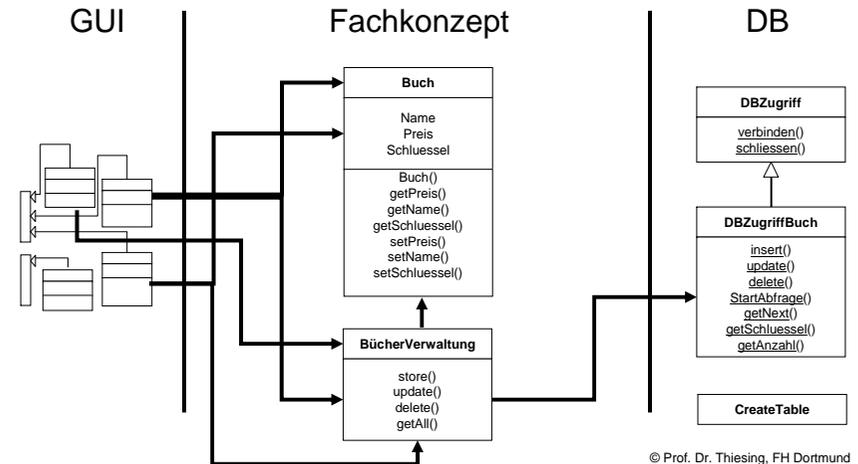
VL 17
17

- ◆ Erstellen → beliebigen Dateinamen und Ordner auswählen → OK
- ◆ Meldung bestätigen
- ◆ Erweitert →
- ◆ Anmeldenname: dbadmin (Bitte auf die richtige Eingabe achten!!!)
- ◆ Kennwort: info2 (Bitte auf die richtige Eingabe achten!!!)
- ◆ OK
- ◆ OK
- ◆ Alle restlichen Fenster schließen.

Anbindung FK - DB

VL 17
19

■ Damit ergibt sich als Klassendiagramm:



Anbindung FK - DB

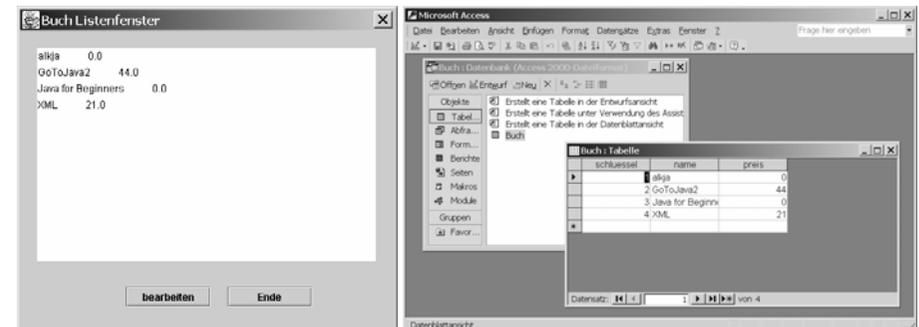
VL 17
18

- 2.) Datenbank einrichten
 - ◆ Einmaliges compilieren und ausführen der CreateTable.java - Datei
 - 3.) BuchHauptprogramm starten
- Alle Quelltexte befinden sich im Verzeichnis Schritt_4.

Anbindung FK - DB

VL 17
20

- Beim Aufruf des Programms wird nun jedes neu erzeugte oder geänderte Buchobjekt direkt in die relationale Datenbank abgespeichert.
- Diese kann unter Access angesehen werden:



Anbindung FK - DB

- Nach Beendigung des Programms und erneutem Start sind alle abgespeicherten Bücher wieder anzeigbar.
- Eine Variante zur direkten Speicherung aller neuen bzw. aller geänderten Objekte wäre, dass nur einmal beim Start des Programms alle Bücher aus der Datenbank gelesen werden und in einer Objektverwaltung im Hauptspeicher abgespeichert würden. Vor der Beendigung des Programms sollten dann alle Buchobjekte wieder in die Datenbank abgespeichert werden.