

# 9 Spielzeugauktion mit MySQL

Stand: 28. August 2003; Erläuterung der Einzelprogramme fehlt noch

<http://vsite.de/auktion.html>

## Inhalt

9.1	Das Konzept	432
9.1.1	Das Datenmodell	433
9.1.2	Die GUIs	434
9.2	Das Programm	436
9.2.1	MVC-Struktur	437
9.2.2	Der Controller	438
9.2.3	Die Models	442
9.2.3.1	Verbindung.java	442
9.2.3.2	Bieter.java	443
9.2.3.3	Auktionskatalog.java	443
9.2.3.4	Artikel.java	445
9.2.3.5	Bieterkatalog.java	446
9.2.3.6	Gebot.java	447
9.2.4	Die Views	448
9.2.4.1	Login.jsp	448
9.2.4.2	Bieterkatalog.jsp	450
9.2.4.3	Auktionskatalog.jsp	454
9.2.4.4	Artikel.jsp	457
9.2.4.5	Artikelbearbeiten.jsp	459
9.3	Übungsaufgaben	463

## 9.1 Das Konzept

Die „virtuelle“ Spielzeugauktion besteht aus drei Teilen,

1. dem *Auktionskatalog*, in dem die Versteigerungsangebote aufgeführt sind,
2. dem *Bieter* und
3. einem *Bieterkatalog*, in dem der Kunde aus dem Auktionskatalog diejenigen Artikel einstellt, die er beobachten und eventuell ersteigern möchte.

In „Auktionskatalog“ und „Bieter“ manifestieren sich *Stammdaten*, und im „Bieterkatalog“ *Bewegungsdaten*. (Messerscharf ist diese Trennung allerdings nicht vorgenommen, denn der Auktionskatalog enthält z.B. mit dem aktuellen Höchstgebot gleichfalls Bewegungsdaten.) Der Bieterkatalog ist ganz offensichtlich ein so genannter *Warenkorb*.

Die drei Teile spiegeln sich in den Datentabellen wider, die weiter unten eingehend behandelt werden.

Die Versteigerungsregeln sind die folgenden:

- Voraussetzung für die Teilnahme ist, dass der Bieter registriert ist und Bieterprivileg genießt. Nicht registrierte Nutzer haben lediglich Einsicht in den Auktionskatalog.
- Gebote können nur für die Gegenstände im Bieterkatalog abgegeben werden.
- Das Mindestgebot ist durch ein „Limit“ vorgegeben.
- Ein Gebot ist gegenüber dem jüngsten Vorgebot immer um einen festen Prozentsatz (hier 2,5%) erhöht. Centbeträge werden dabei auf einen vollen Euro gerundet.
- Gebote können bis zum angegebenen Endtermin zu jeder Zeit abgegeben werden.
- Ist der Endtermin überschritten, so kann dann weitergesteigert werden, wenn weniger als 5 Minuten seit dem letzten Vorgebot vergangen sind.

Die Versteigerung ist also beendet, wenn (a) der Endtermin überschritten ist *und* wenn (b) seit dem letzten Gebot mehr als 5 Minuten vergangen sind. Den Zuschlag erhält der Bieter, der das letzte Gebot termingerecht abgeben konnte.

Die Programmierung beschränkt sich weitgehend auf die Bieterseite, lediglich die Pflege des Auktionskataloges ist in das Programm aufgenommen. Der Versteigerungsvorgang ist auch noch nicht völlig korrekt, denn

- der Bieter kann sich selbst überbieten und
- der Preis eines angezeigten Artikels kann sich unbemerkt ändern, was beim Bietevorgang nicht berücksichtigt ist.

Die Bereinigung dieser Situation ist relativ einfach möglich und dem Leser als Übungsaufgabe anempfohlen (siehe Abschnitt 9.3).

Die *Datenbank* der Spielzeugauktion besteht aus drei Tabellen, siehe Abschnitt 9.1.1:

1. einem *Auktionskatalog*, in dem alle Versteigerungsgegenstände registriert sind,
2. einem *Bieter*, d.h. dessen Daten und Zugangsberechtigungen, und
3. einem *Bieterkatalog*, in dem diejenigen Gegenstände (Artikel) gespeichert sind, die beobachtet und eventuell ersteigert werden sollen.

### 9.1.1 Das Datenmodell

Das Datenmodell ist sehr einfach gehalten und besteht aus insgesamt drei Tabellen, die im Folgenden genauer ausgeführt sind. Unterstreichungen kennzeichnen dort Primärschlüssel, und Schrägstellung Fremdschlüssel. Die Spalte `item` z.B. ist beides, nämlich zusammen mit der Spalte `bieter` Primärschlüssel in `bieterkataloge` und alleine Fremdschlüssel mit Verweis in die Tabelle `auktionskatalog`.

Zunächst der Auktionskatalog, der neben den Artikel-Stammdaten auch einige Bewegungsdaten enthält:

**auktionskatalog**(snr, *bieter*, *bis*, *gebotsdatum*, *limit*, *gebot*, *kategorie*, *beschreibung*, *bild*)

Die Tabellenspalten sind:

<u>snr</u>	Diese Artikelnummer ( <code>snr</code> - Spielzeugnummer) ist Primärschlüssel der Tabelle. <code>snr</code> ist eine ganze Zahl.
<i>bieter</i>	Eine ganzzahlige Bieternummer als Fremdschlüssel, der auf die Spalte <code>bnr</code> in der Tabelle <code>bieter</code> weist. Enthalten ist die Kennung des Bieters, der das aktuelle Höchstgebot abgegeben hat.
<i>limit</i>	So genanntes Limit, d.h. der zumindest zu bietende Betrag.
<i>bis</i>	Angebot endet mit Ablauf dieses Zeitpunkts (Datum, Uhrzeit).
<i>gebot</i>	Das zuletzt abgegebene (Höchst-) Gebot.
<i>gebotsdatum</i>	Zeitpunkt des zuletzt abgegebenen Gebots (Datum, Uhrzeit).
<i>kategorie</i>	Kategorie des Artikels, z.B. „Auto“ oder „Eisenbahn“
<i>beschreibung</i>	ist Text zur Beschreibung des Artikels.
<i>bild</i>	URL zu einer Illustration des Artikels.

(Sinnvolle Ergänzung wäre die Identität des Anbieters dieses Artikels als Fremdschlüssel, der in eine dazu passende, vierte Tabelle `anbieter` weist.)

Es folgt die Personalie `bieter`. In Ihr sind neben den üblichen Personendaten Kennung (`uid`) und Passwort (`passwd`) enthalten, über die die Berechtigung zum Mitbieten und der Zugang zur Katalogpflege reguliert wird.

**bieter**(bnr, *nachname*, *vorname*, *plz*, *ort*, *strasse*, *uid*, *passwd*, *privileg*)

Die Spalten dieser Tabelle sind:

<u>bnr</u>	Bieternummer als ganzzahliger Primärschlüssel.
<i>nachname</i>	Nachname und
<i>vorname</i>	Vorname des Bieters,
<i>plz</i>	die Postleitzahl seines Wohnorts,
<i>ort</i>	der Wohnort selbst,
<i>strasse</i>	die Straße,
<i>uid</i>	die Benutzerkennung des Bieters und
<i>passwd</i>	sein Passwort.
<i>privileg</i>	Privilegien des Bieters, mit denen der Zugang zum Bieten reglementiert ist.

Im Bieterkatalog schließlich sammelt der Bieter die Versteigerungsgegenstände, zu denen er Gebote abzugeben beabsichtigt. Jeder Bieter verfügt über genau einen Bieterkatalog, denn Gebote können ausschließlich zu Artikeln in Warenkörben abgegeben werden. Einträge sind Fremdschlüsselpaare, die auf Bieter (`bieter`) und Angebot (`item`) weisen.

**bieterkataloge( bieter, item )**

mit den Tabellenspalten:

<u>bieter</u>	Fremdschlüssel mit Verweis in die Tabelle <code>bieter</code> ( Spalte <code>bnr</code> ).
<u>item</u>	Fremdschlüssel mit Verweis in die Tabelle <code>auktionskatalog</code> (Spalte <code>snr</code> ).

## 9.1.2 Die GUIs

### Home / Login

Wird die Homepage der Auktion mittels z.B.

`http://.../auktion/Auktion.jsp`

aktiviert, so erscheint die folgende GUI.

Sie besteht aus drei Teilen: einem Login, einem Logout und in der Mitte einem Teil, der zum Auktionskatalog führt.

Nach Eingabe eines gültigen Bieter-Id/Passwort-Paars wird die individuelle Katalogauswahl des Bieters in Form eines „Bieterkatalog“ genannten Warenkorbs angezeigt. Die Auflistung des Kataloginhalts ähnelt weitgehend der im Auktionskatalog, hat aber einige Besonderheiten, die kurz zu erläutern sind.

**Aktualisieren** Mit diesem Knopf werden die angezeigten Daten auf den neuesten Stand gebracht. Damit wird insbesondere das jüngste Gebot samt der Id des Bieters aktualisiert und angezeigt.

**Zum Auktionskatalog** führt zum Auktionskatalog, beispielsweise um dort weitere Artikel auszuwählen und in den Bieterkatalog einzustellen.

**Bieten** Damit wird das laufende Gebot um ca. 2,5% überboten.

**Passen** Der Bieter gibt auf, indem er den Artikel aus seinem Bieterkatalog entfernt.

**Anschaun!** Den Artikel mitsamt seiner Abbildung einzeln anzeigen!

### Spielzeugauktion - Bieterkatalog bearbeiten

Bieter: Heinrich Karl Schneider (3/2)

<input type="button" value="Aktualisieren!"/>		<input type="button" value="Zum Auktionskatalog!"/>			
Heinrich Karl Schneider Richard-Wagner-Allee 2 69355 Schriesheim					UID: zwei Bieter-Nr./Privileg: 3/2 20.07.2003 16:44:15
Bieten	Termin	Limit	Gebot	Kategorie	Beschreibung
<b>109</b> <input type="button" value="Bieten"/> <input type="button" value="Passen"/>	21.08.2003 12:00:00	1900.00	<b>1948.00 €</b> Bieter 4	Auto <input type="button" value="Anschau!"/>	MARKKLIN, Baukastenauto, Pullmann-Limousine, grün/beige, 28 cm, mit Uhrwerk, sehr guter Orig.-Zust.
<b>115</b> <input type="button" value="Bieten"/> <input type="button" value="Passen"/>	21.08.2003 12:00:00	1500.00	<b>1500.00 €</b> Bieter 3	Puppe <input type="button" value="Anschau!"/>	Biskuitporzellan-Puppe, Frankreich, Dep. Tête JUMEAU, Nr. 8, 46 cm, Kurbelkopf, blaue festeingesetzte Glasaugen, geschl. Mund, durchst. Ohrläppchen, franz. Gliederkörper, alte Kleidung u. Schuhe, Perücke neuer, kleiner Haarriß an der Schläfe, Glas am Auge gesprungen

Der Auktionskatalog wird über den „Zum Auktionskatalog!“-Taster auf der Bieterkatalog- oder Home-Seite (oder den anderen Seiten) aktiviert. Die Artikel werden an Kategorien orientiert angezeigt, wobei die Kategorie über ein Popup-Menü gewählt werden kann, etwa „Spiel“, „Burg“ oder „Puppe“. Nicht mehr ersteigerbare Auktionsartikel (Ablauf des Termins) sind durch Verwendung von Fettschrift kenntlich gemacht.

### Spielzeugauktion - Auktionskatalog anzeigen

Bieter: Heinrich Karl Schneider (3/2)

<input type="button" value="Zum Bieterkatalog"/>				Schiff	
Termin	Limit	Gebot	Kategorie		Beschreibung
Pos. 105 21.08.2003 12:00:00	600.00	<b>600.00 €</b>	Auto <input type="button" value="Anschau!"/>	Schiff	
Pos. 106 21.08.2003 12:00:00	1900.00	<b>1900.00 €</b>	Auto <input type="button" value="Anschau!"/>	Auto	MARKKLIN Baukastenauto, Panzerwagen, mit Uhrwerk, sehr guter Zust., kompl. mit Zugleine, eines der gesuchtesten Baukastenautos
Pos. 107 19.07.2003 20:59:00	280.00	0.00 €	Auto <input type="button" value="Anschau!"/>	Burg	BANDAY, Rolls Royce, 29 cm, Friktionsantrieb, bespielt, gut

Das Drücken des „Anschau!“-Tasters sowohl im Bieter- als auch im Auktionskatalog führt zu einer Einzelansicht des Artikels. Der angezeigte Artikel kann mittels des Tasters „Aufnehmen im Bieterkatalog!“ in den Bieterkatalog aufgenommen werden, sofern er dort noch nicht ist. (Dieser Taster fehlt übrigens, wenn die Einzelartikelanzeige über den Bieterkatalog angefordert wurde.)

**Spielzeugauktion - Artikel anzeigen**

Heinrich Karl Schneider (3/2)

<p><b>Item: 105</b>  Akt. Höchstgebot (Limit): 600.0 € ( 600.0)  Höchstbieter Nr.: 4  Gebotsdatum: 2003-08-20 14:31:28  Kategorie: Auto  Termin: 2003-09-21 12:00:00  Beschreibung:  <i>Rennwagen, Benz, 43 cm, Blech, handgelötet, aufwendiger  Messingkühler mit Symbol BENZ, mit Lorbeer-kranz, elektr.  beluchtbar, ohne übriges Spielzeugmaterial, Gummiwaren etc.</i></p>	
---	--

Hat der Bieter Privilegien ab 4 und darüber, ist unter der Einzelanzeige außerdem noch ein Taster „Artikel bearbeiten!“ zu finden (bei Privilegien unter 4 fehlt er). Betätigen führt zu der Eingabemaske unten. In ihr können bestehende Artikel geändert oder gelöscht werden, oder es kann ein neuer Artikel in den Katalog aufgenommen werden.

<b>Item:</b>	104
<b>Schlussstermin:</b>	2003-09-21 12:00:00
<b>Limit:</b>	1600.0
<b>Kategorie:</b>	Schiff
<b>Bild-URL</b>	bilder/104.jpg
<b>Beschreibung:</b>	ARNOLD, Passagier-Schiff, 31 cm, 2 Schrauben, 7 Rettungsboote auf jeder Seite, 2 Masten, Fahne, mit Schlüssel, unbespielter Orig.-Zust., Uhrwerk intakt
<b>Höchstbietender:</b>	0
<b>Akt. Höchstgebot:</b>	0.0 €
<b>Gebotsdatum:</b>	2003-08-20 10:11:28

## 9.2 Das Programm

Es ist möglichst schlicht gestrickt und sehr direkt codiert. Der Grund ist ganz einfach der, dass eine Ausprogrammierung, die weniger Kompromisse bei Anzeige und Bedienung macht, die eine ordentliche Fehlerbehandlung beinhaltet etc., weitaus mehr Code erfordern würde – was letztlich das Verständnis dessen, was mit dem Programm vermittelt werden sollte, unnötig erschweren würde. So fehlt beispielsweise ein „Recovery“, falls der Datenbank-Server abstürzen sollte, und auf Ergonomie ist weniger geachtet als z.B. darauf, dass Sonderfälle und -situationen soweit als möglich vermieden werden. Da es

sich um ein Lern- und nicht um ein Produktionsprogramm handelt, ist das aber ziemlich egal. Bei schönem Wetter und pflegeleichten Arbeitsumgebungen jedenfalls funktioniert alles ganz hervorragend.

### 9.2.1 MVC-Struktur

Das Programm hat, wie in Abb. 9-1 zu sehen, eine recht einfache MVC-Architektur. Der *Controller* nimmt *alle* http-Anforderungen entgegen. Anhand der Anforderungsdaten erkennt er, welche View für die Beantwortung zuständig ist, und welches *Model* sie dazu benötigt. Die *View* ruft das vom Controller vorbereitete Model ab und bedenkt die Anforderung mit einer http-Antwort. (Die Antwort stammt *immer* von einer View.) Diese Antwort ist in der Regel eine HTML-GUI, von der erneut http-Anforderungen an den Controller gehen können, etc.

Für den Dispatch-Vorgang im Controller („Forward“) wird ausschließlich die `forward()`-Methode verwendet. Für den Anwender präsentiert sich die gesamte Anwendung also immer mit `Auktion.jsp` in der URL als Zielfile, also nicht z.B. mit `Login.jsp` oder `Auktionskatalog.jsp`.

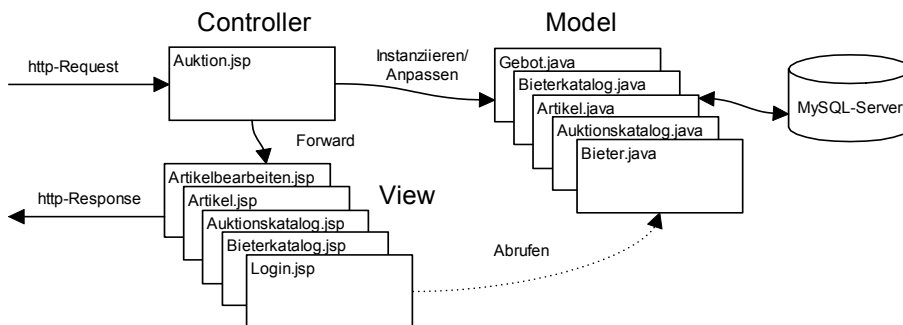


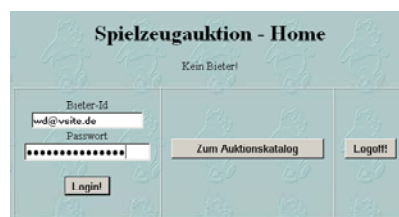
Abb. 9-1: MVC-Architektur der Spielzeugauktion

Die verwendete Technik betreffend ist festzuhalten, dass der Controller zwar JSP-kodiert ist. Er ist aber völlig von HTML- und sonstigen Browser-Artefakten befreit, da seine Aktivität nie unmittelbar zu Anzeigen führt, sondern diese Aufgabe immer an andere JSP-Seiten, die als Views fungieren, per „forward“ delegiert. Er hätte also auch leicht als „bodenständiges“ Servlet programmiert werden können.

Die Models sind gewöhnliche Java-Klassen (d.h. ohne JavaBeans-Eigenschaften), und die Views – naheliegenderweise – JSP-Programme, die Browser-Frontends in (fast) reinem HTML produzieren.

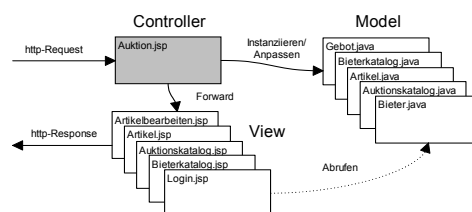
Nach Start des Programmes gibt es die Kontrolle an eine „Home“-View ab, siehe Bild rechts. In Abb. 9-1 ist diese durch die Datei `Login.jsp` repräsentiert.

Drei Aktionen, die im Controller abgehandelt werden, sind dort vorgesehen: Mittels der drei Taster ist ein Login bzw. Logoff möglich, außerdem die Aktivierung des Auktionskataloges. Die beiden Eingabefelder für Bieterkennung und Passwort sind dem Login-Taster zugeordnet.



## 9.2.2 Der Controller

Der Controller ist die zentrale Koordinationsinstanz des Programms. Er besteht aus einer einzigen Datei, in der die Konfigurationsdaten für das Ansprechen und der Verteiler zu den Views enthalten sind.



### Auktion.jsp

Das Programm importiert das Package `auktion`, das die Model-Klassen der Auktionsanwendung enthält. Danach wird entweder die bestehende Sitzung fortgesetzt oder aber eine neue Sitzung definiert, falls es noch keine gibt oder sie ungültig geworden ist. Diese Sitzung wird ihrerseits mit einem Timeout versehen, der nach 2-stündigem Nichtstun erfolgt und die Sitzung ungültig macht.

```
<!-- Spielzeugauktion Version 0.9 -->
<%@ page import="java.util.*, java.sql.*, auktion.*" %>
<%@ page session="true" %>
<% session.setMaxInactiveInterval(2 * 3600); // 2 Stunden %>
<%!
/* Controller: Dispatch-Tabelle
...

```

Das Controller-Programm nimmt alle `http`-Anforderungen entgegen, analysiert sie und delegiert deren Verarbeitung an Models und die Präsentation der Ergebnisse an Views. Es besteht aus zwei Teilen:

1. In einem Definitionsteil ist jede Controller-Aktion bzw. -Reaktion parametrisiert (siehe Programmfragment 9-1). Jeder solchen Aktion ist ein Text zur Identifizierung zugeordnet, im Beispiel etwa "BieterkatalogBieten" oder "AuktionskatalogAnzeigen". Weiter ist für jede Aktion
  - das *Zieldokument*, das „Target“, in der Regel eine JSP-Datei (eine View, um ganz genau zu sein),



- das erforderliche *Nutzungsrecht* in Form einer Art gestuften Nutzerprivilegs (unangemeldet: 0, Bieter: 2, Administrator: 9 etc.) und
- ein *Seitentitel* für die zu aktivierende Browser-Anzeige (als Überschrift, Fensterbetitelung etc. zu nutzen)

definiert. Ziele könnten auch Dokumente in PHP, HTML, Perl etc. sein. Jede Aktion des Controllers ist in einer Hash-Tabelle parametrisiert, und diese Hash-Tabellen sind ihrerseits Elemente eines als Hash-Tabelle ausgelegten Containers. In Programmfragment 9-1a wird beispielhaft im Container `ctrls` eine Hash-Tabelle als Element „BieterkatalogBieten“ definiert und dieses Element mit den Einträgen „target“, „privileg“ und „title“ versehen.

**Programmfragment 9-1a:** Auszug aus dem Controller: Definition der Aktionen der Anwendung als HashTable-Aggregat

```
...
// Im Bieterkatalog bieten
ctrls.put("BieterkatalogBieten", par = new Hashtable());
  par.put("target", "Bieterkatalog.jsp");
  par.put("privileg", "2");
  par.put("title", "Spielzeugauktion - Bieterkatalog bearbeiten");
// Auktionskatalog anzeigen
ctrls.put("AuktionskatalogAnzeigen", par = new Hashtable());
  par.put("target", "Auktionskatalog.jsp");
  par.put("privileg", "0");
  par.put("title", "Spielzeugauktion - Auktionskatalog anzeigen");
...
```

2. Auf der Basis dieser Definitionen ist jeder Aktion ein Handlungsskript zugeordnet, das grundsätzlich zweierlei beinhaltet, nämlich
  - die Vorbereitung eines „Model“, d.h eines Datenmodells, das in einem weiteren Schritt an
  - die zuständige View („target“) weitergegeben wird.

So wird bei "BieterkatalogBieten".equals(ziel) in Programmfragment 9-1b das aktuelle Bieterobjekt beschafft, sodann ein Gebotsobjekt erzeugt, darin ein neues Gebot für den mit "snr" identifizierten Gegenstand abgegeben, sodann ein Katalogobjekt erzeugt und die Referenz auf diese Objekt für den gegebenen Bieter unter dem Namen "bieterkatalog" an das aktuelle session-Objekt gebunden.

Nach diesen Vorbereitungen wird die Kontrolle mit

```
request.getRequestDispatcher(target).forward(request, response);
```

an die „Target“-View übergeben.

**Programmfragment 9-1b:** Auszug aus dem Controller: Dispatcher-Anbschnitt

```

...
//===== Im Bieterkatalog bieten =====
if ("BieterkatalogBieten".equals(ziel)) {
    Bieter bieter = (Bieter) session.getAttribute("bieter");
    Gebot gebot = (Gebot) new Gebot();
    gebot.putGebot(bieter.bnr, Integer.parseInt(request.getParameter("snr")));
    Bieterkatalog bieterkatalog = new Bieterkatalog();
    session.setAttribute("bieterkatalog",
                        bieterkatalog.getBieterkatalog(bieter.bnr));
}

//===== Auktionskatalog anzeigen =====
if ("AuktionskatalogAnzeigen".equals(ziel)) {
    Auktionskatalog auktionskatalog = new Auktionskatalog();
    String[] kategorien = auktionskatalog.getKategorien();
    session.setAttribute("kategorien", kategorien);
    String kategorie = request.getParameter("kategorie") == null
        ? kategorien[0] : request.getParameter("kategorie");
    session.setAttribute("auktionskatalog",
                        auktionskatalog.getAuktionskatalog(kategorie));
}

...
//===== Forward! =====
request.getRequestDispatcher(target).forward(request, response);

```

In Programm 9-1 a, b ist der *vollständige Controller* aufgeführt – mit Lücken in den Controller-Hash-Tabellen und dem Dispatcher-Teil. Diese Lücken werden an den entsprechenden Stellen der Models ausgefüllt (Programme 9-1, 9-4, 9-5 ...).

**Programm 9-1a:** Controller, Konfigurationsteil

```

<!-- Spielzeugauktion Version 0.9 -->

<%@ page import="java.util.*, java.sql.*, auktion.*" %>
<%@ page session="true" %>
<%!
/* Controller: Dispatch-Tabelle
Identifizierung: Verweis auf Zielkenndaten, z.B.
Home -> (target -> Login.jsp, privileg -> 0 ...)
Die erkannte Tabelle wird im session-Objekt gespeichert.
*/
static Hashtable ctrls = new Hashtable();
static {
    Hashtable par; // temporär
    //===== Homepage mit Login
    ctrls.put("Home", par = new Hashtable()); // als Beispiel für die Folgeabschnitte
    par.put("target", "Login.jsp");
    par.put("privileg", "0");
    par.put("title", "Spielzeugauktion - Home");
}

```

```

//===== Homepage mit Login
...
//===== Seite mit Logoff
...
//===== Auktionskatalog anzeigen
...
//===== Bieterkatalog anzeigen
...
//===== in Bieterkatalog aufnehmen
...
//===== im Bieterkatalog löschen
...
//===== im Bieterkatalog bieten
...
//===== Artikel anzeigen
...
//===== Artikel bearbeiten
...
}
String[] loeschattliste = {"target", "privileg", "title", "quelle", "ziel"};
%>

```

### Programm 9-1b: Controller, insbesondere der Dispatcher-Abschnitt

```

<%
//===== Allgemeines =====
String quelle = (String) request.getParameter("quelle"); // Quelle
    quelle = quelle == null ? "Home" : quelle;
String ziel = (String) request.getParameter("ziel"); // Ziel
    ziel = ziel == null ? "Login" : ziel;
    session.setAttribute("ziel", ziel);
String uid = request.getParameter("uid");
String passwd = request.getParameter("passwd");
String target; // Zieldokument

for (int i = 0; i < loeschattliste.length; i++) // Ausgewählte Session-Attribute
    session.removeAttribute(loeschattliste[i]); // löschen
target = (String)((Hashtable) ctrls.get(ziel)).get("target");
Enumeration keys = ((Hashtable) ctrls.get(ziel)).keys(); // session-Attribute
while (keys.hasMoreElements()) {
    String tmp = (String) keys.nextElement();
    session.setAttribute(tmp, ((Hashtable) ctrls.get(ziel)).get(tmp));
}

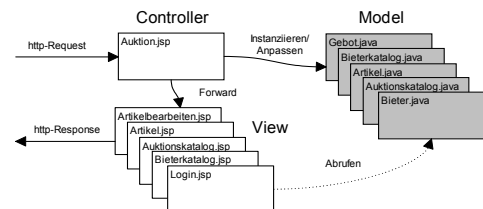
//===== Einloggen =====
if ("BieterkatalogAnzeigen".equals(ziel) && "Login".equals(quelle)) {
    session.setAttribute("bieter", new Bieter(uid, passwd));
}
//===== Ausloggen =====
...
//===== Zum Bieterkatalog (*muss* nach "Einloggen" stehen!) =====
...

```

```
//===== Im Bieterkatalog aufnehmen =====
...
//===== Im Bieterkatalog löschen =====
...
//===== Im Bieterkatalog bieten =====
...
//===== Auktionskatalog anzeigen =====
...
//===== Artikel anzeigen =====

//===== Artikel ändern =====
...
//===== Artikel einfügen =====
...
//===== Artikel löschen =====
...
//===== Forward! =====
request.getRequestDispatcher(target).forward(request, response);
%>
```

## 9.2.3 Die Models



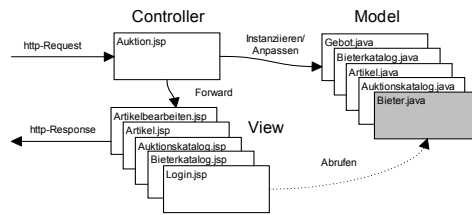
### 9.2.3.1 Verbindung.java

#### Programm 9-2: Klasse für die Verbindungsaufnahme mit der Datenbank

```
// ./WEB-INF/classes/auktion/Verbindung.java
package auktion;
import java.sql.*;

public class Verbindung {
    static Connection c;
    static boolean status;
    static {
        try {
            Class.forName("org.gjt.mm.mysql.Driver");
            c = DriverManager.getConnection("jdbc:mysql://localhost:3306/Spielzeugauktion");
            status = true;
        }
        catch (Exception ex) { status = false; }
    }
    public static Connection getConnection() { return c; }
}
```

### 9.2.3.2 Bieter.java



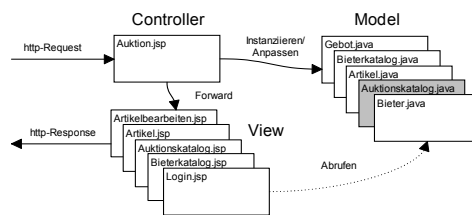
Programm 9-3: Die Klasse für das Bieter-Model

```
// ./WEB-INF/classes/auktion/Bieter.java
package auktion;
import java.sql.*;

public class Bieter {
    public boolean status;
    public int bnr, privileg;
    public String nachname, vorname, plz, ort, strasse, uid, passwd;

    public Bieter(String uid, String passwd) {
        try {
            Statement s = Verbindung.getConnection().createStatement();
            String sql = "SELECT * FROM Bieter WHERE uid LIKE '" + uid +
                "' AND passwd LIKE '" + passwd + "'";
            ResultSet rset = s.executeQuery(sql);
            if (rset.next()) {
                status = true;
                this.uid = uid;
                bnr = rset.getInt("bnr");
                nachname = rset.getString("nachname");
                vorname = rset.getString("vorname");
                plz = rset.getString("plz");
                ort = rset.getString("ort");
                strasse = rset.getString("strasse");
                passwd = rset.getString("passwd");
                privileg = rset.getInt("privileg");
            }
            else status = false;
        }
        catch (Exception ex) { status = false; }
    }
}
```

### 9.2.3.3 Auktionskatalog.java



**Programm 9-4: Das Model für den Auktionskatalog**

```

// ./WEB-INF/classes/auktion/Auktionskatalog.java
package auktion;
import java.sql.*;

public class Auktionskatalog {

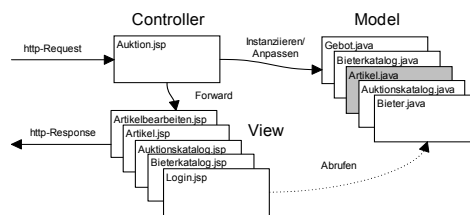
//=====
    public ResultSet getAuktionskatalog(String kategorie) {
        if (kategorie == null || kategorie == "") kategorie = "%";
        String sql;
        sql = "SELECT * FROM auktionskatalog ak " +
            "WHERE kategorie LIKE '" + kategorie + "' " + "ORDER BY ak.snr";
        try {
            Statement s = Verbindung.getConnection().createStatement();
            return s.executeQuery(sql);
        }
        catch (Exception ex) { return null; }
    }

//=====
    public ResultSet getArtikel(String snr) {
        if (snr == null || snr == "") snr = "%";
        String sql = "SELECT * FROM auktionskatalog WHERE snr LIKE '" + snr + "'";
        try {
            Statement s = Verbindung.getConnection().createStatement();
            return s.executeQuery(sql);
        }
        catch (Exception ex) { return null; }
    }

//=====
    public String[] getKategorien() {
        try {
            Statement s = Verbindung.getConnection().createStatement();
            ResultSet rs = s.executeQuery("SELECT COUNT(DISTINCT kategorie) AS anzahl,
                MIN(kategorie) AS kat FROM auktionskatalog");
            rs.next();
            int anzahl = rs.getInt("anzahl");
            String[] liste = new String[anzahl + 1];
            liste[0] = rs.getString("kat");
            String sql = "SELECT DISTINCT kategorie FROM auktionskatalog ORDER BY kategorie";
            ResultSet rset = s.executeQuery(sql);
            int i = 1;
            while (rset.next()) {
                liste[i++] = rset.getString("kategorie");
            }
            return liste;
        }
        catch (Exception ex) { return null; }
    }
}

```

### 9.2.3.4 Artikel.java



**Programm 9-5:** Model für einen einzelnen Auktionsgegenstand (Artikel)

```
// ./WEB-INF/classes/auktion/Artikel.java
package auktion;
import java.sql.*;

public class Artikel {
    public int snr, bieter;
    public String bis, gebotsdatum;
    public double limit, gebot;
    public String kategorie, beschreibung, bild;
    public boolean status;

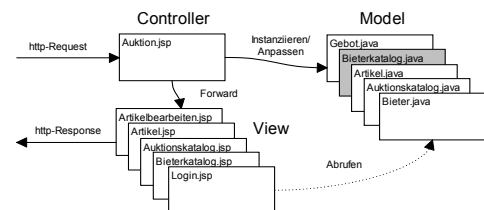
//=====
    public Artikel(String snr) {
        if (snr != null && snr != "") {
            String sql = "SELECT * FROM auktionskatalog WHERE snr LIKE '" + snr + "'";
            try {
                Statement s = Verbindung.getConnection().createStatement();
                ResultSet rset = s.executeQuery(sql);
                if (rset.next()) {
                    this.snr = rset.getInt("snr");
                    gebot = rset.getDouble("gebot");
                    limit = rset.getDouble("limit");
                    bieter = rset.getInt("bieter");
                    bis = rset.getString("bis");
                    gebotsdatum = rset.getString("gebotsdatum");
                }
            } catch (Exception ex) {}
        }
    }

//=====
    public boolean putArtikel(int snr) {
        this.snr = (snr <= 0) ? this.snr : snr;
        String sql = "UPDATE auktionskatalog SET gebot=" + gebot + ", `limit`=" + limit +
            ",bieter=" + bieter + ",bis='" + bis + "',gebotsdatum='" + gebotsdatum + "'" +
            ",kategorie='" + kategorie + "',beschreibung='" + beschreibung + "' " +
            ",bild='" + bild + "'" +
            " WHERE snr=" + this.snr;
        try {
            Statement s = Verbindung.getConnection().createStatement();
            s.executeUpdate(sql);
            return true;
        } catch (Exception ex) { return false; }
    }
}
```

```
//=====
public boolean newArtikel(String kategorie) {
    String sql = "INSERT INTO auktionskatalog(kategorie) VALUES('" + kategorie +
    "')";
    try {
        Statement s = Verbindung.getConnection().createStatement();
        s.executeUpdate(sql);
        return true;
    }
    catch (Exception ex) { return false; }
}

//=====
public boolean delArtikel(int snr) {
    if (snr > 0) {
        String sql = "DELETE FROM auktionskatalog WHERE snr=" + snr;
        try {
            Statement s = Verbindung.getConnection().createStatement();
            s.executeUpdate(sql);
            return true;
        }
        catch (Exception ex) { return false; }
    }
    else return false;
}
}
```

### 9.2.3.5 Bieterkatalog.java



#### Programm 9-6: Model-Klasse für den Bieterkatalog

```
// ./WEB-INF/classes/auktion/Bieterkatalog.java
package auktion;
import java.sql.*;

public class Bieterkatalog {
    public boolean status;

//=====
    public ResultSet getBieterkatalog(int bieter) {
        String sql = "SELECT * FROM auktionskatalog,bieterkataloge " +
            "WHERE snr=item AND bieterkataloge.bieter=" + bieter;
        try {
            Statement s = Verbindung.getConnection().createStatement();
            return s.executeQuery(sql);
        }
        catch (Exception ex) { return null; }
    }
}
```



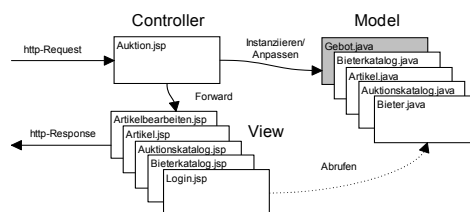
```
//=====
public void setItem(Bieter bieter, String item) {
    String sql = "INSERT INTO bieterkataloge(bieter,item) " +
        "VALUES(" + bieter.bnr + "," + item + ")";

    try {
        Statement s = Verbindung.getConnection().createStatement();
        s.executeUpdate(sql);
        status = true;
    }
    catch (Exception ex) { status = false; }
}

//=====
public void delItem(Bieter bieter, String item) {
    String sql = "DELETE FROM bieterkataloge " +
        "WHERE bieter=" + bieter.bnr + " AND item=" + item;

    try {
        Statement s = Verbindung.getConnection().createStatement();
        s.executeUpdate(sql);
        status = true;
    }
    catch (Exception ex) { status = false; }
}
}
```

### 9.2.3.6 Gebot.java



#### Programm 9-7: Model für die Gebotsabgabe

```
// ./WEB-INF/classes/auktion/Gebot.java
package auktion;
import java.sql.*;
import java.util.*;

public class Gebot {
    public boolean putGebot(int bieter, int item) {
        String sql = "SELECT * FROM auktionskatalog WHERE snr=" + item;
        try {
            Statement s = Verbindung.getConnection().createStatement();
            ResultSet rset = s.executeQuery(sql);
            if (rset.next()) {
```

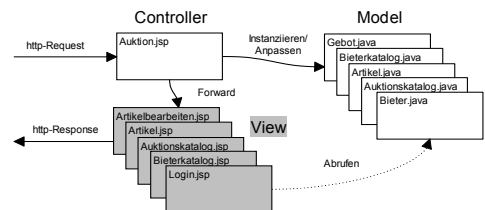
```

double limit = rset.getDouble("limit");
Calendar bis = Calendar.getInstance();
bis.setTime(rset.getTimestamp("bis"));
Calendar lst = Calendar.getInstance();
lst.setTime(rset.getTimestamp("gebotsdatum"));
Calendar akt = Calendar.getInstance();
akt.setTime(new java.util.Date());
long delay = akt.getTime().getTime() - lst.getTime().getTime();
if (akt.getTime().getTime() > bis.getTime().getTime() && delay/1000 > 5*60){
    return false;
}
double gebot, vorgebot = rset.getDouble("gebot");
if (limit < 1.0) limit = 1.0;           // Kleinstes Limit ist 1.0 Euro
if (vorgebot < limit)
    gebot = limit;           // Beginn bei Limit (1. Gebot)
else
    gebot = Math.ceil(vorgebot * 1.025); // 2,5% mehr und nächste ganze Mark

String datum = "" + akt.get(akt.YEAR) + "-" +
                (akt.get(akt.MONTH)+1) + "-" + akt.get(akt.DAY_OF_MONTH);
datum+=" " + akt.get(akt.HOUR_OF_DAY) + ":" +
            akt.get(akt.MINUTE) + ":" + akt.get(akt.SECOND);
sql = "UPDATE auktionskatalog SET gebot=" + gebot +
      ",gebotsdatum=" + datum + " " +
      ",bieter=" + bieter + " " +
      "WHERE snr=" + item;
s = Verbindung.getConnection().createStatement();
s.executeUpdate(sql);
}
return true;
}
catch (Exception ex) { return false; }
}
}

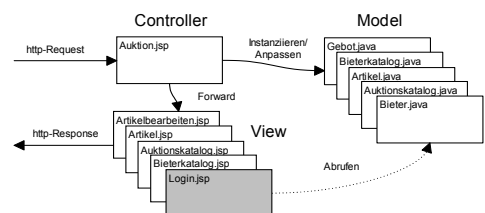
```

## 9.2.4 Die Views



### 9.2.4.1 Login.jsp

Konfigurations- und Dispatch-Abschnitte  
im Controller Auktion.jsp



## Die Einträge in Auswahl.jsp, Konfigurations- und Steuer- bzw. Dispatch-Abschnitt

```

...
// Homepage mit Login
ctrls.put("Home", par = new Hashtable());
par.put("target", "Login.jsp");
par.put("privileg", "0");
par.put("title", "Spielzeugauktion - Home");
// Homepage mit Login
ctrls.put("Login", par = new Hashtable());
par.put("target", "Login.jsp");
par.put("privileg", "0");
par.put("title", "Spielzeugauktion - Login");
// Seite mit Logoff
ctrls.put("Logoff", par = new Hashtable());
par.put("target", "Login.jsp");
par.put("privileg", "2");
par.put("title", "Spielzeugauktion - Logoff");
...

```

```

...
//===== Einloggen =====
if ("BieterkatalogAnzeigen".equals(ziel) && "Login".equals(quelle)) {
    session.setAttribute("bieter", new Bieter(uid, passwd));
}
//===== Ausloggen =====
if ("Logoff".equals(ziel)) {
    session.removeAttribute("bieter");
    target = (String)((Hashtable) ctrls.get(quelle)).get("target");
}
...

```

Programm 9-8: View des Login/Logoff-Teiles

```

<%-- ./Login.jsp --%>
<%@ page import="java.util.*, auktion.*" %>
<% String quelle = "Login";
String uid = request.getParameter("uid");
session.setAttribute("quelle", "Login");
session.setAttribute("ziel", "BieterkatalogAnzeigen");
%>

```

```

<TITLE><%= session.getAttribute("title") %></TITLE>
<BODY Background=bilder/hintergrund.gif Bgcolor=#6666ff>
<H1 Align=center><%= session.getAttribute("title") %></H1>
<DIV Align=center><%=
    Bieter bieter = (Bieter) session.getAttribute("bieter");
    if (bieter != null && bieter.status)
        out.print(bieter.vorname + " " + bieter.nachname +
            " (" + bieter.bnr + "/" + bieter.privileg + ")");
    else
        out.print("Kein Bieter!");
%></DIV><P>

<TABLE Border=1 Cellpadding=15 Cellspacing=0 Align=center><TR Align=center><TD>
<FORM Method=GET Action=Auktion.jsp>Bieter-Id<BR>
<INPUT Type=text Name=uid Value="<%= uid == null ? "" : uid %>"><BR>Passwort<BR>
<INPUT Type=hidden Name=quelle Value="Login">
<INPUT Type=hidden Name=ziel Value="BieterkatalogAnzeigen">
<INPUT Type=password Name=passwd><P>
<INPUT Type=submit Value="Login!">
</FORM></TD>

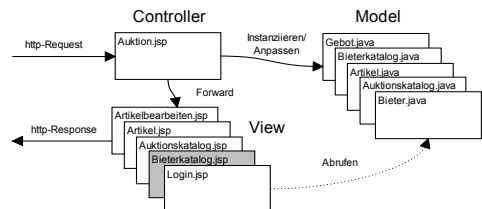
<TD>
<FORM Method=GET Action=Auktion.jsp>
<INPUT Type=submit Value="Zum Auktionskatalog"><BR>&nbsp;<BR>
<INPUT Type=hidden Name=quelle Value="Login">
<INPUT Type=hidden Name=ziel Value="AuktionskatalogAnzeigen">
</FORM></TD>

<TD>
<FORM Method=GET Action=Auktion.jsp>
<INPUT Type=submit Value="Logoff!"><BR>&nbsp;<BR>
<INPUT Type=hidden Name=quelle Value="Login">
<INPUT Type=hidden Name=ziel Value="Logoff">
</FORM></TD></TR>
</TABLE>

```

### 9.2.4.2 Bieterkatalog.jsp

Konfigurations- und Dispatch-Abschnitte  
im Controller Auktion.jsp



```

...
// Bieterkatalog anzeigen
ctrls.put("BieterkatalogAnzeigen", par = new Hashtable());
  par.put("target", "Bieterkatalog.jsp");
  par.put("privileg", "2");
  par.put("title", "Spielzeugauktion - Bieterkatalog bearbeiten");
// in Bieterkatalog aufnehmen
ctrls.put("BieterkatalogAufnehmen", par = new Hashtable());
  par.put("target", "Bieterkatalog.jsp");
  par.put("privileg", "2");
  par.put("title", "Spielzeugauktion - Bieterkatalog bearbeiten");
// im Bieterkatalog löschen
ctrls.put("BieterkatalogLoeschen", par = new Hashtable());
  par.put("target", "Bieterkatalog.jsp");
  par.put("privileg", "2");
  par.put("title", "Spielzeugauktion - Bieterkatalog bearbeiten");
// im Bieterkatalog bieten
ctrls.put("BieterkatalogBieten", par = new Hashtable());
  par.put("target", "Bieterkatalog.jsp");
  par.put("privileg", "2");
  par.put("title", "Spielzeugauktion - Bieterkatalog bearbeiten");
...

```

```

...
//===== Zum Bieterkatalog (*muss* nach "Einloggen" stehen!) =====
if ("BieterkatalogAnzeigen".equals(ziel)) {
  Bieter bieter = (Bieter) session.getAttribute("bieter");
  if (bieter == null || !bieter.status || bieter.privileg < 2) {
    session.setAttribute("ziel", "Login");
    target = (String)((Hashtable) ctrls.get("Login")).get("target");
    session.setAttribute("title",
      (String)((Hashtable) ctrls.get("Login")).get("title"));
  }
  else {
    Bieterkatalog bieterkatalog = new Bieterkatalog();
    session.setAttribute("bieterkatalog",
      bieterkatalog.getBieterkatalog(bieter.bnr));
  }
}
...

```

```

//===== Im Bieterkatalog aufnehmen =====
if ("BieterkatalogAufnehmen".equals(ziel)) {
  Bieter bieter = (Bieter) session.getAttribute("bieter");
  Bieterkatalog bieterkatalog = new Bieterkatalog();
  bieterkatalog.setItem(bieter, request.getParameter("snr"));
  session.setAttribute("bieterkatalog",
    bieterkatalog.getBieterkatalog(bieter.bnr));
}

```

```
//===== Im Bieterkatalog löschen =====
if ("BieterkatalogLoeschen".equals(ziel)) {
    Bieter bieter = (Bieter) session.getAttribute("bieter");
    Bieterkatalog bieterkatalog = new Bieterkatalog();
    bieterkatalog.delItem(bieter, request.getParameter("snr"));
    session.setAttribute("bieterkatalog",
        bieterkatalog.getBieterkatalog(bieter.bnr));
}

//===== Im Bieterkatalog bieten =====
if ("BieterkatalogBieten".equals(ziel)) {
    Bieter bieter = (Bieter) session.getAttribute("bieter");
    Gebot gebot = (Gebot) new Gebot();
    gebot.putGebot(bieter.bnr, Integer.parseInt(request.getParameter("snr")));
    Bieterkatalog bieterkatalog = new Bieterkatalog();
    session.setAttribute("bieterkatalog",
        bieterkatalog.getBieterkatalog(bieter.bnr));
}
...

```

Heinrich Karl Schneider Richard Wagner Allee 2 69355 Schriesheim					UID: zwei Bieter Nr./Privileg: 3/2 20.07.2003 16:44:15	
Bieter	Termin	Limit	Gebot	Kategorie	Beschreibung	
109	21.08.2003 12:00:00	1900.00	1948.00 €	Auto	MÄRKLIN, Baukastenauto, Pullmann- Uhrwerk, sehr guter Orig.-Zust.	
<input type="button" value="Bieten"/>			Bieter 4	<input type="button" value="Anschauen!"/>		
<input type="button" value="Passen"/>						
115	21.08.2003 12:00:00	1500.00	1500.00 €	Puppe	Biskuitporzellan-Puppe, Frankreich, D Kurbelkopf, blaue festeingesetzte Glas Ohrklappchen, franz. Gliederkörper, alt neuer, kleiner Haarriss an der Schläfe, t	
<input type="button" value="Bieten"/>			Bieter 3	<input type="button" value="Anschauen!"/>		
<input type="button" value="Passen"/>						

**Programm 9-9:** View des Bieterkatalogs

```
<!-- ./Bieterkatalog.jsp -->
<STYLE> FORM { margin: 5pt; padding: 0pt } </STYLE>
<%@ page import="java.sql.*, java.util.*, auktion.*" %>
<%!
String toDatum(java.util.Date dat) {
    Calendar d = Calendar.getInstance();
    d.setTime(dat);
    return zwei("0" + d.get(d.DAY_OF_MONTH)) + "." +
        zwei("0" + d.get(d.MONTH)) + "." + d.get(d.YEAR) + " " +
        zwei("0" + d.get(d.HOUR_OF_DAY)) + ":" +
        zwei("0" + d.get(d.MINUTE)) + ":" + zwei("0" + d.get(d.SECOND));
}
String zwei(String s) { return s.substring(s.length() - 2, s.length()); }
%><%
Bieter b = ((Bieter) session.getAttribute("bieter"));
ResultSet rset = ((ResultSet) session.getAttribute("bieterkatalog"));
%>

```

```

<TITLE><%= session.getAttribute("title") %></TITLE>
<BODY Background=bilder/hintergrund.gif>
<H1 Align=center><%= session.getAttribute("title") %></H1>
<DIV Align=center>Bieter: <%
    Bieter bieter = (Bieter) session.getAttribute("bieter");
    if (bieter != null && bieter.status)
        out.print(bieter.vorname + " " + bieter.nachname +
            " (" + bieter.bnr + "/" + bieter.privileg + ")");
    else
        out.print("Kein Bieter!");
%></DIV><P>

<TABLE Border=1 Cellspacing=0 Align=center Width=100%>
  <TR Valign=top>
    <TD Colspan=5><FORM>
      <INPUT Type=Submit Value='Aktualisieren! '>
      <INPUT Type=hidden Name=kategorie
          Value='<%= request.getParameter("kategorie") %>'>
      <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
      <INPUT Type=hidden Name=ziel Value='BieterkatalogAnzeigen'>
    </FORM></TD>
    <TD><FORM>
      <INPUT Type=Submit Value='Zum Auktionskatalog! '>
      <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
      <INPUT Type=hidden Name=ziel Value='AuktionskatalogAnzeigen'>
    </FORM></TD></TR>
  <TR Valign=top><TD Colspan=5><%= b.vorname + " " + b.nachname %>
      <BR><%= b.strasse %><BR><%= b.plz + " " + b.ort %></TD>
    <TD>UID: <%= b.uid %><BR>Bieter-Nr./Privileg: <%= b.bnr + "/" + b.privileg %>
      <BR><%= toDatum(new java.util.Date()) %></TD></TR>
  <TR><TD>Bieten</TD><TD>Termin</TD><TD>Limit</TD><TD>Gebot</TD>
    <TD>Kategorie</TD><TD Align=left>Beschreibung</TD></TR>

  <% while (rset.next()) { %>
    <TR Valign=top><TD Align=center><B><%= rset.getInt("snr") %></B>
      <FORM><INPUT Type=submit Value=Bieten>
        <INPUT Type=hidden Name=snr Value='<%= rset.getInt("snr") %>'>
        <INPUT Type=hidden Name=quelle
            Value='<%= request.getParameter("ziel") %>'>
        <INPUT Type=hidden Name=ziel Value='BieterkatalogBieten'>
      </FORM>
      <FORM><INPUT Type=submit Value=Passen>
        <INPUT Type=hidden Name=snr Value='<%= rset.getInt("snr") %>'>
        <INPUT Type=hidden Name=quelle
            Value='<%= request.getParameter("ziel") %>'>
        <INPUT Type=hidden Name=ziel Value='BieterkatalogLoeschen'>
      </FORM></TD>

```

```

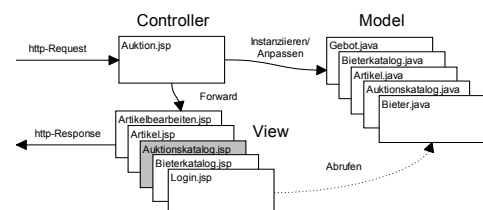
<TD Align=center><%= toDatum(rset.getTimestamp("bis")) %></TD>
<TD Align=right><%= rset.getString("limit") %></TD>
<TD Align=right><B><%= rset.getString("gebot") %>&nbsp;&euro;</B>
<BR><BR>Bieter <%= rset.getString("Auktionskatalog.bieter") %></TD>
<TD Align=center><%= rset.getString("kategorie") %>
<FORM><INPUT Type=submit Value=Anschau!>
<INPUT Type=hidden Name=snr Value='<%= rset.getInt("snr") %>'>
<INPUT Type=hidden Name=kategorie
Value='<%= rset.getString("kategorie") %>'>
<INPUT Type=hidden Name=quelle
Value='<%= request.getParameter("ziel") %>'>
<INPUT Type=hidden Name=ziel Value='ArtikelAnzeigen'>
</FORM></TD>
<TD><%= rset.getString("beschreibung") %></TD></TR>
<% } %>
</TABLE>

<TABLE Align=center><TR><TD><BR>&nbsp;&nbsp;&nbsp;<BR>
<FORM Action=Auktion.jsp Method=GET>
<INPUT Type=submit Value="Home">
<INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
<INPUT Type=hidden Name=ziel Value='Home'>
</FORM>
</TD></TR></TABLE>

```

### 9.2.4.3 Auktionskatalog.jsp

Konfigurations- und Dispatch-Abschnitte  
im Controller Auktion.jsp



```

...
// Auktionskatalog anzeigen
ctrls.put("AuktionskatalogAnzeigen", par = new Hashtable());
par.put("target", "Auktionskatalog.jsp");
par.put("privileg", "0");
par.put("title", "Spielzeugauktion - Auktionskatalog anzeigen");
...

```

```

...
//===== Auktionskatalog anzeigen =====
if ("AuktionskatalogAnzeigen".equals(ziel)) {
    Auktionskatalog auktionskatalog = new Auktionskatalog();
    String[] kategorien = auktionskatalog.getKategorien();
    session.setAttribute("kategorien", kategorien);
    String kategorie = request.getParameter("kategorie") == null ? kategorien[0] :
        request.getParameter("kategorie");
    session.setAttribute("auktionskatalog",
        auktionskatalog.getAuktionskatalog(kategorie));
}
...

```



Zum Bieterkatalog			
Termin	Limit	Gebot	Kategorie
Pos. 105 21.08.2003 12:00:00	600.00	600.00 €	Auto
			Anschau!
Pos. 106 21.08.2003 12:00:00	1900.00	1900.00 €	Auto
			Anschau!

### Programm 9-10: Die View des Auktionskataloges

```

<!-- ./Auktionskatalog.jsp -->
<STYLE> FORM { margin: 5pt; padding: 0pt } </STYLE>
<%@ page import="java.sql.*, java.util.*, auktion.*" %>
<%!
String toDatum(java.util.Date dat) {
    Calendar d = Calendar.getInstance();
    d.setTime(dat);
    return zwei("0" + d.get(d.DAY_OF_MONTH)) + "." +
        zwei("0" + d.get(d.MONTH)) + "." + d.get(d.YEAR) + " " +
        zwei("0" + d.get(d.HOUR_OF_DAY)) + ":" +
        zwei("0" + d.get(d.MINUTE)) + ":" + zwei("0" + d.get(d.SECOND));
}
String zwei(String s) { return s.substring(s.length() - 2, s.length()); }
%>

<TITLE><%= session.getAttribute("title") %></TITLE>
<BODY Background=bilder/hintergrund.gif>
<H1 Align=center><%= session.getAttribute("title") %></H1>
<DIV Align=center>Bieter: <%
    Bieter bieter = (Bieter) session.getAttribute("bieter");
    if (bieter != null && bieter.status)
        out.print(bieter.vorname + " " + bieter.nachname +
            " (" + bieter.bnr + "/" + bieter.privileg + ")");
    else
        out.print("Kein Bieter!");
%></DIV><P>

```

```

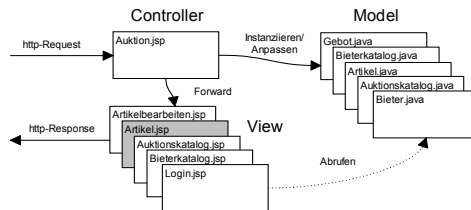
<TABLE Align=center Border=1 Cellspacing=0 Width=100%>
<TR><TD Colspan=4>
  <FORM><INPUT Type=submit Value="Zum Bieterkatalog">
    <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
    <INPUT Type=hidden Name=ziel Value='BieterkatalogAnzeigen'>
  </FORM></TD>
<TD><FORM><SELECT Name=kategorie onchange=submit()><%
String[] kats = (String[]) session.getAttribute("kategorien");
for (int i = 1; i < kats.length; i++) {
  String kat = request.getParameter("kategorie") == null ? kats[0] :
    request.getParameter("kategorie");

  if (kats[i].equals(kat))
    out.print("\n<OPTION Selected Value=" + kats[i] + ">" + kats[i]);
  else
    out.print("\n<OPTION Value=" + kats[i] + ">" + kats[i]);
}
%></SELECT>
  <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
  <INPUT Type=hidden Name=ziel Value='AuktionskatalogAnzeigen'>
</FORM></TD></TR>
<TR><TH>Termin</TH><TH>Limit</TH><TH>Gebot</TH><TH>Kategorie</TH>
  <TH>Beschreibung</TH></TR>
<%
ResultSet rset = (ResultSet) session.getAttribute("auktionskatalog");
while (rset != null && rset.next()) {
  Calendar cbis = Calendar.getInstance();
  cbis.setTime(rset.getTimestamp("bis"));
  Calendar cakt = Calendar.getInstance(); cakt.setTime(new java.util.Date());
  long bis = cbis.getTime().getTime();
  long akt = cakt.getTime().getTime();
%><TR Valign=top>
  <TD Align=center><%= akt < bis ? "<B>" : "" %>Pos.
    <%= rset.getInt("snr") %><BR>
    <%= toDatum(cbis.getTime()) %></TD>
  <TD Align=right><%= rset.getDouble("limit") %>0</TD>
  <TD Align=right><%= akt < bis ? "<B>" : "" %>
    <%= rset.getDouble("gebot") %>0&nbsp;&euro;</B></TD>
  <TD Align=center><%= rset.getString("kategorie") %>
    <FORM><INPUT Type=submit Value=Anschauen!>
    <INPUT Type=hidden Name=snr Value='<%= rset.getInt("snr") %>'>
    <INPUT Type=hidden Name=kategorie
      Value='<%= rset.getString("kategorie") %>'>
    <INPUT Type=hidden Name=quelle
      Value='<%= request.getParameter("ziel") %>'>
    <INPUT Type=hidden Name=ziel Value='ArtikelAnzeigen'>
    </FORM></TD>
  <TD><%= rset.getString("beschreibung") %></TD></TR>
<% } %>
</TABLE>

```

```
<TABLE Align=center><TR><TD><BR>&nbsp;<BR>
<FORM Action=Auktion.jsp Method=GET>
  <INPUT Type=submit Value="Home">
  <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
  <INPUT Type=hidden Name=ziel Value='Home'>
</FORM>
</TD></TR></TABLE>
```

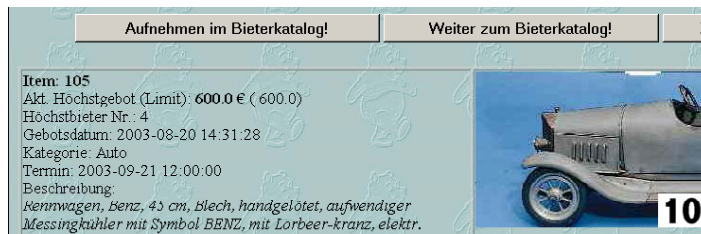
### 9.2.4.4 Artikel.jsp



Konfigurations- und Dispatch-Abschnitte im Controller Auktion.jsp

```
...
// Artikel anzeigen
ctrls.put("ArtikelAnzeigen", par = new Hashtable());
par.put("target", "Artikel.jsp");
par.put("privileg", "6");
par.put("title", "Spielzeugauktion - Artikel anzeigen");
...
```

```
//===== Artikel anzeigen =====
if ("ArtikelAnzeigen".equals(ziel)) {
  Auktionskatalog auktionskatalog = new Auktionskatalog();
  session.setAttribute("artikel", auktionskatalog.getArtikel(request.getParame-
ter("snr")));
}
...
```



**Programm 9-11: View für individuelle Artikel aus dem Versteigerungskatalog**

```

<%-- Artikel.jsp --%>
<%@ page import="java.sql.*, java.util.*, auktion.*" %>

<TITLE><%= session.getAttribute("title") %></TITLE>
<BODY Background=bilder/hintergrund.gif>
<H1 Align=center><%= session.getAttribute("title") %></H1>
<DIV Align=center><%
    Bieter bieter = (Bieter) session.getAttribute("bieter");
    if (bieter != null && bieter.status)
        out.print(bieter.vorname + " " + bieter.nachname +
            " (" + bieter.bnr + "/" + bieter.privileg + ")");
    else
        out.print("Kein Bieter!");
%></DIV><P>

<TABLE Align=center><TR>
<% if (!"BieterkatalogAnzeigen".equals(request.getParameter("quelle")) &&
    !"BieterkatalogAufnehmen".equals(request.getParameter("quelle")) &&
    bieter != null && bieter.status) { %>
<TD><FORM>
    <INPUT Type=Submit Value="Aufnehmen im Bieterkatalog!">
    <INPUT Type=hidden Name=snr Value='<%= request.getParameter("snr") %>'>
    <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
    <INPUT Type=hidden Name=ziel Value='BieterkatalogAufnehmen'>
</FORM></TD>
<% } %>

<TD><FORM>
    <INPUT Type=Submit Value="Weiter zum Bieterkatalog!">
    <INPUT Type=hidden Name=snr Value='<%= request.getParameter("snr") %>'>
    <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
    <INPUT Type=hidden Name=ziel Value='BieterkatalogAnzeigen'>
</FORM></TD>

<TD><FORM>
    <INPUT Type=Submit Value="Zum Auktionskatalog!">
    <INPUT Type=hidden Name=kategorie
        Value='<%= request.getParameter("kategorie") %>'>
    <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
    <INPUT Type=hidden Name=ziel Value='AuktionskatalogAnzeigen'>
</FORM></TD></TR>
</TABLE>

```

```

<%
  ResultSet rset = (ResultSet) session.getAttribute("artikel");
  rset.next();
%>
<TABLE Border=1><TR Valign=top><TD>
  <B>Item: <%= rset.getInt("snr") %></B><BR>
  Akt. Höchstgebot (Limit): <B><%= rset.getDouble("gebot") %> &euro;</B> (
    <%= rset.getDouble("limit") %>)<BR>
  Höchstbieter Nr.: <%= rset.getInt("bieter") %><BR>
  Gebotsdatum: <%= rset.getString("gebotsdatum") %><BR>
  Kategorie: <%= rset.getString("kategorie") %><BR>
  Termin: <%= rset.getString("bis") %><BR>
  Beschreibung:<BR><I><%= rset.getString("beschreibung") %></I></TD>
<TD><IMG Src='<%= rset.getString("bild") %>'></TD></TR>
</TABLE>

```

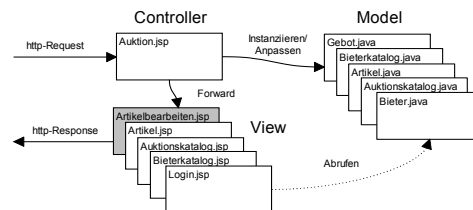
```

<TABLE Align=center Cellpadding=30><TR><TD>
<FORM Action=Auktion.jsp Method=GET>
  <INPUT Type=submit Value="Home">
  <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
  <INPUT Type=hidden Name=ziel Value='Home'>
</FORM></TD><TD>
<% if (bieter != null && bieter.privileg >= 4) { %>
<FORM Action=Auktion.jsp Method=GET>
  <INPUT Type=submit Value="Artikel Bearbeiten!">
  <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
  <INPUT Type=hidden Name=ziel Value='ArtikelAendern'>
  <INPUT Type=hidden Name=snr Value='<%= request.getParameter("snr") %>'>
  <INPUT Type=hidden Name=kategorie
    Value='<%= request.getParameter("kategorie") %>'>
</FORM>
<% } %>
</TD></TR></TABLE>

```

### 9.2.4.5 Artikelbearbeiten.jsp

Konfigurations- und Dispatch-Abschnitte im Controller Auktion.jsp



```

...
// Artikel bearbeiten
ctrls.put("ArtikelAendern", par = new Hashtable());
par.put("target", "Artikelbearbeiten.jsp");
par.put("privileg", "4");
par.put("title", "Spielzeugauktion - Artikel &auml;ndern");
ctrls.put("ArtikelLoeschen", par = new Hashtable());
par.put("target", "Artikelbearbeiten.jsp");
par.put("privileg", "4");
par.put("title", "Spielzeugauktion - Artikel l&ouml;schen");
ctrls.put("ArtikelNeu", par = new Hashtable());
par.put("target", "Artikelbearbeiten.jsp");
par.put("privileg", "4");
par.put("title", "Spielzeugauktion - Artikel einf&uuml;gen");
...

```

```

...
//==== Artikel ändern =====
if ("ArtikelAendern".equals(ziel)) {
    Artikel item = new Artikel(request.getParameter("snr"));
    session.setAttribute("item", item);
}

//==== Artikel einfügen =====
if ("ArtikelNeu".equals(ziel)) {
    Artikel item = (Artikel) session.getAttribute("item");
    item.newArtikel(item.kategorie);
    session.setAttribute("ziel", "AuktionskatalogAnzeigen");
    target =
        (String)((Hashtable) ctrls.get("AuktionskatalogAnzeigen")).get("target");
    session.setAttribute("title",
        (String)((Hashtable) ctrls.get("AuktionskatalogAnzeigen")).get("title"));
    Auktionskatalog auktionskatalog = new Auktionskatalog();
    session.setAttribute("auktionskatalog",
        auktionskatalog.getAuktionskatalog(item.kategorie));
}

//==== Artikel löschen =====
if ("ArtikelLoeschen".equals(ziel)) {
    Artikel item = (Artikel) session.getAttribute("item");
    item.delArtikel(item.snr);
    target =
        (String)((Hashtable) ctrls.get("AuktionskatalogAnzeigen")).get("target");
    session.setAttribute("title",
        (String)((Hashtable) ctrls.get("AuktionskatalogAnzeigen")).get("title"));
    Auktionskatalog auktionskatalog = new Auktionskatalog();
    session.setAttribute("auktionskatalog",
        auktionskatalog.getAuktionskatalog(item.kategorie));
}
...

```

	Rettungsboote auf jeder Seite, 2 Masten, Fahne, mit Schlüssel, unbespielter Orig.-Zust., Uhrwerk intakt
Höchstbietender:	0
Akt. Höchstgebot:	0.0 €
Gebotsdatum:	2003-08-20 10:11:28
<input type="button" value="Artikel ändern!"/>	
<input type="button" value="Home"/> <input type="button" value="Neuer Artikel!"/> <input type="button" value="Artikel löschen!"/>	

**Programm 9-12:** View für die Bearbeitung eines individuellen Artikels

```

<!-- Artikelbearbeiten.jsp -->
<%@ page import="java.sql.*, java.util.*, auktion.*" %>

<TITLE><%= session.getAttribute("title") %></TITLE>
<BODY Background=bilder/hintergrund.gif>
<H1 Align=center><%= session.getAttribute("title") %></H1>
<DIV Align=center><%
    Bieter bieter = (Bieter) session.getAttribute("bieter");
    if (bieter != null && bieter.status)
        out.print(bieter.vorname + " " + bieter.nachname +
            " (" + bieter.bnr + "/" + bieter.privileg + ")");
    else
        out.print("Kein Bieter!");
%></DIV><P>

<TABLE Align=center><TR>
<TD><FORM>
    <INPUT Type=Submit Value="Weiter zum Bieterkatalog!">
    <INPUT Type=hidden Name=snr Value='<%= request.getParameter("snr") %>'\>
    <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'\>
    <INPUT Type=hidden Name=ziel Value='BieterkatalogAnzeigen'\>
</FORM></TD>
<TD><FORM>
    <INPUT Type=Submit Value="Zum Auktionskatalog!">
    <INPUT Type=hidden Name=kategorie
        Value='<%= request.getParameter("kategorie") %>'\>
    <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'\>
    <INPUT Type=hidden Name=ziel Value='AuktionskatalogAnzeigen'\>
</FORM></TD></TR>
</TABLE>

```

```

<%
Artikel item = (Artikel) session.getAttribute("item");
if (request.getParameter("quelle").equals(request.getParameter("ziel")) &&
    request.getParameter("quelle").equals("ArtikelAendern")) {
    item.bis = request.getParameter("bis");
    item.limit = Double.parseDouble(request.getParameter("limit"));
    item.kategorie = request.getParameter("kategorie");
    item.bild = request.getParameter("bild");
    item.beschreibung = request.getParameter("beschreibung");
    item.bieter = Integer.parseInt(request.getParameter("bieter"));
    item.gebot = Double.parseDouble(request.getParameter("gebot"));
    item.gebotsdatum = request.getParameter("gebotsdatum");
    item.putArtikel(-1);
} %>

<% if (item != null) { %>
<TABLE Border=1 Align=center Cellspacing=0>
<FORM Action=Auktion.jsp Method=GET>
  <TR Valign=top><TD><B>Item: </TD>
    <TD><B><INPUT Type=text Name=snr Value='<%= item.snr %>' readonly</TD></TR>
  <TR Valign=top><TD>Schlusstermin: </TD>
    <TD><INPUT Type=text Name=bis Value='<%= item.bis %>'></TD></TR>
  <TR Valign=top><TD>Limit: </TD>
    <TD><INPUT Type=text Name=limit Value='<%= item.limit %>'></TD></TR>
  <TR Valign=top><TD>Kategorie: </TD>
    <TD><INPUT Type=text Name=kategorie Value='<%= item.kategorie %>'></TD></TR>
  <TR Valign=top><TD>Bild-URL</TD>
    <TD><INPUT Type=text Name=bild Value='<%= item.bild %>' Size=60</TD></TR>
  <TR Valign=top><TD>Beschreibung:</TD>
    <TD><TEXTAREA Cols=60 Rows=5 Name=beschreibung>
      <%= item.beschreibung %></TEXTAREA></I></TD></TR>
  <TR Valign=top><TD>H&ouml;chstbietender: </TD>
    <TD><INPUT Type=text Name=bieter Value='<%= item.bieter %>'></TD></TR>
  <TR Valign=top><TD>Akt. H&ouml;chstgebot:</TD>
    <TD><INPUT Type=text Name=gebot Value='<%= item.gebot %>'> &euro;</TD></TR>
  <TR Valign=top><TD>Gebotsdatum: </TD>
    <TD><INPUT Type=text Name=gebotsdatum
      Value='<%= item.gebotsdatum %>'></TD></TR>
  <TR><TD Colspan=2 Align=center><BR>
    <INPUT Type=submit Value="Artikel &auml;ndern!">
    <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>'>
    <INPUT Type=hidden Name=ziel Value='ArtikelAendern'></TD></TR>
</FORM></TABLE>
<% } %>

```



```

<TABLE Align=center Cellpadding=20><TR><TD>
<FORM Action=Auktion.jsp Method=GET>
  <INPUT Type=submit Value="Home">
  <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>''>
  <INPUT Type=hidden Name=ziel Value='Home''>
</FORM></TD>
<TD>
<FORM Action=Auktion.jsp Method=GET>
  <INPUT Type=submit Value="Neuer Artikel!">
  <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>''>
  <INPUT Type=hidden Name=ziel Value='ArtikelNeu''>
  <INPUT Type=hidden Name=kategorie Value='<%= item.kategorie %>''>
</FORM></TD><TD>
<FORM Action=Auktion.jsp Method=GET>
  <INPUT Type=submit Value="Artikel löschen!">
  <INPUT Type=hidden Name=quelle Value='<%= request.getParameter("ziel") %>''>
  <INPUT Type=hidden Name=ziel Value='ArtikelLoeschen''>
  <INPUT Type=hidden Name=snr Value='<%= item.snr %>''>
  <INPUT Type=hidden Name=kategorie Value='<%= item.kategorie %>''>
</FORM></TD></TR></TABLE>

```

## 9.3 Übungsaufgaben

### Aufgabe 9-1: Exception und Stack-Trace

Mittels der Anweisungsfolge

```

StringWriter out = new StringWriter();
ex.printStackTrace(new PrintWriter(out));
String strace = out.toString()

```

lassen sich Ursache und Stack-Trace einer Exception in einen String wandeln.

Modifizieren Sie `Bieter.java` (class `Bieter`) so, dass zusätzlich zum zweiwertigen Status `status` noch die Fehlerursache mitsamt eines Stack-Traces angezeigt wird. Weiter ist eine Fehlerbehandlung an den Stellen der Views, die diese Klasse verwenden, vorzunehmen.

### Aufgabe 9-2: Verbindungsüberwachung

Bauen Sie in die Klasse `Verbindung` (`Verbindung.java`) einen Überwacher (Watchdog) ein, mittels dessen die Verbindung in regelmäßigen Zeitabständen überprüft und gegebenenfalls die Verbindung wiederhergestellt wird. (Threads sind z.B. in Abschnitt 5.3 des Buches beispielhaft verwendet.)

**Aufgabe 9-3: Anbieter von Versteigerungsgegenständen**

Ein Auktionator versteigert Artikel ausschließlich im Auftrag.

Erweitern Sie die Datenbank um eine Tabelle „Anbieter“, in der die Kenndaten der Auftraggeber von Versteigerungsgegenständen festgehalten sind.

Wie muss die Tabelle „Auktionskatalog“ ergänzt werden, um jeden Artikel einem Auftraggeber zuordnen zu können?

**Aufgabe 9-4: Bieter/Anbieter-Kommunikation nach Zuschlag**

Modifizieren Sie die Anwendung einschließlich der Tabellen so, dass dem erfolgreichen Bieter nach Abschluss der Versteigerung eines Gegenstandes zur Geschäftsabwicklung automatisch die Mail-Adresse des Anbieters zugeschickt wird (per E-Mail natürlich). Wie E-Mail bearbeitet wird, erfährt man aus Abschnitt 7.7 des Buches.

**Aufgabe 9-5: Im Auktionskatalog die in den Bieterkatalog übernommenen Artikel kennzeichnen**

Artikel, die bereits in den Bieterkatalog eingetragen sind, sollen im Auktionskatalog entweder deutlich gekennzeichnet (z.B. durch Fett- oder andersfarbige Schrift) oder ganz weggelassen werden. Damit soll verhindert werden, dass für bereits übernommene Artikel weitere Übernahmeversuche unternommen werden.

**Aufgabe 9-6: Gegen sich selbst bieten**

In der vorliegenden Version kann ein Bieter sich selbst überbieten. Ändern Sie das Programm so, dass das nicht mehr möglich ist.

**Aufgabe 9-7: Angezeigter Bietestand wird inaktuell**

Bei konkurrierendem Bieten kann der Fall eintreten, dass ein Bieter von einem Stand aus bietet, der von anderen bereits überboten ist. Verhindern Sie, dass ein solcher Fall eintreten kann!

**Aufgabe 9-8: Anfrage vor der Ausführung**

An folgenden Stellen soll eine Anfrage erscheinen, ob wirklich so verfahren werden soll:

- beim Bieten,
- beim Löschen eines Artikels aus dem Bieterkatalog,
- beim Löschen oder Ändern eines Gegenstandes im Auktionskatalog.

**Aufgabe 9-9: Bieteagent**

Statt in vorgeschriebenen prozentualen Stufen von 2,5% zu steigern, wird ein *maximales* Gebot abgegeben. Es muss über dem Limit bzw. dem augenblicklichen Gebot liegen. Ein „Agent“ sorgt dafür, dass davon aber nur der Betrag verwendet werden, um bei der Versteigerung zu führen, d.h. die Auktionssoftware übernimmt das Bieten im vorgegebenen finanziellen Rahmen (siehe z.B. ebay.de).

Zusatzaufgabe: Wenn das maximale Gebot überboten wird, erhält der Bieter eine E-Mail mit dem Hinweis auf diesen Tatbestand (E-Mail vgl. Abschnitt 7.7 des Buches).

