

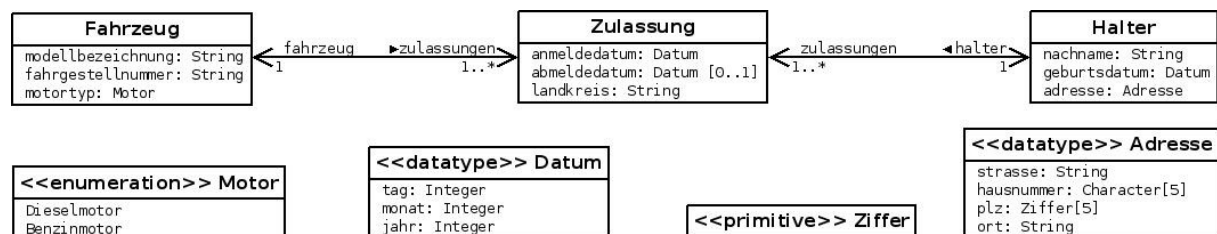
## Übungen zur Vorlesung Informatik II, Blatt 6

Abgabe: Donnerstag, 31.05, 14.00 Uhr, Postkasten der Veranstaltung und Mail an Tutor  
Dieses Übungsblatt muss im Team abgegeben werden.  
Bitte Namen und Matrikelnummer aller Teammitglieder angeben.

- \* leichte Aufgabe
- \*\* mittelschwere Aufgabe
- \*\*\* schwere Aufgabe

### Aufgabe 21 \* (Objektrelationale Abbildung und SQL)

Übersetzen Sie den folgenden Ausschnitt eines Klassendiagramm in Tabellen einer relationalen Datenbank und tragen Sie in jede Tabelle exemplarisch zwei Datenobjekte ein.



Tabellen dürfen Sie in folgender Form angeben:

Tabellenname		
Spaltenname1	Spaltenname2	...
...	...	...

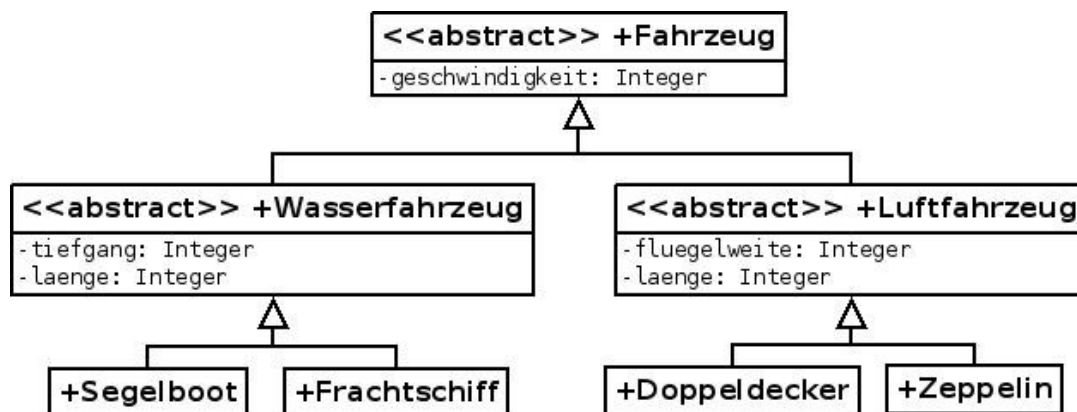
Entwerfen Sie SQL-Anfragen, um:

- die Tabelle zur Klasse **Zulassung** zu kreieren
- einen neuen Halter aufzunehmen
- alle Datum-Datensätze ab dem 01.01.2010 zu löschen
- die Modellbezeichnung aller Fahrzeuge mit Dieselmotoren auszugeben

Diskutieren Sie die Vor- und Nachteile, den Datentyp **Datum** als eigene Tabelle darzustellen. Überlegen Sie sich eine alternative Vorgehensweise.

## Aufgabe 22 \*\* (Objektrelationale Abbildung und Generalisierungen)

Geben Sie drei verschiedene Übersetzungen der folgenden Vererbungshierarchie in Tabellen einer relationalen Datenbank an und füllen Sie jedes Modell mit Beispieldaten (mit mindestens einem Objekt pro konkreter Klasse).



Tabellen dürfen Sie in folgender Form angeben:

Tabellenname		
Spaltenname1	Spaltenname2	...
...	...	...

Versuchen Sie, für alle drei Alternativen SQL-Anfragen zu entwerfen, um:

- bei allen Segelbooten den Tiefgang um 1 zu erhöhen
- bei allen Fahrzeugen die Geschwindigkeit um 10 Prozent zu erhöhen
- ein neues Frachtschiff einzufügen

Welche Schwierigkeiten hatten Sie? Diskutieren Sie die Vor- und Nachteile der drei Alternativen auf der Basis Ihrer Erfahrungen mit der Formulierung von SQL-Anfragen.

## Aufgabe 23\*\* (Datenbankzugriff mit Pong)

Wir haben auf unserem Server `aiomr.informatik.uni-augsburg.de` eine MySQL-Datenbank mit dem Namen `pong` bereitgestellt. Das Kennwort ist `user`, das Passwort ist `password`. Diese Datenbank enthält u.a. die (bis jetzt leere) Tabelle:

Users		
id	username	password
...	...	...

Die Spalte `id` hat den SQL-Datentyp `int(11)`, wird als Primärschlüssel verwendet und automatisch erhöht (bei einem `INSERT`-Statement übergibt man für diese Spalte daher immer 0 und die Datenbank kümmert sich selbst um die Eintragung des nächsthöheren Werts). Die Spalten `username` und `password` haben den SQL-Datentyp `VARCHAR(30)` bzw. `VARCHAR(32)`.

Zusätzlich zur Angabe erhalten Sie mit diesem Übungsblatt die Musterimplementierung von Pong erweitert um den Menüeintrag `onlinemodus` und den beiden Dialogfenstern `LoginDialog` für den Benutzerlogin und `RegisterDialog` für die Benutzerregistrierung.

Der Quellcode der beiden neuen Dialogfenster muss an den kommentierten Stellen nun erweitert werden, damit eine Verbindung zur Datenbank aufgebaut werden kann und Datenbankabfragen sowie Eintragungen in die Datenbank vorgenommen werden können.

Insgesamt gibt es neun kommentierte Lücken im Code (`LoginDialog`: 6, `RegisterDialog`: 3).

### Aufgabe 24\*\*\* (Datenbankzugriff über GUI)

Implementieren Sie nun ein Programm zum Anlegen von Paketsendungen mit GUI und Datenbankzugriff in einer 3-Schichtenarchitektur nach folgender Anleitung. Dazu gibt es auf dem in der vorherigen Aufgabe erwähnten Server `aiomr.informatik.uni-augsburg.de` eine MySQL-Datenbank mit dem Namen `theDatabase`, welche auch über das Kennwort `user` und dem Passwort `password` erreichbar ist. Die Fachkonzeptklassen `Paket` und `PaketContainer` sowie das fertig implementierte Paketverwaltungsfenster `Fenster` sind bereits vorgegeben und im Quelltext beigefügt. Die Klasse `PaketContainer` erzeugt momentan einige Compilerfehler, da die im nachfolgenden Text aufgeführten Klassen noch nicht vorhanden sind. Passen Sie daher Ihre selbst geschriebenen Klassen dementsprechend an, damit es nicht mehr zu Compilerfehlern kommt.

- a) Erzeugen Sie in der Programmdatei `Aufgabe24.java` das Paketverwaltungsfenster.
- b) Implementieren Sie eine Schnittstelle `PaketeDatenhaltung` zur Datenhaltung mit der im Vorlesungsskript als UML-Diagramm vorgegebene Schablone (Kapitel 10 Folie 31-32):
  - i. Es müssen die Fachkonzeptklassen angepasst werden.
  - ii. Eine Schnittstelle enthält nur Methodenrumpfe und keine Implementierung.
- c) Implementieren Sie eine Klasse `PaketeDatenbank` mit folgenden Eigenschaften:
  - i. Die Klasse `PaketeDatenbank` soll die Schnittstelle `PaketeDatenhaltung` implementieren.
  - ii. Sie soll die Klassenattribute `treiber` und `datenbankURL` sowie die Attribute `benutzername` und `password` (alle vom Typ `String`) für Verbindungsinformationen zur Datenbank enthalten.
  - iii. Sie soll ein Attribut `connection` vom Typ `Connection` zur Repräsentation der Verbindung zur Datenbank enthalten.
  - iv. Im Konstruktor `public PaketeDatenbank(String benutzer, String password)` soll versucht werden, eine Verbindung zur Datenbank herzustellen. Tritt eine Ausnahme auf, geben Sie eine entsprechende Meldung auf Kommandozeile aus.
  - v. Sie muss eine `add`-Methode zum Eintragen der Daten eines übergebenen `Paket`-Objekts in die Tabelle `Paket` enthalten (wird von `addPaket` aus `PaketeContainer` aufgerufen). Tritt eine Ausnahme auf, geben Sie eine entsprechende Meldung auf Kommandozeile aus.
  - vi. Es muss eine `delete`-Methode zum Löschen eines zum übergebenen `Paket`-Objekt zugehörigen Datenbankeintrags in der Tabelle `Paket` existieren (wird von `deletePaket` aus `PaketeContainer` aufgerufen). Tritt eine Ausnahme auf, geben Sie eine entsprechende Meldung auf Kommandozeile aus.
  - vii. Sie muss eine `modify`-Methode zum Ändern eines vorhandenen Datenbankeintrags mit den Daten des übergebenen `Paket`-Objekts in die Tabelle `Paket` enthalten (wird von `modifyPaket` aus `PaketeContainer` aufgerufen). Tritt eine Ausnahme auf, geben Sie eine entsprechende Meldung auf Kommandozeile aus.
  - viii. Es muss eine `load`-Methode, in der für jede Zeile in der Tabelle `Paket` ein zugehöriges `Paket`-Objekt erzeugt und in das übergebene `PaketContainer`-Objekt eingefügt wird, existieren (wird im Konstruktor von `PaketeContainer` aufgerufen). Tritt eine Ausnahme auf, geben Sie eine entsprechende Meldung auf Kommandozeile aus.
  - ix. Alle anderen zu implementierenden Methoden sollen beim Aufruf nur einen Hinweis auf der Konsole ausgeben und sonst keine Funktionalität haben.
- d) Diskussion:
  - i. Wie läuft das Anlegen einer neuen Sendung im Detail ab, damit von der Benutzereingabe in der GUI letztendlich der dazugehörige Eintrag in der Datenbanktabelle erstellt worden ist? Ist eine Warnung auf Kommandozeile (wie oben vorgeschlagen) ausreichend, wenn es zu Verbindungsfehlern mit der Datenbank kommt? Machen Sie sich über Alternativen Gedanken.
  - ii. Aus welchem Grund wird das Attribut `isFirstLoad` in der Klasse `PaketContainer` benötigt?
  - iii. Welchen Effekt können Sie beobachten, wenn Sie das Programm zwei Mal starten (also gleichzeitig zwei Instanzen des Paketverwaltungsfenster laufen) und in einem Fenster neue Sendungen anlegen oder vorhandene Sendungen ändern bzw. löschen und danach im anderen Fenster die gleiche Aktion durchführen?