

Betreutes Programmieren Vorlesung Informatik II, Blatt 7

Musterlösung: ca. 122 Codezeilen*

Aufgabenstellung

Heute sollen Sie eine kleine Anwendung erstellen, die sich mit einer Datenbank verbindet, Information über diese Datenbank ermittelt und schließlich anzeigt.

Setzen Sie das UML-Diagramm aus Abbildung 1 in Java-Code um. Aus Platzgründen wurden Schnittstellen teilweise als gestrichelte Klassen dargestellt. Berücksichtigen Sie auch die Beschreibungen nach dem Diagramm.



Abbildung 1: UML-Klassendiagramm der Datenkomponenten

Die Information soll dabei wie folgt gewonnen werden:

***Nicht gezählt:** Leerzeilen, Kommentare, **imports**. **In eigenen Zeilen:** `{` und `}` (ca. 52x), jede Anweisung, außer bei Deklarationen gleichen Typs (z. B. `int a, b, c = 0, ...`) oder bei einer einzelnen Anweisung nach einem `if` oder einem `else`. **Nicht benutzt:** `?:`-Operator.

- Zuerst wird ein Objekt der Klasse **DataBase** erzeugt, wobei der Name der Datenbank, über die Information gesammelt werden soll, dem Konstruktor übergeben wird.
- Ein Aufruf der Methode **connectToDataBase ()** stellt die Verbindung zum Server und der gewählten Datenbank her und signalisiert über den Rückgabewert, ob die Aktion erfolgreich war. Im Erfolgsfall wird das Verbindungsobjekt auch im Attribut **connection** gespeichert.
- Wird die Methode **collectInformationFromDataBase ()** aufgerufen, holt sie sich die Menge aller Tabellen der gespeicherten Datenbank, erzeugt für jede Tabelle ein Objekt der Klasse **Table** und fügt es dem Attribut **tables** hinzu.
- Jedes Objekt der Klasse **Table** holt sich wiederum die Menge aller Spalten seiner Tabelle, erzeugt für jede Spalte ein Objekt der Klasse **Column** und fügt es seinem Attribut **columns** hinzu.
- Jedes Objekt der Klasse **Column** ermittelt den Namen und den Typ seiner Spalte und speichert diese Information in den Attributen **name** und **type**.

Die gesammelte Information soll dann vom Programm ausgegeben werden. Es ist Ihnen dabei freigestellt, ob Sie lediglich eine Ausgabe auf der Konsole erzeugen oder sogar eine grafische Anwendung erstellen. Je nachdem kann der Name der Datenbank auch ein Kommandozeilenparameter oder eine Eingabe in ein Textfeld sein.

Schreiben Sie ein Hauptprogramm, das Ihre Implementierung testet. Benutzen Sie für den Test die Datenbank `theOtherDatabase`, die auf unserem Server bereitliegt.

Recherchieren Sie in der Dokumentation von Java nach geeigneten Klassen und Methoden. Sie finden die Doku im Internet unter der Adresse <http://docs.oracle.com/javase/7/docs/api/index.html>. Das Paket **sql** ist dabei sehr interessant. Zum leichteren Einstieg können Sie die nachfolgenden Hinweise verwenden.

Hinweise

Nachdem Sie eine Verbindung zum Datenbanksystem hergestellt haben, bietet das Verbindungsobjekt über die Schnittstelle **Connection** Zugriff auf die Methode **getMetaData ()**, die ein Objekt zurückliefert, das die Information über die Datenbank repräsentiert.

Das zurückgelieferte Objekt implementiert die Schnittstelle **DatabaseMetaData** und stellt bspw. die Methoden **getTables ()** und **getColumns ()** zur Verfügung. Beide Methoden erwarten Parameter, die das Ergebnis jeweils einschränken sollen. Jeder einzelne Parameter kann den Wert **null** bekommen, um anzuzeigen, dass keine Einschränkung zu machen ist. Beide Methoden liefern ihr Ergebnis als ein Objekt zurück, das die Schnittstelle **ResultSet** implementiert. Dessen Methode **next ()** kann dazu benutzt werden, über alle Ergebnisse zu iterieren, wobei die Methode **getString ()** verwendet werden kann, um einzelne Einträge der Ergebnisse auszulesen. Welche Einträge jeweils für die Lösung der Aufgabe sinnvoll sind entnehmen Sie der Dokumentation.

Beachten Sie, dass die Methoden, die Sie benutzen, jeweils verschiedene Ausnahmen werfen können. Aus diesem Grund ist es zwingend notwendig, die Methoden in einen sog. **try-catch**-Block einzufassen. Das sieht dann einfach so aus:

```
try
{
    ⟨Anweisungen, die Ausnahmen werfen⟩
}
catch (Exception e)
{
    ⟨ggf. Anweisungen, die den Fehlerfall betreffen⟩
}
```

Auf diese Weise werden alle Ausnahmen abgefangen. Später werden Sie noch lernen, wie man gezielt auf verschiedene Ausnahmen reagieren kann.