

# Gedanken zur Verwendung von Datenbanken in Java-Projekten

## 1. Wahl des Datenbanktyps

- **MS Access** ermöglicht das einfache Aktualisieren der Datenbank mit einem vorinstallierten Datenbankprogramm, wird aber nicht unter Linux oder Mac laufen
- **Libre Office** kann .odb-Datei erzeugen und bearbeiten, für Java muss das .odb-Paket aber entzippt werden und alle entstandenen Dateien (die zu einer HSQLDB-Datenbank gehören) müssen einen Bezeichner vor dem Punkt kriegen
- **MySQL** erfordert einen laufenden MySQL-Server, lokal oder im Web, die Zugangsdaten dazu sind im Code fest
- **SQLite** erfordert keinen Server, Schüler brauchen zum Bearbeiten ein eigenes Programm wie den SQLite Database Browser: <http://sqlitebrowser.sourceforge.net/>
- **Derby** ist das DB-Format von Apache.
- SQLSyntaxfeinheiten: HSQLDB verlangt Anführungszeichen um Tabellen- und Spaltenbezeichner, MySQL erlaubt genau die nicht; Access und SQLite sind am tolerantesten.

## 2. Auftreiben und Integrieren des Treibers

- Herunterladen im Web, nötig jeweils für SQLite, MySQL, HSQLDB (d.h. Base)
- Entweder bei BlueJ über Tools/Preferences/Libraries einbauen...  
...oder zentral bei BlueJ in **C:/Programme/BlueJ/lib/userlib/** einbauen...  
...oder jeweils bei BlueJ in **"+libs"**-Verzeichnis innerhalb Projektordner einbauen
- beim Export in jar-Archiv beachten: der Pfad zum Treiber-Paket muss im Class Path des aktuellen Java-Projekts stehen; macht BlueJ automatisch durch Eintrag in META-INF/MANIFEST.MF, wo zum Beispiel auch der Ort der main-Methode angegeben ist

## 3. Speicherort der Datenbank (falls keine Serverlösung)

- Problem 1:
  - Dateien in einem .jar-Archiv, das gerade ausgeführt wird, sind immer schreibgeschützt
  - deshalb macht es wenig Sinn, wenn die Datenbankdatei *in* diesem Archiv ist
  - also *nicht im* BlueJ-Verzeichnis anlegen, sondern parallel dazu, da nur dann ein Export in .jar sinnvoll
- Problem 2:
  - alle relativen Pfadangaben sind relativ zu dem Verzeichnis, von dem aus Java aufgerufen wird, und das kann alles mögliche sein (auch wenn es oft gut geht, weil man oft im selben Verzeichnis Java aufruft, in dem sich die Projektdatei befindet)
  - das Verzeichnis, in dem Java ausgeführt wird, ist bei einem BlueJ-Projekt und dem daraus exportierten jar-Archiv jeweils ein anderes
  - Lösung: zur Laufzeit den absoluten Pfad zur Datenbankdatei ermitteln und bei der Verbindung zur Datenbank benutzen

Auch die JGUIToolbox ([http://www.ovtg.de/3\\_arbeit/informatik/JGUIWeb/JGUIWebstart.html](http://www.ovtg.de/3_arbeit/informatik/JGUIWeb/JGUIWebstart.html)) stellt Methoden zur Verfügung, mit denen Datenbankeinsatz vereinfacht werden kann. Der Export in ein .jar-Archiv funktioniert damit noch nicht, aber dafür gibt es sehr viel mehr Möglichkeiten für den Datenbankzugriff (rollback usw.)