

# TdI 2013 – Datenbanken in Projekten

1. Beispiele von Oberstufenprojekten
2. Schwierigkeiten bei Datenbanken

# Kriegergame: nach Anmeldung



# Kriegersgame: Ausrüstung



# Kriegergame: Duell



# Quiz: Startmenü



# Quiz: Frage

? Wer wird Abiturient?

Datei Hilfe

**WER WIRD ABITURIENT**

Frage 1 : 1. Klasse

Wie viel ist  $3+10$  ?

A: 7 B: 13

C: 3 D: 9

Spicken **PUNKTE:** 0 Beenden

# Quiz: Datenbanktabelle (q\_frage)

	id	frage	aw1	aw2	aw3	aw4	ra	sg
1	1	Wie heißt die Schnur am Bogen, die die Pfeile absc...	Sehne	Sense	Seele	Senkel	1	1
2	2	In welchem Jahr betrat der erste Mensch den Mond?	1968	1969	1970	1971	2	8
3	3	Wie heißt der berühmte Wahlspruch der Benedikti...	Esse ...	Bete ...	Scha...	Bete...	2	2
4	5	Wie oft wurde Deutschland Weltmeister?	1	2	3	4	3	4
5	6	In welcher Stadt wurde der "Trabant" hergestellt?	Jena	Alten...	Dresen	Zwic...	4	6
6	7	Wo wurde Deutschland Weltmeister?	Berli...	Rom,...	Rio, ...	Bern...	4	6
7	8	Gegen wen gewann Deutschland das Finale 1954?	Ungarn	Italien	Arge...	Brasi...	1	7
8	9	Das Wort "Omnibus" kommt aus dem Lateinischen. Was bedeutet es?	Großer Wagen	Viel Platz	Für alle	Viele Leute	3	9
9	10	Die Erde hat einen Durchmesser von 12.756 km. Welcher Planet hat einen ähnlichen Durchmesser?	Venus	Mars	Uranus	Saturn	1	11
10	11	Wie heißt der erste DSDS-Gewinner?	Alex...	Alexa...	Alex...	Alex...	2	2
11	12	Alle Wege führen nach ....	St. P...	Rom	Mün...	Hause	2	1

"SELECT \* FROM `q\_frage` WHERE sg="+grad+";"

# Erfahrungen mit Datenbanken

- Funktioniert auf dem einen Betriebssystem, auf anderem nicht.
- Funktioniert in BlueJ/Eclipse, als exportiertes .jar-Archiv nicht.
- Funktioniert auf dem einen Rechner, auf dem anderen nicht.



# Drei Probleme bei Datenbanken

1. Aufbau einer Abfrage  
(Grundlagen)
2. Wahl des Datenbanksystems und Finden der  
Treiber dazu  
(logistisches Problem)
3. Speicherort der Datenbankdatei  
(tückisch)

# 1. Aufbau einer Abfrage

daneben: korrekten Treiber finden

//1. Treiber laden

Class.forName(treiberbezeichner);

Hauptproblem

//2. Verbindung herstellen

Connection conn = DriverManager.getConnection(protokoll+urlstring, user, kennwort);

//3. Statement erzeugen

Statement stmt = conn.createStatement();

//4. Abfrage durchfuehren und Ergebnismenge erhalten

ResultSet rset = stmt.executeQuery(anfragestringInSQL);

//bzw. stmt.executeUpdate(auftragInSQL)

daneben: je nach DB  
Syntaxvarianten

//5. Ergebnismenge verarbeiten (in Schleife, mit getInt/getString)

while (rset.next()) { System.out.println (rset.getString(spaltenbezeichner); }

//6. Ergebnismenge schliessen

rset.close();

//7. Statement schliessen

stmt.close();

//8. Verbindung schliessen

conn.close();

# Feinheiten bei der Syntax

			Access	MySQL	Base	SQLite
"SELECT * FROM fragen	WHERE aw1	= 'H2O';"	ok	ok	-*	ok
"SELECT * FROM \"fragen\"	WHERE \"ID\"	= 0;"	ok	-	ok	ok
"SELECT * FROM `fragen`	WHERE `ID`	= 0;"	ok	ok	-	ok
"SELECT * FROM 'fragen'	WHERE 'aw1'	= 'H2O';"	-	-	-	-

Je nach System unterschiedlich:  
mit Anführungszeichen, nichts, oder Backtick um Tabellen- und Spaltenbezeichner

\* automatisches Umbenennen der Bezeichner in Großbuchstaben

## 2. Wahl des Datenbanksystems

- a) MS Access
- b) SQL-Server
- c) Libre/Open Office Base
- d) SQLite

## a) Microsoft Access

- Treiber ist in Java dabei
- Wenn Access installiert:  
Leichte Bearbeitung der  
Datenbankdatei  
außerhalb von Java
- Fkt. nur in Windows, und  
nur wenn Access  
installiert

## b) SQL-Server

- OS-unabhängig
- Keine Probleme bei der Pfadangabe zur Datenbank
- Treiber muss importiert werden (BlueJ/Tools/Preferences/Libraries)
- Server muss laufen, lokal (z.B. XAMPP) oder im Web (Zugangsdaten im Code)
- Umständliche Bearbeitung der DB z.B. über phpMyAdmin

## c) Libre/Open Office Base

- OS-unabhängig
- Leichte *Erstellung* der Datenbankdatei außerhalb von Java
- Treiber muss importiert werden (BlueJ/Tools/Preferences/Libraries)
- Arbeiten nur auf der entzippten .odb-Datei\* und nach Umbenennung möglich > keine einfache Bearbeitung nach der Erstellung

\* .odb-Dateien sind gezippte Pakete mit u.a. einer Datenbank im HSQLDB-Format. Zum Nutzen auspacken, und alle ".xxx"-Dateien umbenennen in "dbname.xxx"

## d) SQLite

- OS-unabhängig
- Speicherung der DB in einer einzigen Datei, die mit Open-Source-Editor bearbeitet werden kann
- Einfache Bearbeitung der DB außerhalb von Java mit Open-Source-Programm, z.B. "SQLite Database Browser"
- Treiber muss importiert werden (BlueJ/Tools/Preferences/Libraries)
- nur Teilmenge von SQL (für Schule ausreichend), keine echten Datentypen, keine Fremdschlüssel!

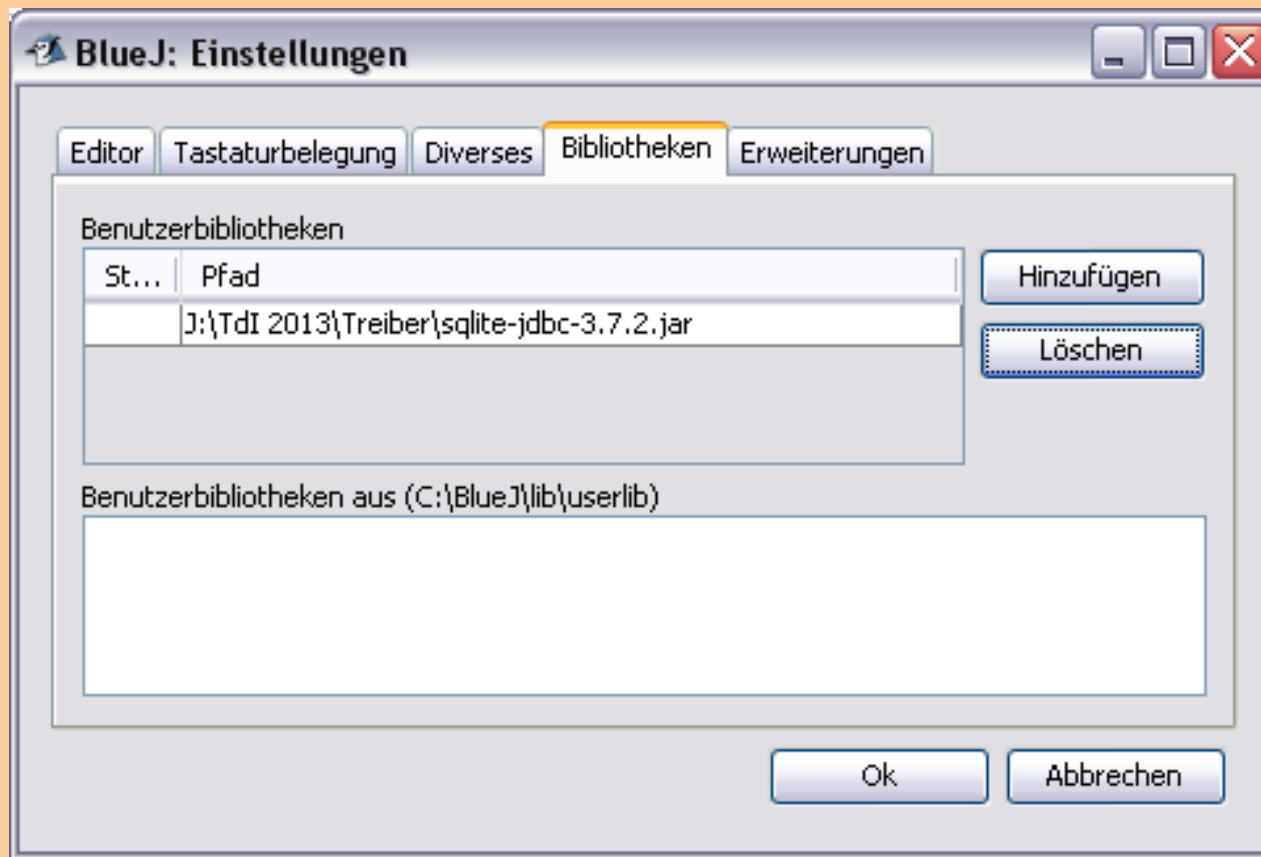


# Treiber und deren Einbau

```
try {  
    //1. Passenden Treiber laden  
    Class.forName(treiberbezeichner) ;  
    //2. Verbindung zur DB erstellen  
    Connection conn = DriverManager.getConnection(  
        protokoll+datenbank, user, kennwort  
    );  
}  
catch (Exception e) {  
}
```

DB-Typ	Treiberbezeichner	Protokoll	Treiber zu finden z.B. in
MS Access	"sun.jdbc.odbc.JdbcOdbcDriver"	"jdbc:odbc:Driver={Micro-integriert-soft Access Driver (*.mdb)};DBQ="	
MySQL	"com.mysql.jdbc.Driver"	"jdbc:mysql://"	mysql-connector-java-5.1.25-bin.jar
Base	"org.hsqldb.jdbcDriver"	"jdbc:hsqldb:"	hsqldb.jar
SQLite	"org.sqlite.JDBC"	"jdbc:sqlite:"	sqlite-jdbc-3.7.2.jar

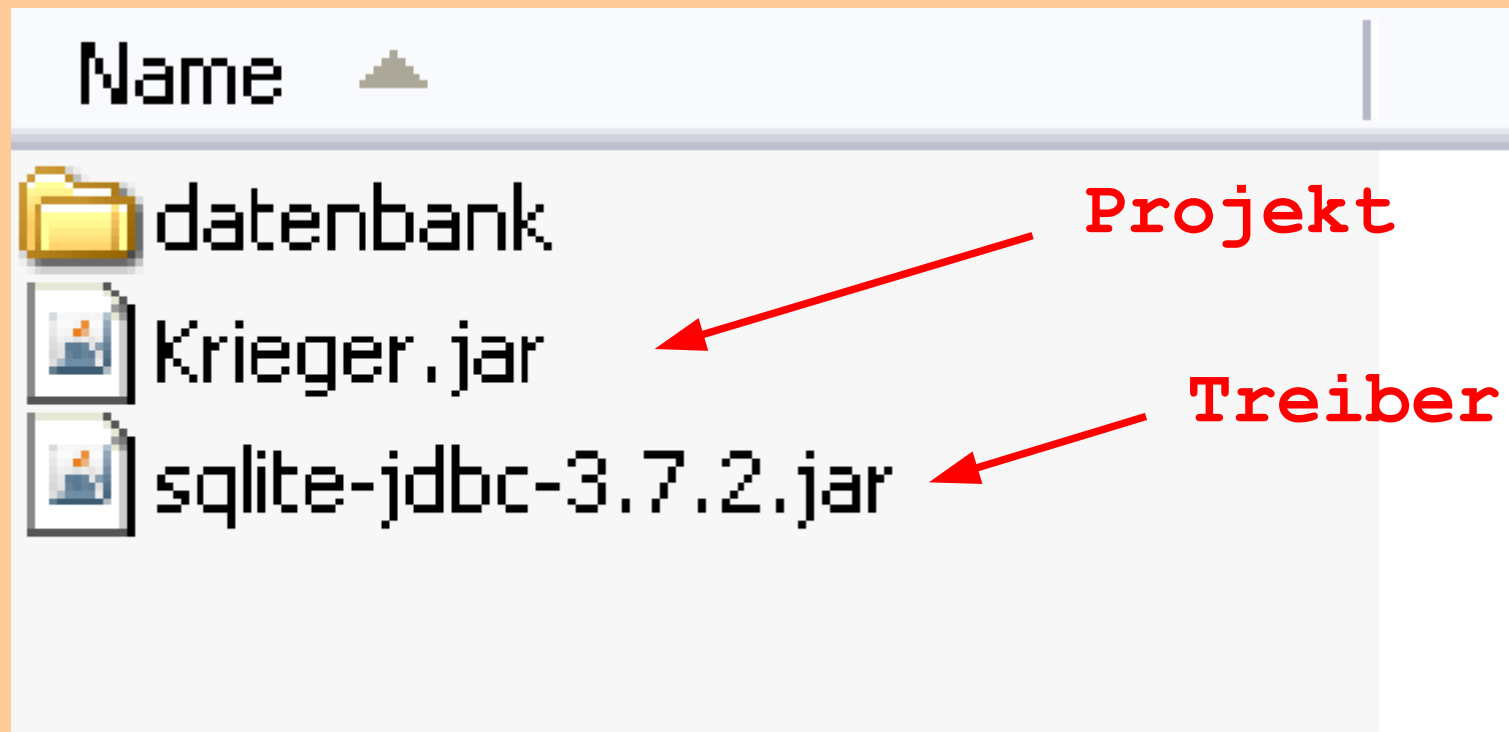
# Treiber einbauen



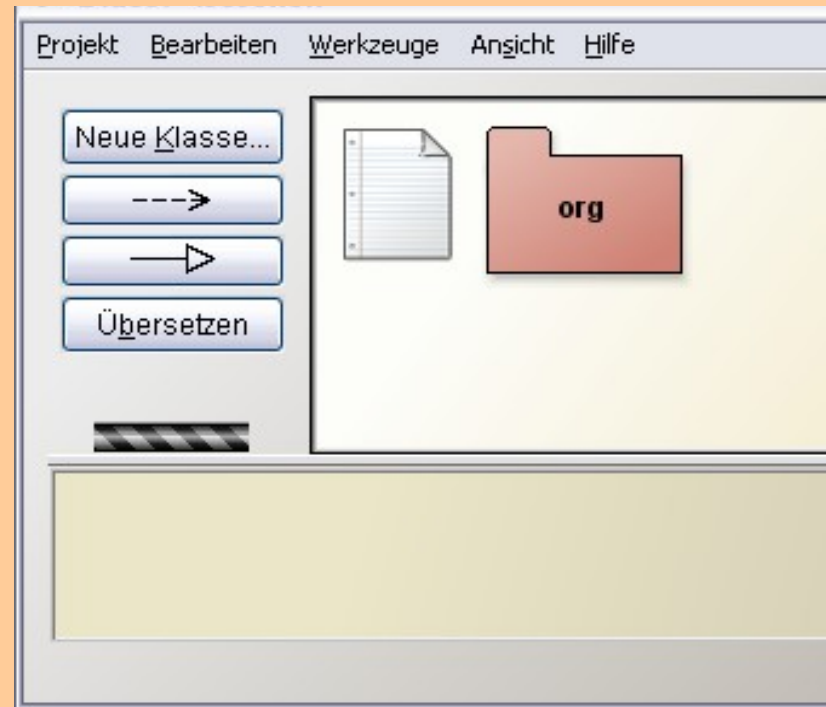
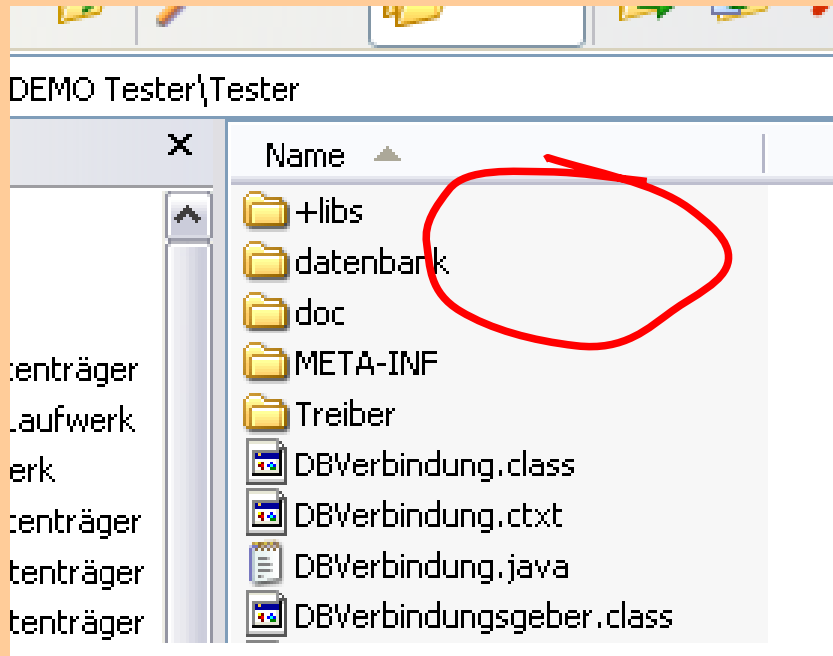
- Hinzufügen zu BlueJ:  
Tools/Preferences/Libraries

# Erzeugen eines .jar-Archivs

- Treiber-Archiv in der Regel parallel zum Projekt-Archiv



# Treiber einbauen (Varianten)

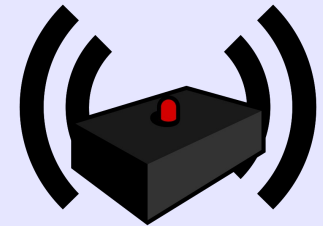


- .jar-Archiv in Projekt-Unterordner namens `+libs`
- Treiber-Archiv entpacken und dem Projekt hinzufügen

### 3. Speicherort der Datenbankdatei

# Ursprünglicher Gedanke: DB-Server

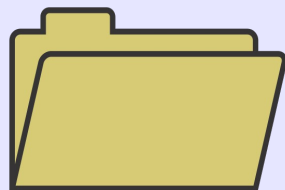
```
Connection conn = DriverManager.getConnection (  
    protokoll+"127.0.0.1/MeineDB",  
    user,  
    kennwort  
);
```



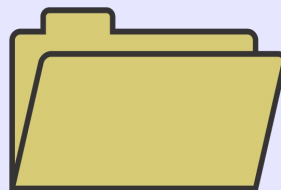
Eigene Dateien

MeinProjekt

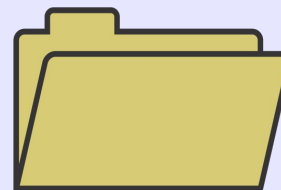
BlueJ-Projektordner



enthält>



enthält>

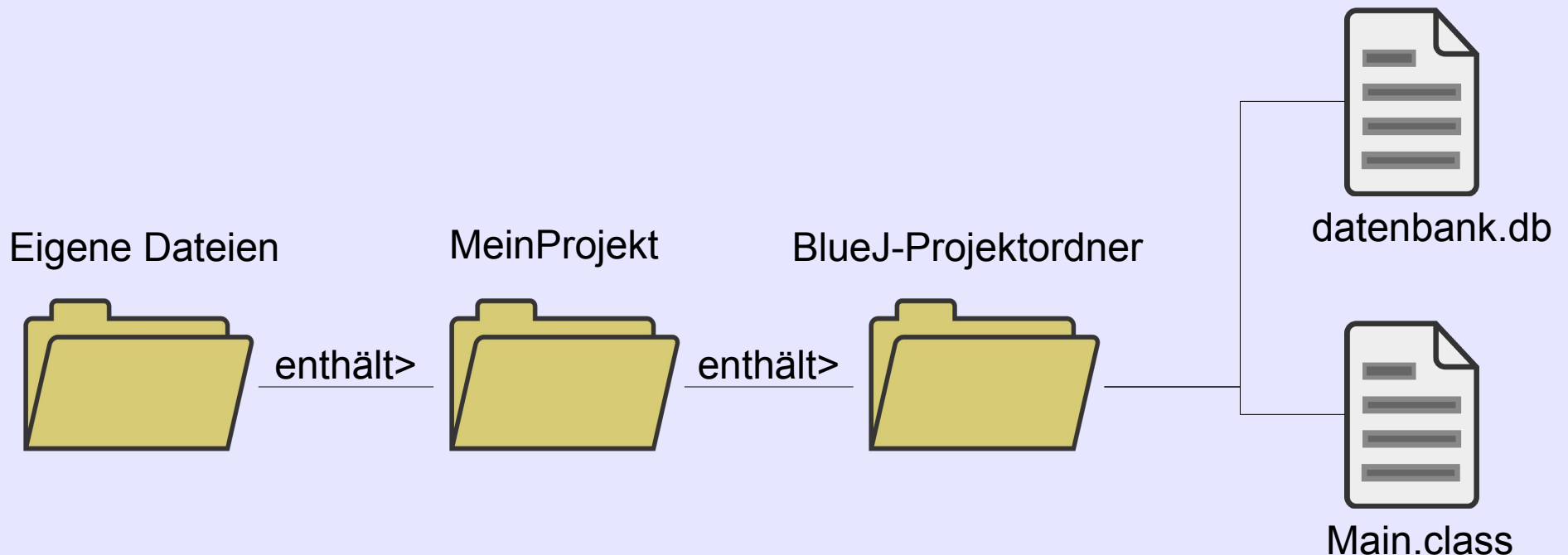


Main.class

# Lokale Datei: Absoluter Pfad

```
Connection conn = DriverManager.getConnection (  
    protokoll+"C:\Eigene Dateien\MeinProjekt\BlueJ-  
Projektordner\datenbank.db",  
    user,  
    kennwort  
);
```

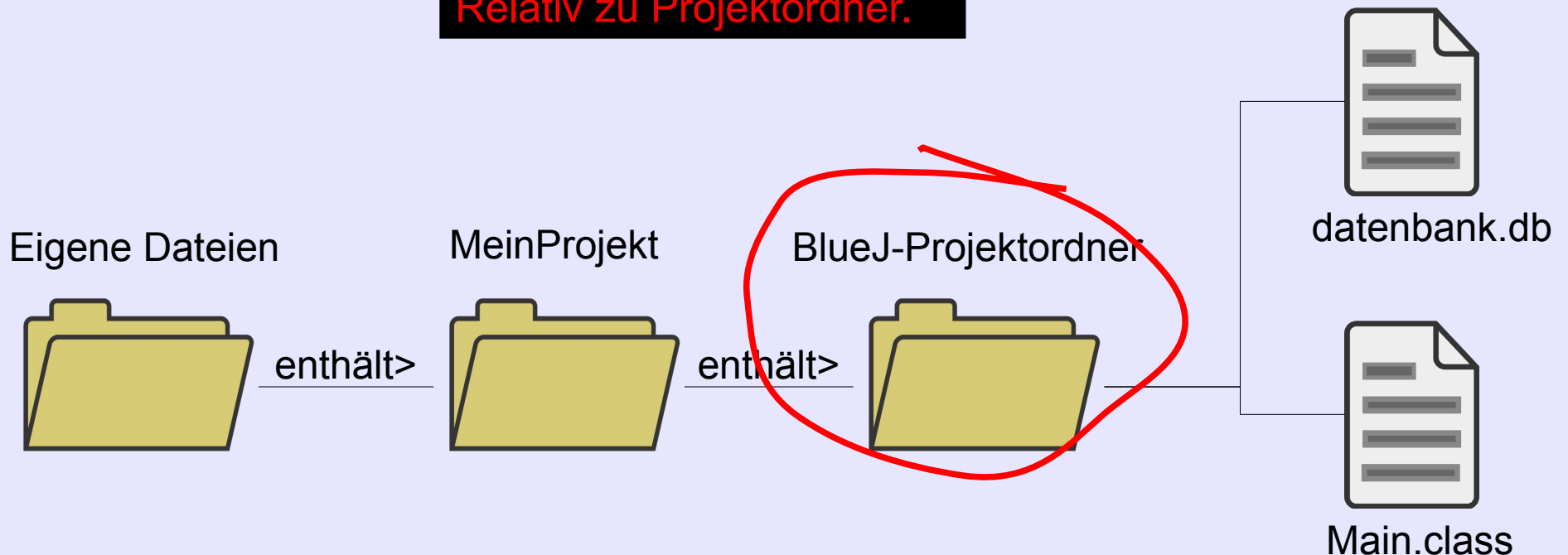
NICHT KLUG



# Lokale Datei: Relativer Pfad (1)

```
Connection conn = DriverManager.getConnection (  
    protokoll+"datenbank.db",  
    user,  
    kennwort  
);
```

Relativ wozu?  
Beim Arbeiten mit BlueJ:  
Relativ zu Projektordner.

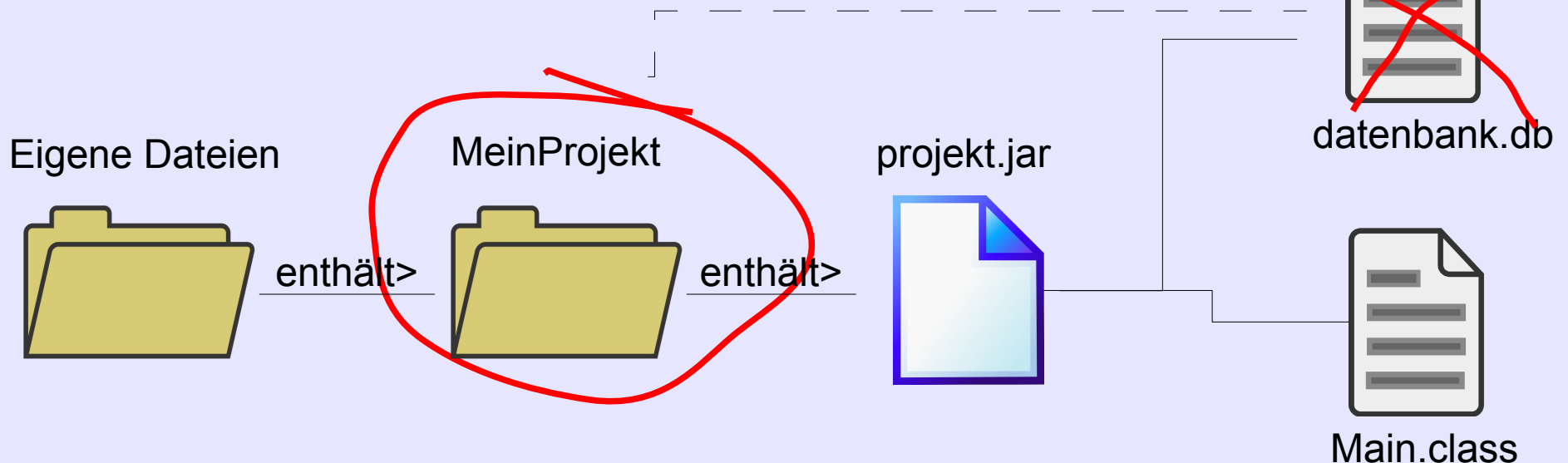




# Lokale Datei: Relativer Pfad (2)

```
Connection conn = DriverManager.getConnection (  
    protokoll+"datenbank.db",  
    user,  
    kennwort  
);
```

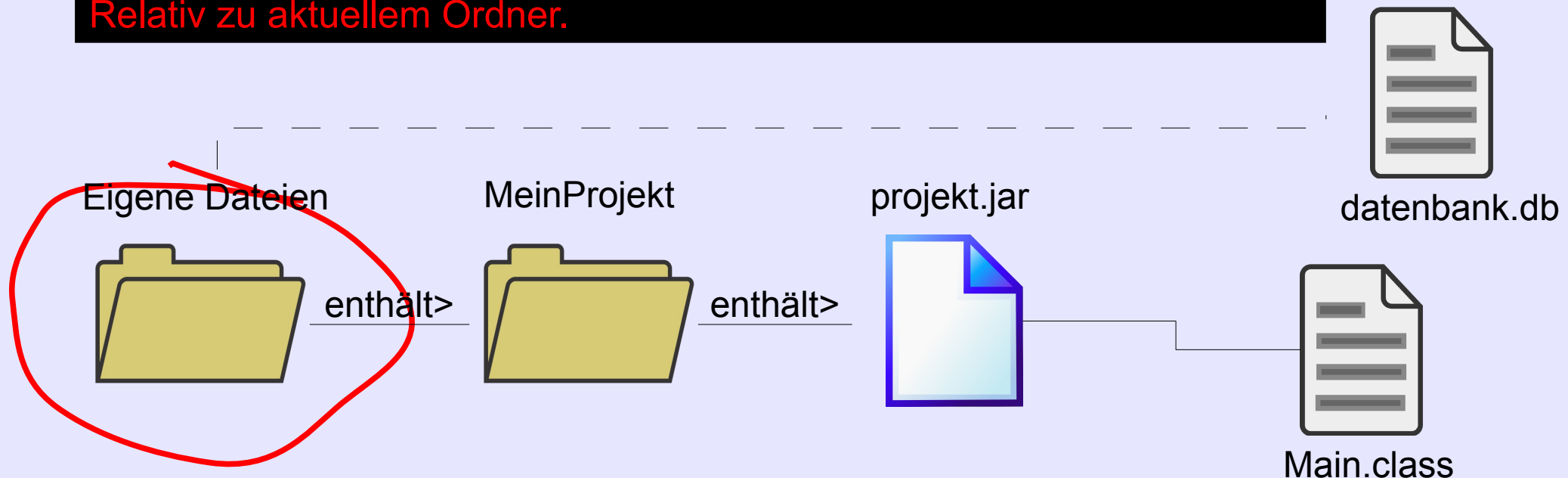
```
C:\Eigene Dateien\MeinProjekt> java -jar projekt.jar  
Relativ zu aktuellem Ordner.
```



# Lokale Datei: Relativer Pfad (3)

```
Connection conn = DriverManager.getConnection (  
    protokoll+"datenbank.db",  
    user,  
    kennwort  
);
```

```
C:\Eigene Dateien> java -jar MeinProjekt\projekt.jar  
Relativ zu aktuellem Ordner.
```



# Grundsätzlich bei Java

- Alle relativen Pfadangaben sind relativ zum *aktuellen* Verzeichnis.
- Welches das ist, kann man vorher nicht wissen.
- Gilt für alle eingebauten Dateien, auch Bilder.

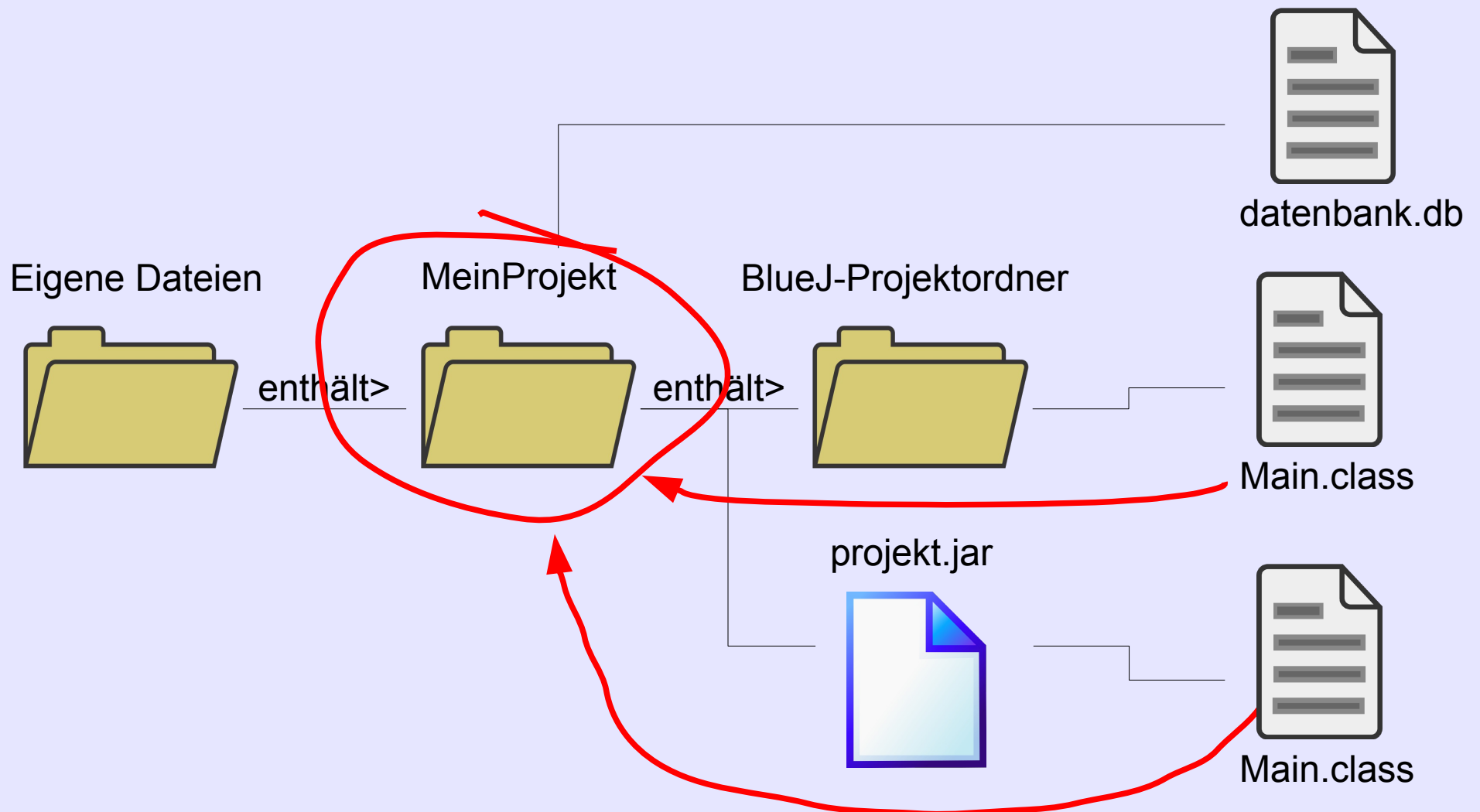
Lösung für Bilder (<http://www.bluej.org/help/archive.html#tip10>):

```
String s = "bild.jpg";  
ImageIcon icon = new ImageIcon();  
java.net.URL imageURL = getClass().getResource(s);  
try{  
    icon = new ImageIcon(imageURL);  
}
```

# Lösung: den absoluten Pfad zur Laufzeit ermitteln

```
Connection conn = DriverManager.getConnection(  
    protokoll+pfadZumOberverzeichnis()+"datenbank.db",  
    user,  
    kennwort  
);
```

`pfadZumOberverzeichnis()` in `Main.class` gibt zur Laufzeit:  
`C:\Eigene Dateien\MeinProjekt\`  
 (also den übergeordneten Ordner, wo die DB ohnehin sein sollte)



```
// Gibt absoluten Pfad zum Oberverzeichnis, in dem sich das Verzeichnis mit der  
// angegebenen Klasse befindet - http://stackoverflow.com/questions/16076911/how-to-  
find-absolute-path-from-a-relative-path-in-file-system
```

```
private String pfadZumOberverzeichnis() {  
    java.security.ProtectionDomain pd = getClass().getProtectionDomain();  
    if ( pd == null ) return null;  
    java.security.CodeSource cs = pd.getCodeSource();  
    if ( cs == null ) return null;  
    java.net.URL url = cs.getLocation();  
    if ( url == null ) return null;  
    java.io.File f = new java.io.File( url.getFile() );  
    if (f == null) return null;  
  
    String pfad = "";  
    try {  
        pfad = f.getParentFile().getAbsolutePath();  
        pfad = java.net.URLDecoder.decode(pfad, "utf-8");  
    } catch (Exception e) { }  
    return pfad + "/";  
}
```