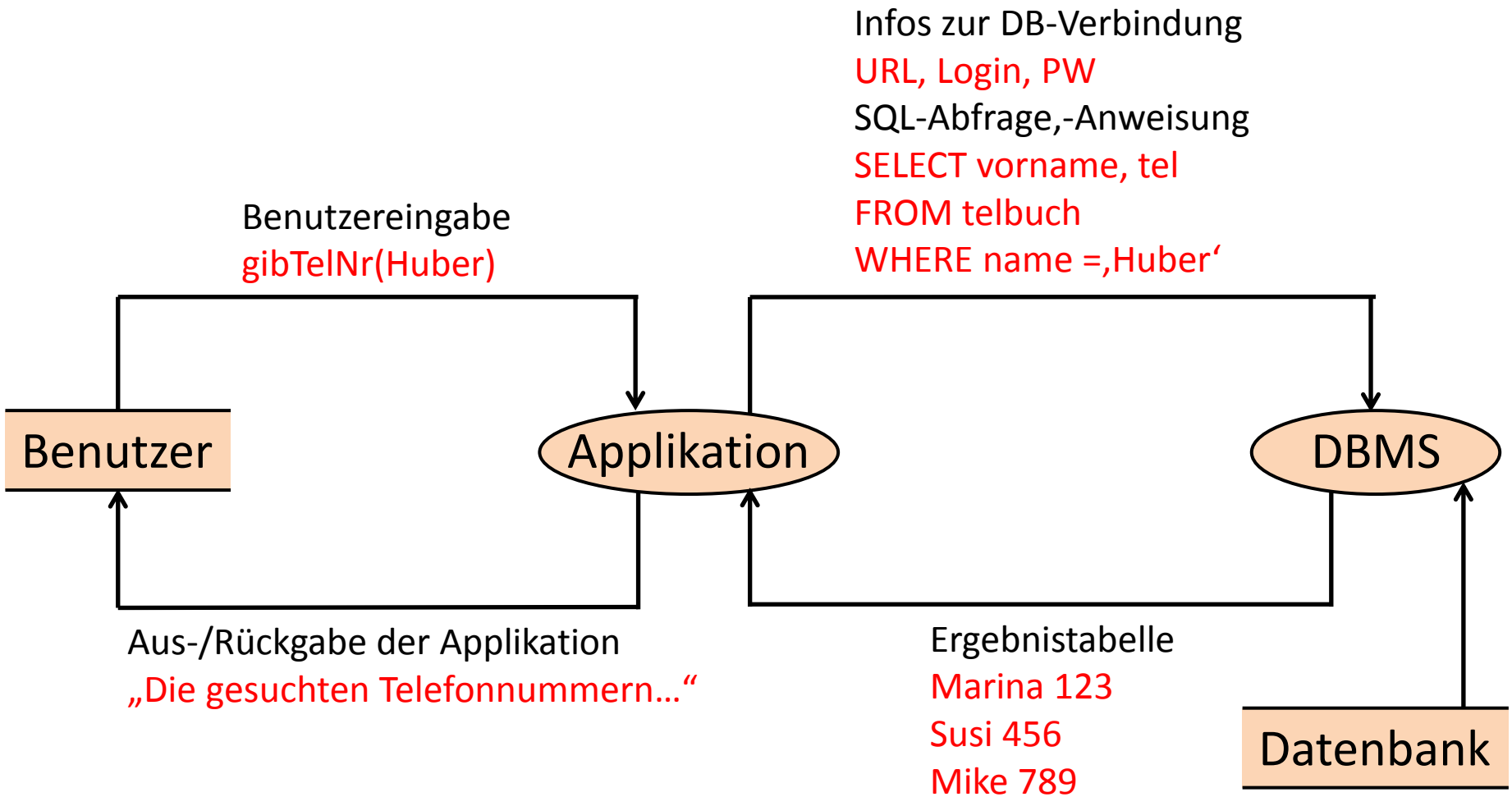
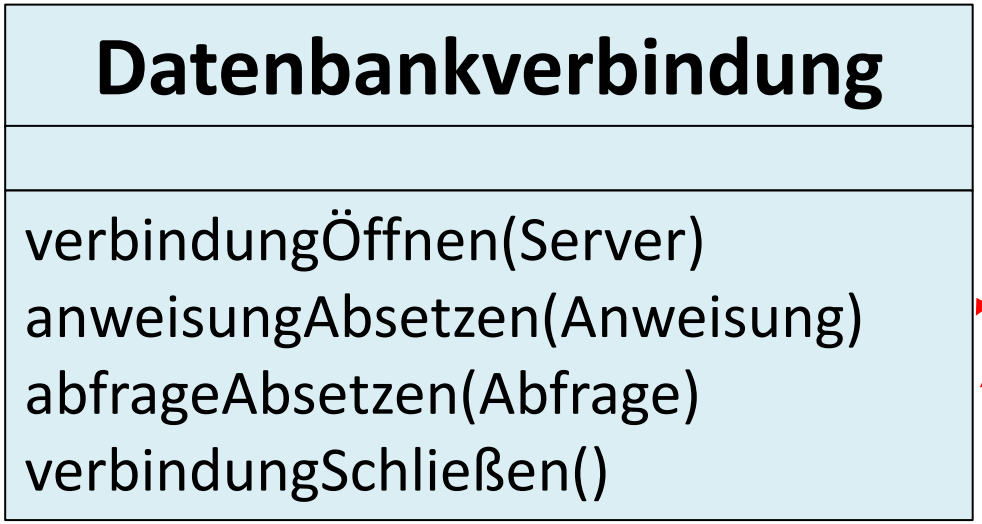
Entwurf der Datenbankverbindung

In einer Datenbank können große Datenmengen persistent (dauerhaft) gespeichert werden. Sie stehen damit auch nach dem Ablauf des Programms zur Verfügung.

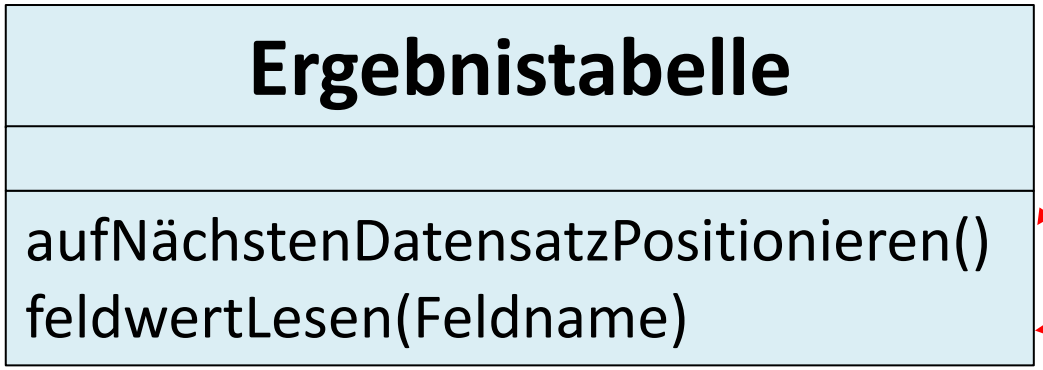
Für den Datenbankzugriff ist ein Datenaustausch zwischen Programm und Datenbanksystem nötig.


Infos zur DB-Verbindung
URL, Login, PW
SQL-Abfrage,-Anweisung
SELECT vorname, tel
FROM telbuch
WHERE name =‚Huber‘
Benutzereingabe
gibTelNr(Huber)
Benutzer
Applikation
DBMS
Aus-/Rückgabe der Applikation
„Die gesuchten Telefonnummern…“
Ergebnistabelle
Marina 123
Susi 456
Mike 789
Datenbank

Grundschema



Erinnerung:
Anweisung
→ CREATE TABLE, SET, UPDATE, DELETE…
Abfrage:
→ SELECT … FROM …
(Antwort ist Tabelle)



Rückgabe: boolean
true, falls nächster Datensatz existiert.

gibt entsprechendes Feld des aktuellen Datensatzes zurück

Ergebnistabelle
aufNächstenDatensatzPositionieren() feldwertLesen(Feldname)

Vorname	Telefonnummer
Marina	123
Susi	456
Mike	789

aufNächstenDatensatzPositionieren()

feldwertLesen(Telefonnummer)
liefert 123

Beispiel: **JGUIToolbox**

Beide datenbankspezifischen Klassen zusammengefasst:
DB_MySQL

```
public class DB_MySQL {
        public Connection db = null;
        public ResultSet res = null;
        …

public boolean conOeffnen(String datenbank,
                                  String user, String kennwort)

public boolean conExecute(String updStr)

public void conAbfrage(String queryStr){ … res = myResult;}

public boolean conClose()
```

Anzahl im ResultSet:

```
public int anzahlDatensaetze()
```

Auf neuen Datensatz schalten:

```
public boolean neuerDatensatz()
```

Get-Methoden:

```
public String getString(int columnIndex)
public String getString(String columnLabel)
```

```
public class DBFormel1_MySQL {

        DB_MySQL        db;

        public DBFormel1_MySQL() {
                db = new DB_MySQL();
        }

        public void dbAufgabe() {
                // Abbruch bei Fehler Datenbank oeffnen
                if (! db.conOeffnen("localhost", "test", "1234") ) return ;

                db.conExecute("CREATE Database IF NOT EXISTS Formel1;");
                db.conExecute("USE Formel1");

                db.conExecute("DROP TABLE IF EXISTS fahrer ;");
```

```
db.conExecute("CREATE TABLE  fahrer (" + "Name VARCHAR(50) NOT NULL,"
                                      + "Geburtsjahr Integer ," + "PRIMARY KEY
                                      (Name)" + "          );");

        db.conExecute("INSERT INTO fahrer VALUES('Sebastian Vettel',1987);");
        db.conExecute("INSERT INTO fahrer VALUES('Mark Webber',1976 );");
        db.conExecute("INSERT INTO fahrer VALUES('Jenson Button',1980);");
        db.conExecute("INSERT INTO fahrer VALUES('Rubens Barrichello',1972);");
        db.conExecute("INSERT INTO fahrer VALUES('Lewis Hamilton',1985);");
        db.conExecute("INSERT INTO fahrer VALUES('Kimi Raeikkoenen',1979);");
        db.conExecute("INSERT INTO fahrer VALUES('Nick Heidfeld',1977);");
```

```
db.conAbfrage("SELECT f.Name , t.Bezeichnung FROM fahrer f ,
                                    faehrt_bei a  , team t"
                    + " where (f.Name = a.Name) AND
                                        (a.Bezeichnung =t.Bezeichnung) ;");

int anzahl = db.anzahlDatensaetze();
System.out.println("Anzahl der Datensaetze Fahrer - Team: " + anzahl);

while (db.neuerDatensatz()) {
          System.out.println("     " + db.getString("Name")
                                        + " --- "
                                        + db.getString(2));
}


           db.conClose();
 }
```

Dazu nötig:

- Datenbank
  (z.B. xampp (Webserver Apache + MySQL-Datenbank))
- Aktueller Driver für den Datenbankzugriff
  (MySQL Connector mysql-connector-java-x.y.z-bin.jar)
  muss dem ClassPath hinzugefügt werden.