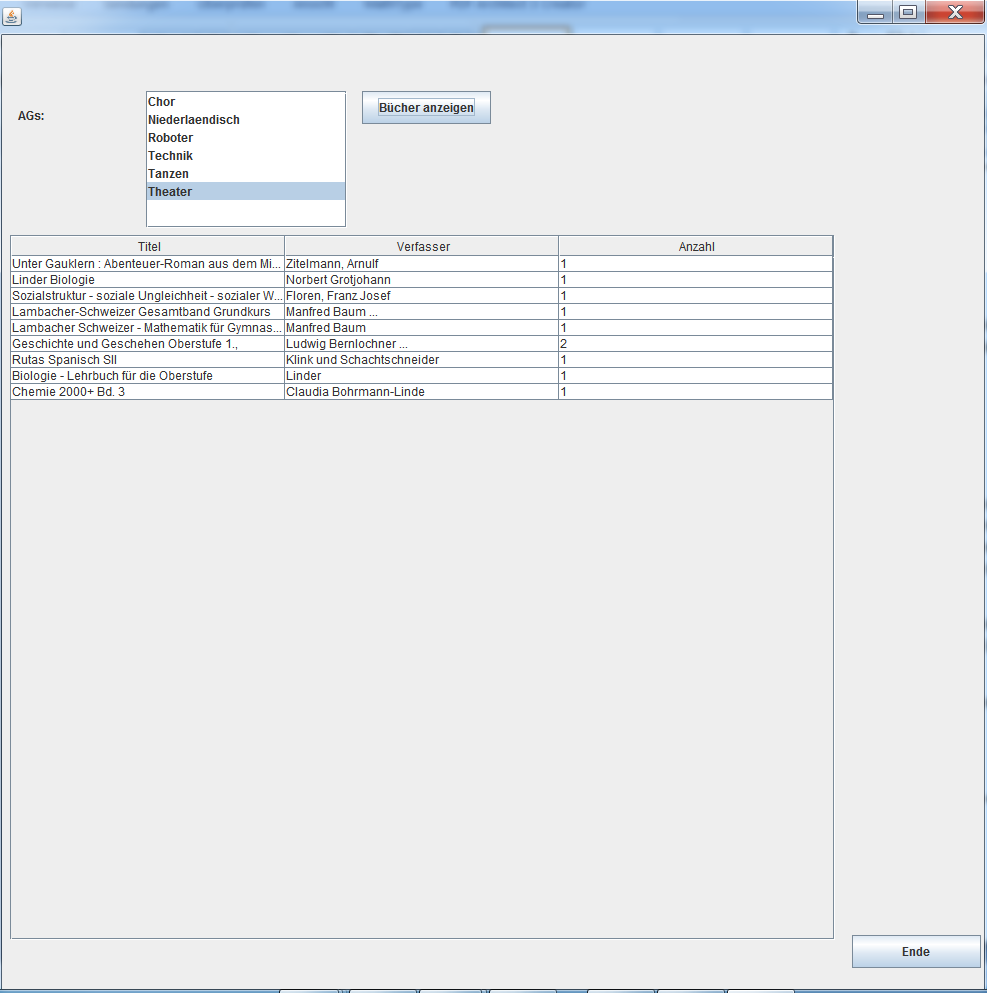
**Anwendung BuecherlisteAG**

Zur Verfügung stehen eine Datenbank BIB einer Schulbuchausleihe sowie und die Datenbank AGVerwaltung mit der in der gleicehn Beispielschule die Daten der AGen verwaltet werden. Beide Datenbanken sind in SQLite realisiert. Während die Daten der Datenbank BIB ursprünglich den Daten einer tatsächlichen Schule entsprachen und damit eine hohe Anzahl an Datensätzen zur Verfügung stellt (wobei die Schülerdaten so verwürfelt wurden, so dass keine Zuordnung zu den ursprünglichen Daten mehr möglich ist), wurde die Datenbank AGVerwaltung extra für diese Anwendung erstellt. Dem entsprechend finden sich hier nur einige Beispieldatensätze. Natürlich kann sie gerne ergänzt werden.

Wenn das Progarmm durch Ausführung der Datei BuecherlistenStart gestartet wird, werden die beiden oben beschriebenen Datenbanken geöffnet. Es erscheint ein Fenster, in dem die vorhandenen AGen angezeigt werden.

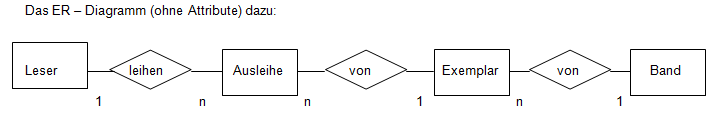
Nach Auswahl einer AG und Klick auf die Schaltfläche mit der Aufschrift „Bücher anzeigen“ wird im unteren Teil des Fensters eine Liste der Bücher angegeben, die von Teilnehmerinnen und Teilnehmer das Buch gelesen haben. (Unterchiedliche Auflagen werden hier nicht unterschieden.) Falls keine AG angeklickt wurde, wird nichts ausgegeben.

Da die AGen „Tanzen“, „Technik“ und „Niederländisch“ in der Datenbank noch keine Mitglieder haben, werden von Mitgliedern dieser Agen zur Zeit noch keine Bücher gelesen. Es ist also richtig, wenn hier keine Bücher angezeigt werden.

Durch Anklicken der Schaltfläche mit der Aufschrift „Ende“ wird das Programm beendet.

**Datenbanken:**

Die Datenbank BIB ist die gleiche, die auchin dem Projekt „Schulbücherei“ verwendet wird.



Datenbankschema:

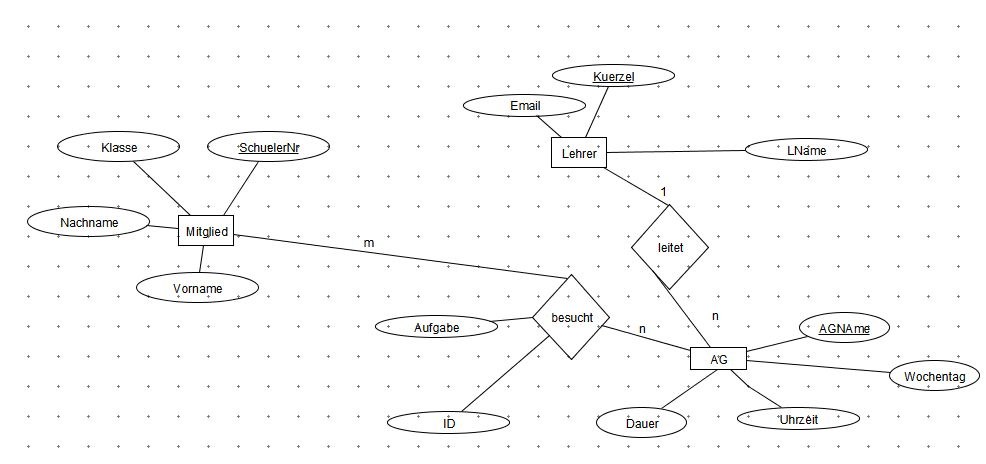
**Leser**(benutzer\_id, nachname, vorname, klasse, Geburtsdatum, Ort, PLZ, Strasse, Hausnummer)

**Ausleihe**(ausleih\_id,↑benutzer\_id,↑exemplar\_id,ausleih\_datum,rueckgabe\_datum, faellig\_datum,status)

**Band**(band\_id,ean13,titel,verfasser,auflage)

**Exempla**r(exemplar\_id,↑band\_id,barcode,zustand,flags)

Die Datenbank AGVerwaltung entspricht dem folgenden ER-Diagramm :



Datenbankschema:

**AG**(AGName,Wochentag,Dauer,Uhrzeit, ↑Lehrerkuerzel)

**Lehrer**(Kuerzel, LName, Email)

**Mitglied**(SchuelerNr, Nachname,Vorname, Klasse)

**besucht**(ID, ↑ AGName, ↑SchuelerNr, Aufgabe)

Der Fremdschlüssel Lehrerkuerzel verweist auf den Lehrer, der die AG leitet, in der Tabelle besucht verweisen AGName und SchuelerNr auf das Mitglied und die AG, die von dem Schüler mit der entsprechenden Nummer besuct wird.

Die Java-Anwendung besteht aus einer kurzen Startdatei BuecherlistenStart, einer Klasse, mit der das Ein- und Ausgabefenster realisiet wird, einer Klasse KlickEmpfaenger, mit der die eigentliche Funktion der Anwendung (d.h. die Reaktionen auf das Klicken einer Schaltfläche) realisiert wird, einer Klasse Leser von Objekten, in denen jeweils die Daten eines Lesers gespeichert und wieder abgerufen werden können, einer Klasse Buch von Objekten, in denen jeweils für ein Buch Titel, Verfasser und die Anzahl der Leser aus der gewählten AG gespeichert werden können, sowie die Klassen Queue, List, DatabaseConnnector (für SQLite) und QueryResult aus den Materialien für das Zentralabitur NRW.

**Dokumentation der Java-Anwendung:**

**Dokumentation der Klasse Leser**

Ein Objekt der Klasse Leser verwaltet die Daten eines Lesers.

|  |  |
| --- | --- |
| Konstruktor | **Leser(nach:String, vor:String, kl:String, anz:String)**  Ein Objekt für einen Leser mit Nachnamen nach, Vornamen vor, Klasse kl und der Anzahl anz an Ausleihen wird erzeugt. |
| Anfrage | **nachname():String**  Der Nachname wird zurückgegeben. |
| Anfrage | **vorname():String**  Der Vorname wird zurückgegeben. |
| Anfrage | **klasse():String**  Die Klasse wird zurückgegeben. |

**Dokumentation der Klasse Buch**

Ein Objekt der Klasse Buch verwaltet von einem Buch den Titel, den Autor und die Anzahl der Mitglieder, die dieses Buch ausgeliehen haben.

|  |  |
| --- | --- |
| Konstruktor | **Buch(ti:String, au:String)**  Ein Objekt für ein Buch mit dem Titel ti und dem Verfasser au wird erzeugt. |
| Auftrag | **erhoehe()**  Die Anzahl der erfassten Mitglieder der ausgewählten AG, die dieses Buch ausgeliehen haben, wird um eins erhöht. |
| Anfrage | **titel():String**  Der Titel des Buchs wird zurückgegeben. |
| Anfrage | **Autor: String**  Der Autor des Buchs wird zurückgegeben. |
| Anfrage | Der Anzahl der erfassten Mitglieder der ausgewählten AG, die dieses Buch ausgeliehen haben, wird zurückgegeben. |

**Dokumentation der Klasse Buecherliste**

Ein Objekt der Klasse Buecherliste erzeugt und verwaltet das Fenster der Anwendung.

|  |  |
| --- | --- |
| Konstruktor | **Buecherliste()**  Ein Fenster für die Anwendung wird erzeugt.Um die Auswahl de AGen angeben zu können, wird ein Objekt der Klasse DatabaseConnector verwendet um die Namen der AGen mit Hilfe einer SQL-Anweisung aus der entsprechenden Datenbank zu erfragen. |
| Auftrag | **schliesseDatenbankverbindungundProgramm() ()**  Die Verbindungen zu der Datennbank wirden geschlossen und die Java-Anwendung beendet. |
| Auftrag | **zeigeBuchListe(List<Buch> li)**  Die Daten der in der Liste li gespeicherten Bücher werden in einer Tabelle im Fenster angezeigt. Dabei werden eventuell vorhandene borherige Ausgaben gelöscht. |
| Anfrage | **String aktAG()**  Falls eine Ag ausgewählt wurde, wird der Name dieser AG zurückgegeben, ansonsten wird null zurückgegeben. |

**Dokumentation der Klasse Klickempfaenger**

Ein Objekt der Klasse Klickempfaenger realisiert die Reaktionen auf einen Klick auf eine Schaltfläche.

Hierfür werden zwei Objekte der Klasse DatenbankConnector für die Datenbankanbindungen an die AG-Datenbank und die Bibliotheksdatenbank verwendet, sowie eine Liste, um die Daten der Mitglieder der Ag zwischenzspeichern und eine Liste mit den Informationen über die betreffenden Bücher. Außerdem besteht eine Assoziation zur Oberfläche (einem Objekt einer Klasse Buecherliste), um Ein- und Ausgaben realisieren zu können.

|  |  |
| --- | --- |
| Konstruktor | **KlickEmpfaenger(Buecherliste d)**  Die Datenbankanbindungen werden realisiert. Falls dies nicht gelingt erscheinen Fehlermeldungen, ansonsten erscheinen Bestätigungen im Systemfenster. |
| Auftrag | **beenden()**  Alle Verbindungen zu den Datennbanken werden geschlossen und das Programm beendet. |
| Anfrage | **List<Leser> ermittleTeilnehmer(String ag)**  Auf der Basis der Daten in der Datenbank AGVerwaltung wird eine Liste dr Teilnehmer der AG ag erstellt und zurückgegeben. |
| Anfrage | **List<Buch> geleseneBuecherVon(List<Leser> li)**  Auf der Basis der Daten in der Datenbank BIB wird eine Liste der Bücher erstellt und zurückgegeben, die von den in der Lister li gespeicherten Lesern ausgeliehen wurden. Dabei wird jeder Titel nur einmal gespeichert ggf. jedoch auch die Anzahl von wie vielen Lesern aus der Liste das Buch gelesen wurde aktualisiert. |
| Auftrag | **void actionPerformed(ActionEvent e)**  Fallsdie Schaltfläche mit der Aufschrift „Bücher anzeigen“ wird im unteren Teil des Fensters eine Liste der Bücher angegeben, die von Teilnehmerinnen und Teilnehmer das Buch gelesen haben. (Unterchiedliche Auflagen werden hier nicht unterschieden.) Falls keine AG angeklickt wurde, wird nichts ausgegeben. Falls die Schaltfläche mit der Aufschrift „Ende“ angeklickt wurde, werden die Datenabnkanbindungen gechlossen und das Programm beendet. |