

Klassendiagramm und Relationales Datenmodell

Die Programmierung setzt auf folgende Datenstrukturen auf:

Relationales Datenmodell der Datenbank:	Klassendiagramm Schueler
schueler (<u>id</u> , name, vorname, ↑klasse_id) klasse (<u>id</u> , name, ↑klassenlehrer_id) unterricht (<u>id</u> , ↑klasse_id, ↑lehrer_id, ↑raum_id, fach, stunden) lehrer (<u>id</u> , name, vorname) raum (<u>id</u> , nummer, etage, plaetze)	<div style="border: 1px solid black; padding: 5px;"> <div style="border-bottom: 1px solid black; padding-bottom: 5px;">Schueler</div> <div style="padding: 5px;"> - name: String - vorname: String - klasse: String </div> <div style="padding: 5px;"> + Schueler(pName: String, pVorname: String, pKlasse: String) + getName(): String + getVorname(): String + getKlasse(): String + toString(): String </div> </div>

```
public int anzahlDerStundenVon(String pFach) {
    String sqlStatement =
        _____
        _____
        _____

    System.out.println(sqlStatement);
    connector.executeStatement(sqlStatement);
    QueryResult queryResult = connector.getCurrentQueryResult();
    String[][] data = queryResult.getData();
    String summeString = _____;
    int summe = _____;
    return summe;
}
```

```
public List<Schueler> schuelerDerKlasse(String pKlasse){
    List<Schueler> ergebnis = new ListWithViewer<Schueler>();
    String sqlStatement =
        _____
        _____
        _____
        _____

    connector.executeStatement(sqlStatement);
    QueryResult queryResult = connector.getCurrentQueryResult();
    String[][] data = queryResult.getData();
    for(_____ ) {
        String name = _____;
        String vorname = _____;
        String klasse = _____;
        Schueler s = _____;
        _____;
    }
    return ergebnis;
}
```