# 3. Übung: OOP - UML



### 1. Ziel

Erlernen des Arbeitens mit UML Klassendiagrammen und Umsetzung eines Klassendiagramms in ein Java Programm.

#### 2. Punkte

Bei dieser Aufgabe gibt es maximal **100 Punkte** zu erreichen. Die Punkte teilen sich wie folgt auf die Teilaufgaben auf:

Nr.	Name	Punkte
1	Implementierung Klassendiagramm	60
2	Applikation HTL	20
3	Präsentation	20

### 3. Abgabetermin

Siehe Microsoft Teams.

### 4. Abgabeform

Die Abgabe soll ein komprimiertes Verzeichnis (\*.zip, \*.7z, \*.tar) sein, in dem sich die entwickelten Java Dateien (\*.java) befinden.

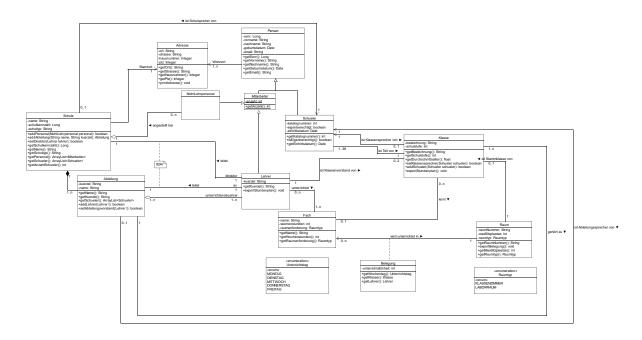
## 5. Aufgabenstellung

Diese Aufgabe ist in Teams zu je zwei Personen zu erledigen.

Gegeben ist folgendes UML Klassendiagramm (separat auch in Teams in Aufgabe 3 hochgeladen – schule.svg).

## 3. Übung: OOP - UML





### Teilaufgaben:

- 1. **Umsetzung Klassendiagramm**: Setzt das gegebene Klassendiagramm in Java Code um und beachtet dabei die besprochenen Umsetzungsrichtlinien. Das angegebene Klassendiagramm entspricht einer Minimalanforderung und kann nach eigenem Ermessen erweitert werden.
- 2. **Applikation HTL**: Erstellt eine zusätzliche Klasse (z.B. HTL.java) in der eine main() Methode enthalten ist. Darin soll eine Schule (die HTL) mit den zugehörigen Abteilungen (mindestens zwei) erzeugt werden. Von den restlichen Klassen sollen jeweils zumindest 5 Instanzen (wo sinnvoll/notwendig auch mehr) erzeugt werden, um die volle Funktionalität abzudecken.

Zusätzlichen sollen die verschiedenen Positionen besetzt werden (Direktor, Abteilungsvorstände, Schulsprecher, Abteilungssprecher, Klassensprecher, Klassenvorstand, Stammklasse) und eine Raumbelegung erzeugt werden.

Im Anschluss sollen die Methoden Schule.getAnzahlSchueler(), Lehrer.exportStundenplan(), Klasse.exportStundenplan(), Raum.exportBelegung() auf jeweils einer Instanz aufgerufen werden, um die Funktionalität zu zeigen.

3. **Präsentation**: Nach Fertigstellung der Implementierung stellt ihr eure Lösung der Klasse vor. Im Rahmen der Präsentation soll der Code gezeigt und auf spezielle Umsetzungen eingegangen werden.