

### 3 Arbeiten mit UML

Ihnen wird die Entwicklung einer Musik-App namens *LinAmp* übertragen und sollen neue Features planen und umsetzen. Den Code für die Applikation können Sie unter folgendem Link herunterladen:

<https://git.noc.rub.de/herbepdl/propro2023-linampplayer>

Alternativ kann das Projekt mit folgendem Befehl direkt mit Git geklont werden:

```
git clone https://git.noc.ruhr-uni-bochum.de/herbepdl/propro2023-linampplayer.git
```

Der *LinAmp*-Player kann Alben und Tracks verwalten und abspielen. In der `main()`-Methode der Klasse `Player` ist eine beispielhafte Ausführung implementiert für das Erstellen und Wiedergeben von Alben.

Die Entwicklung dieser Applikation ist bisher ohne UML erfolgt. Zuerst sollten Sie sich also einen Überblick über den Code verschaffen, indem Sie ein UML-Klassendiagramm zu dem Projekt erstellen.



**Tipp**

Zum Erstellen von UML-Diagrammen können Sie z.B. die Webapplikation draw.io verwenden: <https://draw.io>

#### Aufgaben

1. Erstellen Sie ein Klassendiagramm für `Player`, `Album`, und `Track`. Achten Sie auf sinnvolle Multiplizitäten.
2. Die Applikation hat einige Designprinzipien nicht beachtet. Passen Sie das Klassendiagramm so an, dass das Geheimnisprinzip gewahrt wird.
3. Die Information zu einem Artist wird derzeit im Album/Track gespeichert, was zu duplizierten Daten führt. Trennen Sie im Klassendiagramm die Daten zum Artist von den Alben/Tracks.
4. Album und Track beinhalten viel duplizierten Code. Erweitern Sie das Klassendiagramm um eine gemeinsame Oberklasse oder Interface.
5. Die Applikation soll ein Playlist-Feature bekommen. Erweitern Sie das Klassendiagramm um eine Playlist, welche Tracks von verschiedenen Alben und Artists beinhalten kann. Eine Playlist soll wie ein Album im Player abspielbar sein.
6. Setzen Sie die Änderungen die Sie mit dem UML-Klassendiagramm geplant haben in Programmcode um.