

Im Rahmen einer Softwareentwicklung für die einheitliche Verwaltung von Flugzeugen wurde die Klasse *Flugzeug* entwickelt. Die Klasse *Flugzeug.java* findet ihr in unserem gemeinsamen Ordner.

Ausgehend von dieser Klasse sind zwei Klassen *Verkehrsflugzeug* und *Doppeldecker* zu modellieren. Diese sollen folgende Spezifikationen erfüllen:

Klasse *Verkehrsflugzeug*:

- Ein Verkehrsflugzeug ist ein Flugzeug, das genau ein **Flügelpaar** sowie eine zusätzliche Variable für die **Anzahl der Passagiere** hat.
- Ein Verkehrsflugzeug fliegt keine Loopings. Stellen Sie deshalb sicher, dass die Methode **getLooping** immer **false** zurückgibt, auch in allen Unterklassen von Verkehrsflugzeug.
- Ein Verkehrsflugzeug-Objekt kann mit **Angabe des Herstellers** (String), der **maximalen Geschwindigkeit** (int), der **Immatrikulationsnummer** (String) und der **Anzahl** der Passagiere (int) erzeugt werden.
- Die Klasse Verkehrsflugzeug hat die Methoden **getAnzahlPassagiere** und **setAnzahlPassagiere** zum Abfragen und Setzen der Anzahl der Passagiere.

Klasse *Doppeldecker*:

- Ein Doppeldecker ist ein Flugzeug, das genau zwei Flügelpaare hat.
- Weiter ist ein Doppeldecker akrobatiktauglich, d.h. man kann damit Loopings fliegen. Für einen Looping muss der Doppeldecker eine Mindestgeschwindigkeit von 320 km/h erreichen. Definieren Sie dafür eine Konstante **LOOPINGSPEED**. Die Methode **getLooping** soll **true** zurückgeben, falls die zulässige max. Geschwindigkeit (**maxSpeed**) grösser **LOOPINGSPEED** ist.
- Die Klasse Doppeldecker hat eine Variable **offenesCockpit** vom Typ **boolean**. Sie gibt an, ob der Doppeldecker ein offenes oder geschlossenes Cockpit hat. Nachdem **offenesCockpit** gesetzt worden ist, darf sie nicht mehr verändert werden. Der Wert des Attributes darf gelesen werden.

- Ein Doppeldecker-Objekt kann durch 2 Konstruktoren initialisiert werden:
 - Der 1. Konstruktor hat folgende Parameter: Hersteller (String), maximale Geschwindigkeit (int), die Immatrikulationsnummer (String) und einen boolean `offenesCockpit`, der angibt, ob der Doppeldecker ein offenes oder geschlossenes Cockpit hat.
 - Der 2. Konstruktor hat folgende Parameter: Hersteller (String), maximale Geschwindigkeit (int), die Immatrikulationsnummer (int). Der Defaultwert für `offenesCockpit` bei einem Doppeldecker ist `true`.
 - Die Klasse `Doppeldecker` soll nicht erweiterbar sein.
1. Modellieren Sie die beschriebenen Klassen in einem UML-Klassendiagramm.
 2. Implementieren Sie die Klassen *Verkehrsflugzeug* und *Doppeldecker*.