

## Vererbung

(20 Punkte)

Im Rahmen einer Softwareentwicklung für die einheitliche Verwaltung von Flugzeugen wurde die Klasse `Flugzeug` entwickelt. Die Klasse `Flugzeug` ist auszugsweise angegeben:

```
abstract class Flugzeug {
    private String hersteller;           // Herstellername
    private int maxSpeed;                // Max. Geschwindigkeit
    private String immatNummer;          // Immatrikulationsnummer
    private int anzahlFluegel = 1;       // Anzahl Flügelpaare

    public Flugzeug(String hersteller, int maxSpeed, int anzahlFluegel) {
        this.hersteller = hersteller;    this.maxSpeed = maxSpeed;
        this.anzahlFluegel = anzahlFluegel;
    }
    public String getImmatNummer() {
        return immatNummer;
    }
    protected void setImmatNummer(String immatNummer) {
        this.immatNummer = immatNummer;
    }
    public int getMaxSpeed() {
        return maxSpeed;
    }
    abstract public boolean getLooping();

    // weitere Methoden
}
```

Schreiben Sie ausgehend von dieser Klasse eine Klasse `Verkehrsflugzeug` sowie eine Klasse `Doppeldecker`. Diese sollen folgende Spezifikation 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 allfälligen Unterklassen von `Verkehrsflugzeug`.
- Schreiben Sie einen Konstruktor, mit dem ein `Verkehrsflugzeug`-Objekt erzeugt werden kann. Dazu müssen der Hersteller (`String`), die maximale Geschwindigkeit (`int`), die Immatrikulationsnummer (`String`) und die Anzahl Passagiere (`int`) angegeben 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.
- Schreiben Sie eine Methode `isOffenesCockpit`, die den Wert von `offenesCockpit` zurückgibt.
- Schreiben Sie zwei Konstruktoren, mit denen ein `Doppeldecker`-Objekt initialisiert werden kann.
  - Der 1. Konstruktor hat folgende Parameter:  
Hersteller (`String`), maximale Geschwindigkeit (`int`), die Immatikulationsnummer (`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 Immatikulationsnummer (`int`). Der Defaultwert für `offenesCockpit` bei einem Doppeldecker ist `true`.
- Die Klasse `Doppeldecker` soll nicht erweiterbar sein.

a) (8 Punkte) Implementieren Sie die vollständige Klasse `Verkehrsflugzeug` gemäss der oben angegebenen Spezifikation.

b) (12 Punkte) Implementieren Sie die vollständige Klasse `Doppeldecker` gemäss der oben angegebenen Spezifikation.