

## Aufgabe Auto & Pickup

Erstellen Sie zuerst ein Klassendiagramm für die Aufgabe!

Modellieren Sie die Klassen Auto und Pickup nach den folgenden Vorgaben. Eigenschaften von Objekten dieser Klassen sollen ausschließlich beim Erzeugen gesetzt und ggf. durch Methoden verändert werden können.

### Klasse Auto:

Ein Auto soll ein Kennzeichen, einen Kilometerstand und eine Anzahl Sitzplätze haben. Wenn nicht anders angegeben, ist das Kennzeichen eines Auto-Objekts standardmäßig »DO-OM 3«, es sollen aber auch Autos mit anderen Kennzeichen erzeugt werden können. Ein neu erzeugtes Auto hat grundsätzlich noch keine Kilometer gefahren. Wenn nicht anders angegeben wird hat ein Auto 5 Sitzplätze. Es sollen jedoch auch zweisitzige Autos erzeugt werden können. Zu guter Letzt hat ein Auto eine Antenne, die aus- oder eingefahren werden/sein kann. Die Klasse soll mindestens die folgenden Methoden besitzen:

- `GetKennzeichen()` : liefert das Kennzeichen als String zurück
- `GetKilometerstand()` : liefert den aktuellen Kilometerstand zurück
- `Fahre(int kilometer)` : schreibt eine Meldung an die Standardausgabe, dass das Auto x Kilometer fährt, und setzt dann den Kilometerstand entsprechend weiter
- `AntenneEinfahren()` : schreibt eine Meldung an die Standardausgabe, dass die Antenne eingefahren wird, und setzt den Status der Antenne
- `AntenneAusfahren()` : schreibt eine Meldung an die Standardausgabe, dass die Antenne ausgefahren wird, und setzt den Status der Antenne
- `IstAntenneDraussen()` : liefert true, wenn die Antenne ausgefahren ist ansonsten false
- `VorDemWaschen()` : bereitet das Auto auf den Aufenthalt in der Waschstraße vor – konkret bedeutet das bei einem Auto, dass die Antenne eingefahren wird
- `Waschen()` : bereitet das Auto für eine Autowäsche vor und schreibt dann eine Meldung an die Standardausgabe, dass der Wagen gewaschen wird
- `GetSitzplätze()` : liefert die Anzahl Sitzplätze zurück
- `ToString()` : gibt eine passende String-Repräsentation des Autos zurück

### Klasse Pickup:

Ein Pickup ist eine spezielle Art von Auto. Es hat nur 2 Sitzplätze, dafür jedoch zusätzlich eine offene Ladefläche mit einem beim Erzeugen bestimmaren Fassungsvermögen. Die Ladefläche ist beim Erzeugen des Pickups zunächst leer und kann maximal mit x Dingen beladen werden. Die Klasse muss mindestens folgende Methoden besitzen:

- `Beladen(int ladungsMenge)` : falls auf der Ladefläche noch genügend Platz ist, wird der Inhalt der Ladefläche um x Dinge erhöht, eine passende Meldung an die Standardausgabe geschrieben, und `true` zurückgegeben, andernfalls `false`
- `Entladen(int ladungsMenge)` : falls auf der Ladefläche mindestens x Dinge vorhanden sind, wird der Inhalt der Ladefläche um x Dinge verringert, eine passende Meldung geschrieben, und `true` zurückgegeben, andernfalls `false`
- `VorDemWaschen()` : bei einem Pickup soll vor dem Waschen nicht nur die Antenne eingefahren werden, sondern auch die Ladefläche geleert werden
- `GetLadung()` : liefert die Anzahl Dinge, die derzeit auf der Ladefläche liegen, zurück, ohne den Ladezustand zu verändern
- `ToString()` : gibt eine passende String-Repräsentation des Pickups zurück

Definieren Sie in der Main-Methode mehrere verschiedene Autos und Pickups. Dort ist unter anderem Folgendes zu implementieren:

- Lassen Sie sich die Kennzeichen und sonstigen Zustände der erzeugten Objekte ausgeben
- Lassen Sie die Autos und Pickups fahren und waschen
- Testen Sie die Beladelogik der Pickups