

Gewerbliche Schule Bad Mergentheim		
E3FI - BT	Vererbung	
OStR Bauer	Projekt „Fuhrpark“	

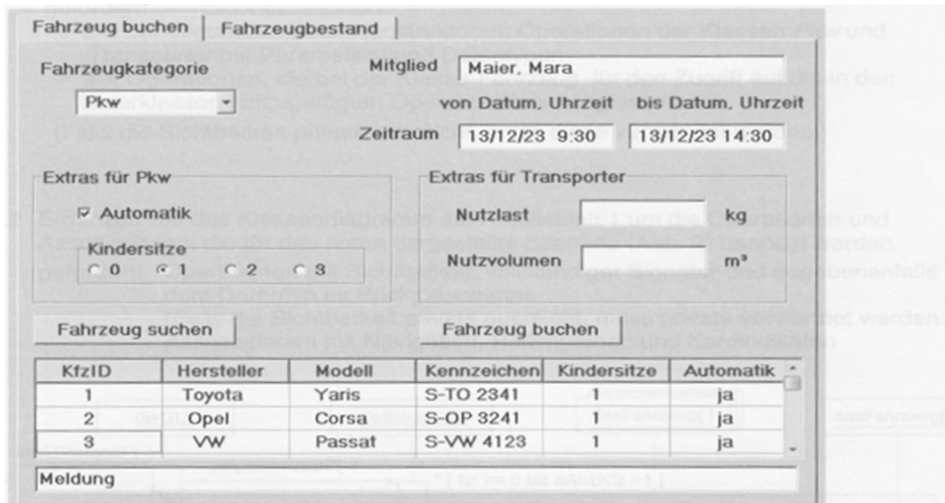
Der Car-Sharing-Verein Citymobil plant die Entwicklung einer Software zur Verwaltung seiner Fahrzeuge. Die Verwaltungssoftware benötigt deshalb die Klassen *Pkw* und *Transporter*. Ergänzen Sie diese im vorgegebenen Klassendiagramm.

Hierfür gelten folgende Festlegungen:

- Pkw- und Transporter-Objekte müssen gemeinsam in einer Liste verwaltet werden.
- Pkw und Transporter benötigen beide alle Attribute der Klasse Fahrzeug.
- Pkw können jedoch zusätzlich mit Kindersitzen ausgestattet sein und ein Automatik- oder ein Schaltgetriebe haben.
- Bei Transportern soll die Nutzlast und das Nutzvolumen gespeichert werden.

Aus Sicht der Mitglieder des Vereins kann man mit dem Programm:

- Den Fahrzeugbestand von einer **CSV-Datei** einlesen und anzeigen.  
(wahlweise: alle Pkw oder alle Transporter oder alle verfügbaren Fahrzeuge)
- Fahrzeuge suchen, die bestimmte Kriterien erfüllen (s. Abb. 1).
- Ein Fahrzeug für einen gewünschten Zeitraum buchen.
- Alle Buchungen laden/ speichern.

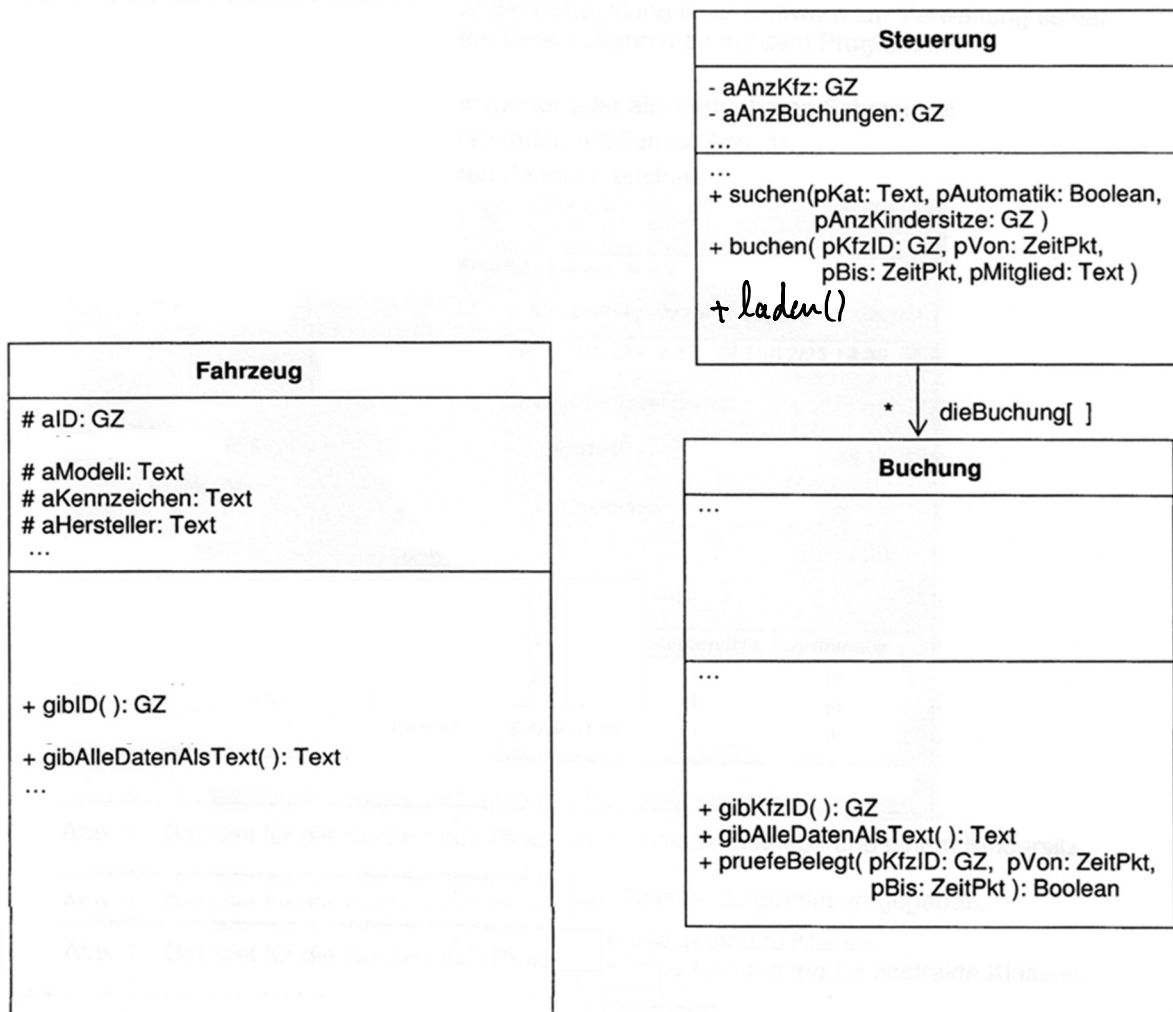


KfzID	Hersteller	Modell	Kennzeichen	Kindersitze	Automatik
1	Toyota	Yaris	S-TO 2341	1	ja
2	Opel	Corsa	S-OP 3241	1	ja
3	VW	Passat	S-VW 4123	1	ja

Abb. 1: Beispiel für die Suche nach Pkws mit Automatik-Getriebe und einem Kindersitz

Gewerbliche Schule Bad Mergentheim		
E3FI - BT	Vererbung	
OStR Bauer	Projekt „Fuhrpark“	

## Klassendiagramm



1.2 Klassendiagramm: Fahrzeug 1.2.1 Fahrzeug-Klasse 1.2.1.1 Attribute 1.2.1.2 Methoden 1.2.1.3 Zusammenfassung
--

1.2 Klassendiagramm: Fahrzeug 1.2.2 Fahrzeug-Klasse 1.2.2.1 Attribute 1.2.2.2 Methoden 1.2.2.3 Zusammenfassung
--

Gewerbliche Schule Bad Mergentheim		
E3FI - BT	Vererbung	
OStR Bauer	Projekt „Fuhrpark“	

### Buchung eines Fahrzeugs

<u>dieBuchung[12]: Buchung</u>	<u>dieBuchung[23]: Buchung</u>
aKfzID = 14 vonZeitPkt = 2023-10-10 12:45 bisZeitPkt = 2023-10-10 22:00 aMitglied = "Maier, Mara"	aKfzID = 8 vonZeitPkt = 2023-12-15 08:00 bisZeitPkt = 2023-12-17 17:00 aMitglied = "Bauer, Paul"

Abb. 3: Beispiele für Objekte der Klasse *Buchung*

- a) Mit den Objekten der Klasse Buchung (s. Abb.3) werden die getätigten Fahrzeugbuchungen verwaltet. Ergänzen Sie auf dem Arbeitsblatt 1 bei der Klasse Buchung die Attribute. Durch den Konstruktor werden alle Attribute initialisiert.
- b) Eine Buchung für ein Fahrzeug kann genau dann getätigt werden, wenn noch keine Buchung mit überschneidenden Zeitdaten für das gewünschte Fahrzeug existiert. Die Klasse Buchung besitzt die Methode *pruefeBelegt* ( *pKfzID: GZ, pVon: ZeitPkt, pBis: ZeitPkt*): *Boolean*. Diese Methode überprüft, ob es eine Terminüberschneidung zwischen dem als Parameter übergebenen, gewünschten Buchungszeitraum und dem im angesprochenen Buchungsobjekt gespeicherten Buchungsdaten gibt. Falls sich die Buchung auf das gewünschte Fahrzeug bezieht und es eine Terminüberschneidung gibt, antwortet die Methode mit *true*. Falls es keine Terminüberschneidung gibt, antwortet die Methode mit *false*. Entwickeln Sie den Algorithmus der Methode *pruefeBelegt*( ) und stellen Sie diesen in einem Struktogramm dar. Implementieren Sie die Methode *pruefeBelegt*( ) in der an Ihrer Schule verwendeten Programmiersprache.
- Lösungshinweis: Es liegt keine Terminüberschneidung vor, wenn das Ende des gewünschten Buchungszeitraums vor dem Beginn des gespeicherten Buchungszeitraums oder wenn der Beginn des gewünschten Buchungszeitraums nach dem Ende des gespeicherten Buchungszeitraums liegt.
- c) Die Methode *buchen*(*pKfzID: GZ, pVon:ZeitPkt, pBis:ZeitPkt, pMitglied: Text*) der Klasse *Steuerung* erzeugt, falls eine Buchung möglich ist, ein neues Buchungsobjekt mit den als Parameter übergebenen Daten. Eine Buchung ist jedoch nur möglich, wenn das gewünschte Fahrzeug im gewünschten Buchungszeitraum noch frei ist, also keine Terminüberschneidung mit einer schon vorhandenen Buchung besteht. Um sicherzustellen, dass keine Terminüberschneidung vorliegt, müssen in der Liste *dieBuchung* alle Buchungsobjekte überprüft werden. Verwenden Sie hierzu die Methode *pruefeBelegt*( ), die bei b) beschrieben ist. Wird beim Durchsuchen der Liste der Buchungen eine Terminüberschneidung gefunden, ist die Meldung „Buchung nicht möglich!“ auszugeben. Wird keine Terminüberschneidung festgestellt, wird ein neues Buchungsobjekt erzeugt und die Meldung „gebucht!“ angezeigt. Beachten Sie, dass das Attribut *aAnzBuchungen* die Anzahl der vorhandenen Buchungsobjekte enthält. Entwickeln Sie den Algorithmus der Methode *buchen*( ) und stellen Sie diesen in einem Struktogramm dar. Implementieren Sie die Methode *buchen*( ) in der an Ihrer Schule verwendeten