Objektorientierte Programmierung soll Teamarbeit unterstützen. Wir betrachten beispielhaft ein Projekt, bei dem eine Software für die Verwaltung des Fuhrparks einer Firma (also der firmeneigenen Fahrzeuge) entwickelt wird.

Ein Programmierer übernimmt dabei die Implementierung der **Klasse Fahrzeug**, deren Objekte die einzelnen Fahrzeuge des Fuhrparks darstellen. Für die Software sind dabei die Kilometerstände, die Größe der Benzintanks und der aktuelle Füllstand relevant. Der Kilometerstand läuft beim Fahren immer weiter, wird also nie auf 0 zurückgesetzt. Vereinfachend verbraucht jedes Fahrzeug pro 100 km 10 L Benzin.

Ein anderer Programmierer übernimmt die **Klasse Fuhrpark**, die folgenden Ablauf simulieren soll:

- Es werden zwei Fahrzeuge mit 50 L bzw. 70 L Tankgröße erzeugt.
- Die Fahrzeuge fahren eine Strecke von 450 km bzw. 550 km.
- Falls die Fahrzeuge danach einen Füllstand von weniger als 10 L haben, werden sie vollgetankt.

Der folgende Quellcode zeigt einen Vorschlag der beiden Programmierer:

Programmierer A

```
01  class Fuhrpark
02  {
03     Fahrzeug f1, f2;
04
05     Fuhrpark()
06     {
07        f1 = new Fahrzeug();
08        f2 = new Fahrzeug();
09        f1.tankGröße = -50.0;
10        f2.tankGröße =  70.0;
11     }
12
13     void main()
14     {
15        f1.kmStand += 450.0;
16        f2.kmStand += 550.0;
17
18        if (f1.füllstand < 10.0)
19        {
20           f1.füllstand += 50.0;
21        }
22        if (f2.füllstand < 10.0)
23        {
24           f2.füllstand += 70.0;
25        }
26     }
27  }
```

Programmierer B

```
01  class Fahrzeug
02  {
03     double tankGröße;
04     double füllstand;
05     double kmStand;
06  }
```

Aufgabe

a) Der Quellcode ist syntaktisch korrekt, kann also z.B. von BlueJ übersetzt werden. Programmierer A (Fuhrpark) sind aber einige **inhaltliche Fehler** unterlaufen.

   Beschreibe diese Fehler (mit Zeilenangaben).

b) Um solche inhaltlichen Fehler auszuschließen, soll für die Klasse Fahrzeug ein **besseres Modell** entworfen werden: Die Attribute der Klasse Fahrzeug sollen mit dem Zugriffsmodifikator „private“ versehen werden, und es sollen Methoden entwickelt werden, mit denen Fahrzeuge dann bearbeitet werden können, ohne dass solche Fehler auftreten können.

   i. Erläutere, warum es sinnvoll ist, die Fahrzeug-Attribute als private zu deklarieren.

   ii. Gib an, welche Regeln für die Attribute der Klasse Fahrzeug eingehalten werden müssen (gültige Wertebereiche, Zusammenhänge zwischen Attributen).

   iii. Überlege, welche Methoden und ggf. welche Parameter in der Klasse Fahrzeug nützlich wären, insbesondere, damit Programmierer A den Ablauf in der Klasse Fuhrpark implementieren kann.
   Gib jeweils den Methodenkopf und eine Beschreibung der Methode in Stichpunkten an.

c) **Implementiere** das neue Modell der Klasse Fahrzeug mit der bereitgestellten BlueJ-Vorlage.

d) Passe die Implementierung der Klasse Fuhrpark an, so dass die neuen Methoden der Klasse Fahrzeug verwendet werden. Auch in dieser Klasse gib die Zugriffsmodifikatoren an.

e) Zeichne ein **Klassendiagramm** zu deinem überarbeiteten Quellcode (mit den Zugriffsmodifikatoren).