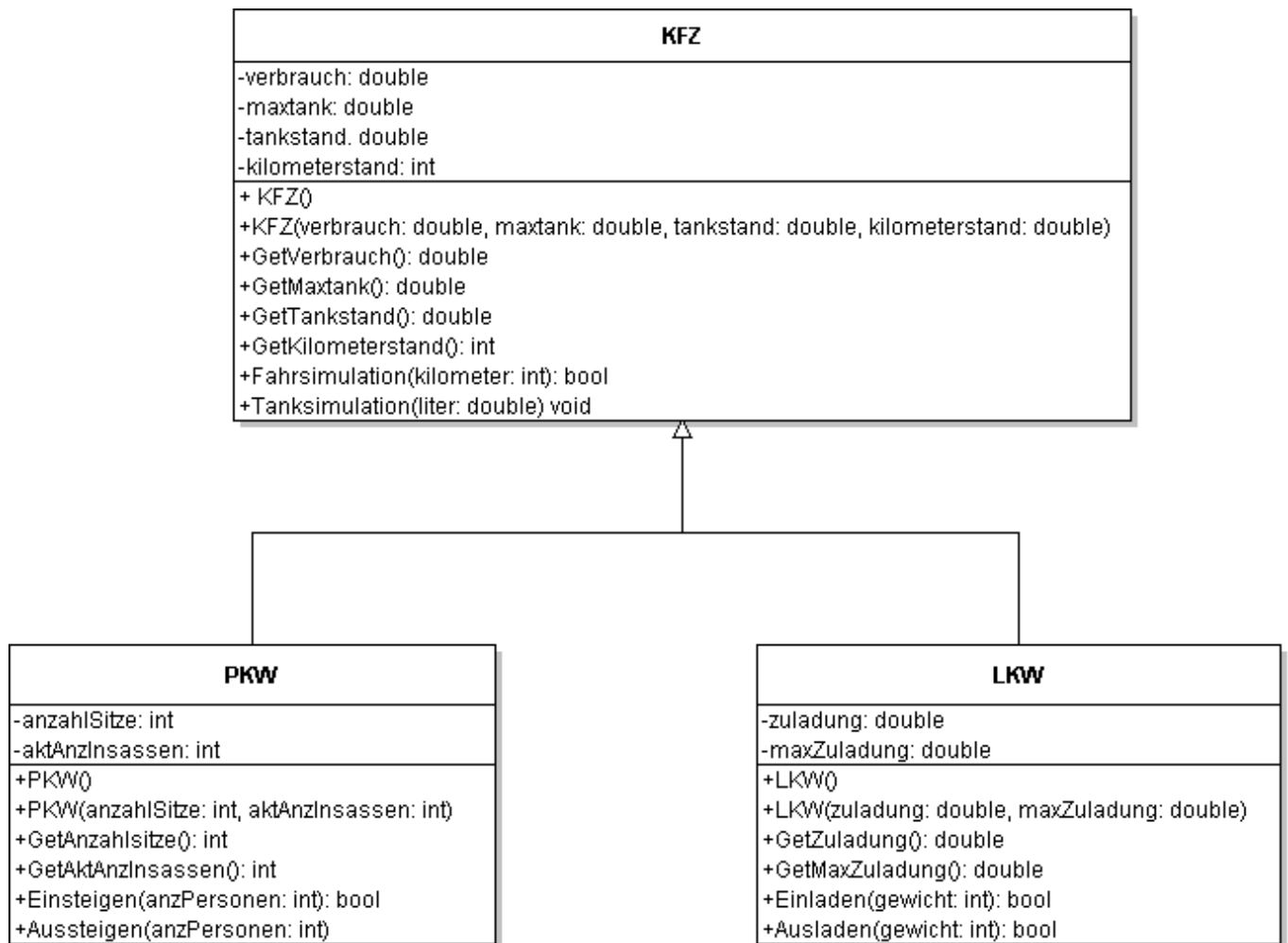


## UML - Klassenhierarchie

**Aufgabe 1:**

- 1.1 Deklarieren Sie die Klasse **KFZ**.
- 1.2 Implementieren Sie die Methoden der Klasse **KFZ**.

**Aufgaben der Methoden:**

Der Methode **Fahrsimulation** wird ein km-Wert übergeben, wobei die Methode den booleschen Wert true zurückgibt, falls die Kilometerstrecke mit dem aktuellen Tankstand zurücklegbar ist. Die entsprechenden Attribute müssen dann aktualisiert werden.

Der Methode **Tanksimulation** wird ein Literwert übergeben, womit diese einen Tankvorgang simuliert. Diese Methode besitzt keinen Rückgabewert.

**Aufgabe 2:**

- 2.1 Deklarieren Sie die Klasse **PKW**.  
Ein PKW besitzt eine festgelegte Anzahl von Sitzplätzen. Es soll außerdem noch die aktuelle Anzahl der Insassen im PKW gespeichert werden.
- 2.2 Implementieren Sie die Methoden der Klasse **PKW**.

**Aufgaben der Methoden:**

**Einsteigen:** Übergabeparameter ist die Anzahl der einzusteigenden Personen; Rückgabewert true, falls dies möglich ist, sonst false.

**Aussteigen:** simuliert das Aussteigen von Personen im PKW

**Aufgabe 2 Fortsetzung:**

- 2.3 Ergänzen Sie die Klassen PKW und LKW mit je einem weiteren überladenen Konstruktor so, dass alle Attribute der Klassenhierarchie individuell beim Erzeugen eines Objektes initialisiert werden können. Verwenden Sie dazu das Prinzip des **Basisinitialisierers in C#**.

**Aufgabe 3:**

- 3.1 Deklarieren Sie die Klasse **LKW**.  
Die Eigenschaft eines LKW ist, dass dieser gewisse Zuladung besitzt.
- 3.2 Implementieren Sie die Methoden der Klasse LKW. Die Methoden **Einladen** und **Ausladen** können analog interpretiert werden wie die Methoden Einsteigen und Aussteigen der Klasse PKW.

**Aufgabe 4:**

- 4.1 Erzeugen Sie in der statischen Main-Methode verschiedene Instanzen der Klasse PKW sowie der Klasse LKW.
- 4.2 Simulieren Sie die Eigenschaften der Objekte durch Aufruf der entsprechenden Methoden mit entsprechenden Ausgaben.