

UML-Klassendiagramme

Auto
+ farbe: string = "rot"
- leistung: int
<u>+ Starten() : bool</u>
- Beschleunigen (int) : int

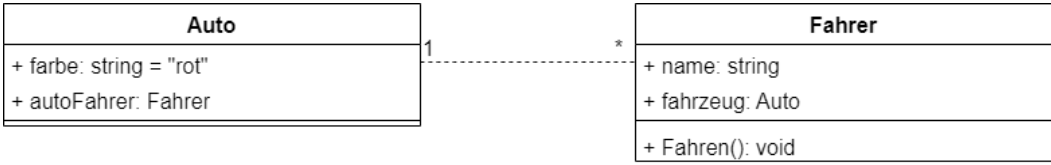

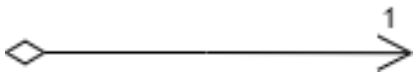


```
public class Auto
{
    public string farbe = "rot";
    private int leistung;

    public static bool Starten()
    {
        return true;
    }
    private int Beschleunigen(int wert)
    {
        return wert + 1;
    }
}
```

Aufgabe1: Ergänzen Sie die leeren Stellen in der nachfolgenden Tabelle. Formulieren Sie hierfür eine Beschreibung der einzelnen Elemente eines UML-Klassendiagramms.

Klassenbezeichnung	
Auto	
Attribute	
+	
-	
farbe / leistung	
string / int	
"rot"	
Methoden	
<u>+ Starten() : bool</u>	
Starten / Beschleunigung	
bool / int	
(int)	

Beziehungen zwischen Klassen


<pre> public class Auto { public string farbe = "rot"; // ... andere Eigenschaften und Methoden ... public Fahrer autoFahrer = new Fahrer(); // Assoziation-Beziehung } </pre>	<pre> public class Fahrer { public string name; public List<Auto> Fahrzeuge = new List<Auto>(); // Assoziation-Beziehung public void Fahren() { // Fahrer fährt das Auto } } </pre>
	
<p>-----</p>	<p>Assoziation (allgemeine Beziehung): Ein Fahrer ist mit dem Auto verbunden, weil der Fahrer das Auto steuert. Es ist eine bidirektionale Beziehung, da das Auto auch einen Fahrer "kennt".</p>
<pre> public class Auto { public string farbe = "rot"; // ... andere Eigenschaften und Methoden ... public Radio autoRadio; // Aggregation-Beziehung } </pre>	<pre> public class Radio { public void Spielen() { // Radio spielt Musik } } </pre>
	
	<p>Aggregation (schwache Beziehung): Ein Auto könnte eine Beziehung zu einem Radio haben. Das Radio kann ohne das Auto existieren, aber das Auto könnte durchaus ein Radio enthalten.</p>
<pre> public class Auto { public string farbe = "rot"; private Motor motor = new Motor(); // Komposition-Beziehung // ... andere Eigenschaften und Methoden ... } </pre>	<pre> public class Motor { public void Starten() { // Motor startet } } </pre>
	
	<p>Komposition (starke Beziehung): Ein Motor ist ein wesentlicher Bestandteil eines Autos. Ein Motor wird in der Regel nicht ohne das Auto existieren, in dem er eingebaut ist.</p>

Multiplizitätsangaben	
1	Genau eine Instanz (eine-zu-eins).
0..1	Keine oder eine Instanz (optional).
1..*	Mindestens eine Instanz, aber potenziell unendlich viele (eine-zu-viele).
1..*	Keine, eine oder viele Instanzen (optional viele).
*	Eine beliebige Anzahl von Instanzen, einschließlich null (viele-zu-viele).

Aufgabe 2: Erstellen sie ein UML-Klassendiagramm für folgenden Quellcode:
 Digitale UML-Diagramme können Sie mit www.draw.io erstellen.

Quellcode	
<pre> public class Bank { private List<Bankkonto> konto_liste = new List<Bankkonto>(); private Bankkonto aktives_konto = new Bank- konto(); public void MenueAufruf() { //... } } </pre>	<pre> public class Bankkonto { public double kontostand { get; set; } = 0; private int _kontonummer; public void Einzahlung(double einzahlbetrag) { //.. } public void Auszahlung(double abhebebetrag) { //.. } public void Kontostand() { //... } } </pre>

UML-Klassendiagramm

Übung zu den UML- Klassendiagrammen	System und- Anwendungssoftware	 Georg-Simon-Ohm Berufskolleg
Arbeitsblatt		