

Vererbung in JavaScript

Übung 1: Vererbung in Fahrzeugklassen

Erstelle eine Basisklasse `Vehicle` mit den Eigenschaften `id`, `status`, `latitude` und `longitude`. Dann erstelle eine abgeleitete Klasse `Car`, die zusätzlich die Eigenschaft `brand` hat.

1. Erstelle ein Objekt der Klasse `Car` mit `id = "C001"`, `brand = "Toyota"`, `latitude = 48.8566` und `longitude = 2.3522`.
 2. Gib die Eigenschaften `id`, `brand`, `latitude` und `longitude` in der Konsole aus.
-

Übung 2: Tiere und Vögel

Erstelle eine Basisklasse `Animal` mit der Eigenschaft `species` und einer Methode `makeSound`. Dann erstelle eine abgeleitete Klasse `Bird`, die zusätzlich die Methode `fly` hat.

1. Erstelle ein Objekt der Klasse `Bird` mit `species = "Eagle"`.
 2. Rufe die Methoden `makeSound` und `fly` auf.
-

Übung 3: Überschattung in Haushaltsgeräten

Erstelle eine Basisklasse `Appliance` mit einer Methode `turnOn`. Dann erstelle eine abgeleitete Klasse `Oven`, die die Methode `turnOn` überschreibt.

1. Erstelle ein Objekt der Klasse `Oven` und rufe die Methode `turnOn` auf.
 2. Verwende `super.turnOn()`, um die Methode der Basisklasse aufzurufen.
-

Übung 4: Vererbung von Konstrukturfunktionen

Erstelle eine Konstrukturfunktion `Device` mit der Eigenschaft `model`. Dann erstelle eine Klasse `Phone`, die von `Device` erbt und eine zusätzliche Methode `call` hat.

1. Erstelle ein Objekt der Klasse `Phone` mit `model = "iPhone 12"`.
 2. Rufe die Methode `call` auf und gib das `model` in der Konsole aus.
-

Übung 5: Schattenmethoden in Fahrzeugklassen

Erstelle eine Basisklasse `Vehicle` mit der Methode `startEngine`. Dann erstelle eine abgeleitete Klasse `ElectricCar`, die die Methode `startEngine` überschreibt, aber auch die Methode der Basisklasse aufruft.

1. Erstelle ein Objekt der Klasse `ElectricCar` und rufe die Methode `startEngine` auf.
-

Übung 6: Vererbung mit Konstruktoren

Erstelle eine Klasse `Shape` mit dem Attribut `type`. Dann erstelle eine abgeleitete Klasse `Circle`, die das Attribut `radius` hinzufügt und die Methode `getArea` implementiert.

1. Erstelle ein Objekt der Klasse `Circle` mit `type = "circle"` und `radius = 5`.
 2. Berechne und gib die Fläche des Kreises aus.
-

Übung 7: Abgeleitete Klasse `Dog` von `Animal`

Erstelle eine Klasse `Animal` mit den Eigenschaften `name` und `age`. Dann erstelle eine Klasse `Dog`, die von `Animal` erbt und eine Methode `bark` hat.

1. Erstelle ein Objekt der Klasse `Dog` mit `name = "Buddy"` und `age = 3`.
 2. Rufe die Methode `bark` auf und gib das Alter des Hundes in der Konsole aus.
-

Übung 8: Vererbung mit privater Methode

Erstelle eine Klasse `Gadget` mit einer privaten Methode `#updateSoftware`. Dann erstelle eine abgeleitete Klasse `Smartphone`, die die Methode `updateSoftware` aufruft.

1. Erstelle ein Objekt der Klasse `Smartphone` und rufe die Methode `updateSoftware` über eine öffentliche Methode auf.
-

Übung 9: Überschattung in Fahrzeugklassen

Erstelle eine Klasse `Vehicle` mit einer Methode `getFuelType`. Dann erstelle eine abgeleitete Klasse `Truck`, die diese Methode überschattet und `super.getFuelType()` aufruft.

1. Erstelle ein Objekt der Klasse `Truck` und rufe die Methode `getFuelType` auf.
-

Übung 10: Abgeleitete Klasse mit zusätzlichen Methoden

Erstelle eine Klasse `Device` mit der Methode `powerOn`. Dann erstelle eine abgeleitete Klasse `Laptop`, die zusätzlich die Methode `sleep` hat.

1. Erstelle ein Objekt der Klasse `Laptop` und rufe beide Methoden auf.
-