

# V8 Fahrzeug

---

## Abstrakte Klassen

Abstrakte Klassen sind Klassen, die **nicht direkt instanziiert** werden können und mindestens eine **abstrakte Methode** enthalten. Diese dienen als Vorlage für Unterklassen, die die **abstrakten Methoden implementieren müssen**.

## Warum braucht man abstrakte Klassen

### 1. Symbolische Konzepte:

Oftmals repräsentieren Superklassen abstrakte Konzepte, die nicht direkt existieren. Zum Beispiel gibt es in der Realität Autos, Busse und LKWs, aber kein allgemeines "Fahrzeug". Es macht also keinen Sinn, ein **Fahrzeug-Objekt** zu erstellen, da dieses Konzept nur als Oberbegriff für spezifischere Klassen dient. In solchen Fällen ist es sinnvoll, eine **abstrakte Klasse** zu verwenden, die **Eigenschaften** und **Methoden** an die Subklassen weitervererbt, aber selbst keine Objekte erzeugen kann. Nur Subklassen wie **Auto** oder **Bus** dürfen Instanzen haben.

### 2. Verschiedene Implementierungen in Subklassen:

Manchmal haben Subklassen stark unterschiedliche Implementierungen einer gemeinsamen Methode. Zum Beispiel könnte eine **fahren()**-Methode in einer Auto-Klasse etwas anderes ausgeben als in einer Bus- oder LKW-Klasse:

- Auto: "Auto fährt."
- Bus: "Bus fährt."
- LKW: "LKW fährt."

In solchen Fällen wäre es redundant und fehleranfällig, eine allgemeine **fahren()** Methode in der Superklasse **Fahrzeug** zu implementieren. Stattdessen kann man in der **abstrakten Klasse** eine Methode ohne Implementierung definieren, die dann in den Subklassen individuell überschrieben wird. Dies spart nicht nur unnötigen Code, sondern zwingt auch jede Subklasse, die Methode korrekt zu implementieren.

## Beispiel für eine abstrakte Klasse in Dart

```
// Abstrakte Fahrzeugklasse
abstract class Fahrzeug {

    // Abstrakte Methode
    void fahren(); // Keine Implementierung in der Superklasse
}

// Subklasse Auto
class Auto extends Fahrzeug {

    @override
    void fahren() {
```

```
        print('Auto fährt.');
```

```
    }  
}
```

```
// Subklasse Bus  
class Bus extends Fahrzeug {  
  
    @Override  
    void fahren() {  
        print('Bus fährt.');    }  
}
```

```
void main() {  
    Auto auto = Auto();  
    Bus bus = Bus();  
    auto.fahren(); // Ausgabe: Auto fährt.  
    bus.fahren();  // Ausgabe: Bus fährt.  
}
```

---

## Aufgabe

Suchen Sie im realen Leben nach konzeptionellen oder allgemeinen Dingen, die als abstrakte Klasse repräsentiert werden können, und erstellen Sie eine abstrakte Klasse dafür. In dieser Klasse definieren Sie sowohl normale als auch abstrakte Methoden. Erstellen Sie für diese abstrakte Klasse ein paar Subklassen durch Extends.