# Lektion 14 – Einführung in die Objektorientierte Programmierung

## Nachschlagen bei…

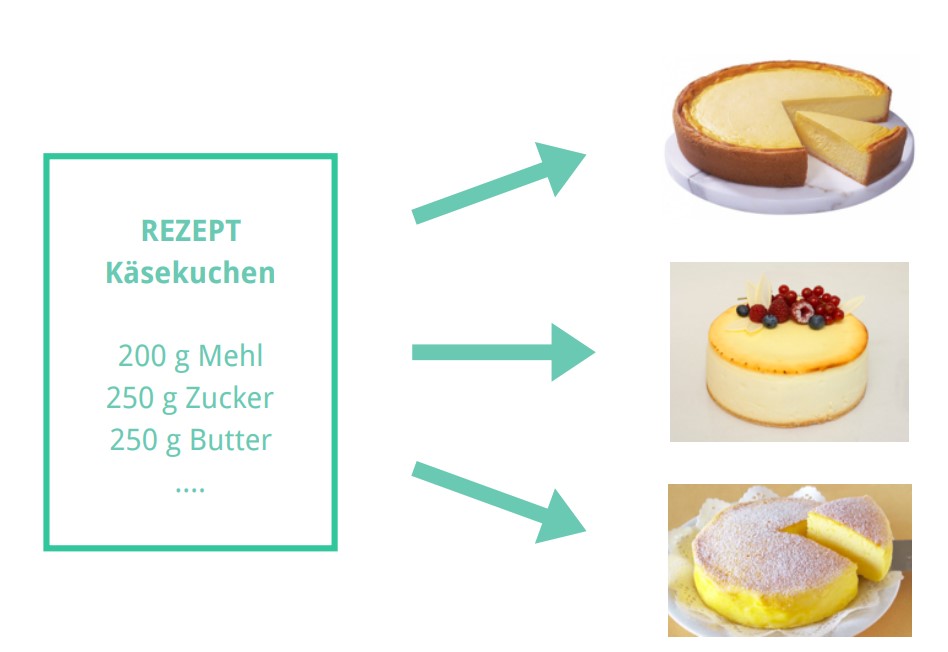
W3Schools (Kapitel Classes/Objects) oder <https://entwickler.de/online/development/einfuehrung-programmierung-objektorientierte-programmentwicklung-197372.html>

## Anweisungen

Studiere die Erläuterungen in diesem Dokument bzw in den Nachschlagewerken. Beantworte die Fragen und kopier den Code der Aufgaben hierher.

## Erläuterungen

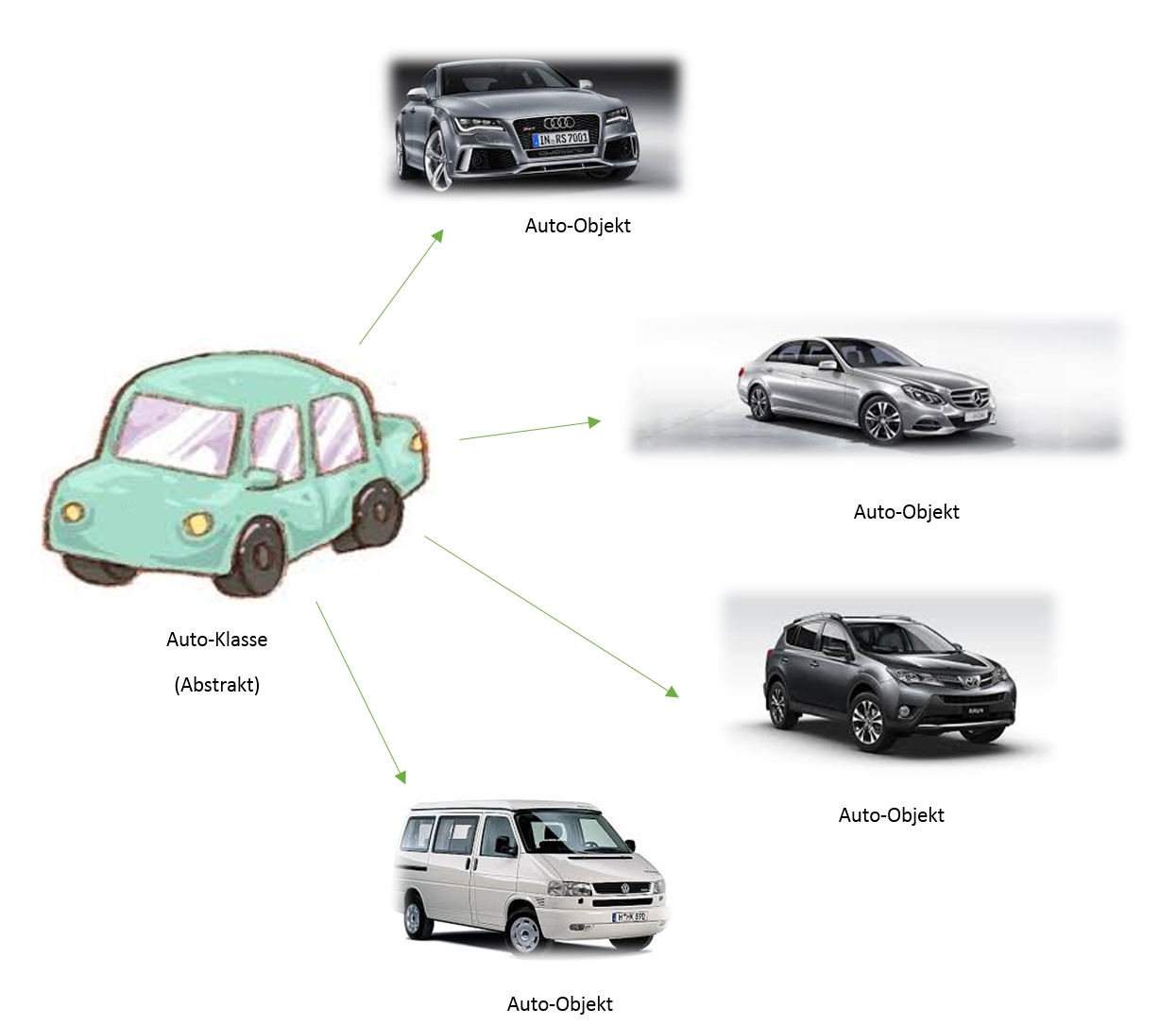
**Wozu OOP?** Bei komplexen Programmen verliert man schnell den Überblick und programmiert ineffizient. Daher wurde die objektorientierte Programmierung erfunden. Die wesentliche Neuerung ist, dass Objekte verschiedene Eigenschaften (das sind dann die Variablen der Objekte) und Methoden (das sind dann die Funktionen der Objekte) besitzen können, die in einer einzigen Variablen (nämlich dem Objekt) gekapselt sind.



**Klasse:** Es gibt einen abstrakten Bauplan für die Objekte. Dieser wird als Klasse bezeichnet. Er enthält die **Eigenschafen** (Mehlmenge, Belagsart etc.), die aber noch keinen Wert besitzen.

**Instanz:** Aus der Klasse können beliebig viele Objekte (=Instanzen) von Käsekuchen erzeugt werden. Die Eigenschaften werden mit unterschiedlichen Werten belegt, daher sind die Objekte dann auch unterschiedlich, aber nicht mehr abstrakt, sondern konkret.

### Objekt Auto



Die abstrakte Klasse „Auto“ beschreibt die Eigenschaften und Methoden eines Autos

**Eigenschaften**: Hersteller, Typ, Farbe, Baujahr, Geschwindigkeit….

**Methoden:** beschleunigen, bremsen, lackieren,….

Erst wenn diese Klasse **instanziert** wird, haben wir ein Objekt, das wir in einem Computerprogramm benützen können.

## Beispiel in Python

# Klassendefinition

class auto(object):

# \_\_init\_\_ ist eine spezielle Funktion und wird Konstruktor genannt. Sie wird automatisch aufgerufen, wenn ein Objekt erzeugt wird

def \_\_init\_\_(self,hersteller, typ, farbe, baujahr):

self.hersteller=hersteller

self.typ=typ

self.farbe=farbe

self.baujahr=baujahr

self.geschwindigkeit=0

def beschreibung(self):

print("ich bin ein",self.hersteller,self.typ,"aus dem Jahr",self.baujahr,"und habe die Farbe",self.farbe)

# der erste Parameter einer Funktion ist immer self, weil er auf das Objekt selbst verweist

def beschleunigen(self,x):

self.geschwindigkeit+=x

def geschwindigkeitsanzeige(self):

print("Ich fahre mit",self.geschwindigkeit,"km/h")

# zuerst wird ein Objekt instanziert

auto1 = auto("Skoda","Oktavia","blau",2017)

auto1.beschreibung()

auto1.beschleunigen(30)

auto1.geschwindigkeitsanzeige()

# es können beliebig viele weitere Objekte instanziert werden

auto2 = auto("Toyota","Avensis","grün",2013)

auto2.beschreibung()

## Fragen

Entwerfe eine Klasse „Tier“. Welche Eigenschaften und Methoden könnte diese Klasse haben? Welche Objekte können daraus erzeugt werden?

Welcher Unterschied besteht zwischen Klasse und Instanz? Erläutere die beiden Begriffe.

## Aufgaben

1. Kopier das Beispielprogramm in die Datei autoklasse.py. Ergänze die Klasse „auto“ durch die Methoden „bremsen“ und „lackieren“. Letztere ändert die Farbe. Demonstriere die Wirkung durch Aufruf der entsprechenden Methoden.
2. Erzeuge eine Datei roboter.py. Definiere in dieser Datei eine Klasse „roboter“. Sie soll die Eigenschaften „name“, „farbe“ und „baujahr“ besitzen. Sie soll eine Methode sayhello() besitzen, die den Satz „Hallo ich bin Robbie und habe die Farbe blau“ ausgibt (natürlich nur wenn die Eigenschaften wirklich mit diesen Werten belegt wurden – man kann auch andere nehmen). Weiters soll sie die Methode lackieren(farbe) besitzen, die die Farbe ändert. Demonstriere die Änderungen durch Aufruf der Methoden. Natürlich soll die Klasse einen Konstruktor besitzen, über den die Eigenschaften bei der Instanzierung übergeben werden können.