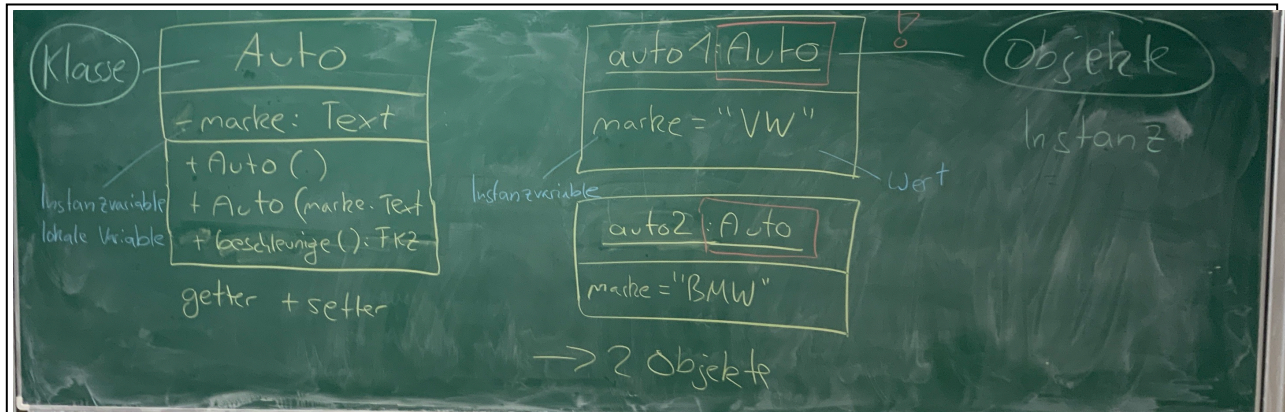


### Klassenvariablen und Klassenmethoden

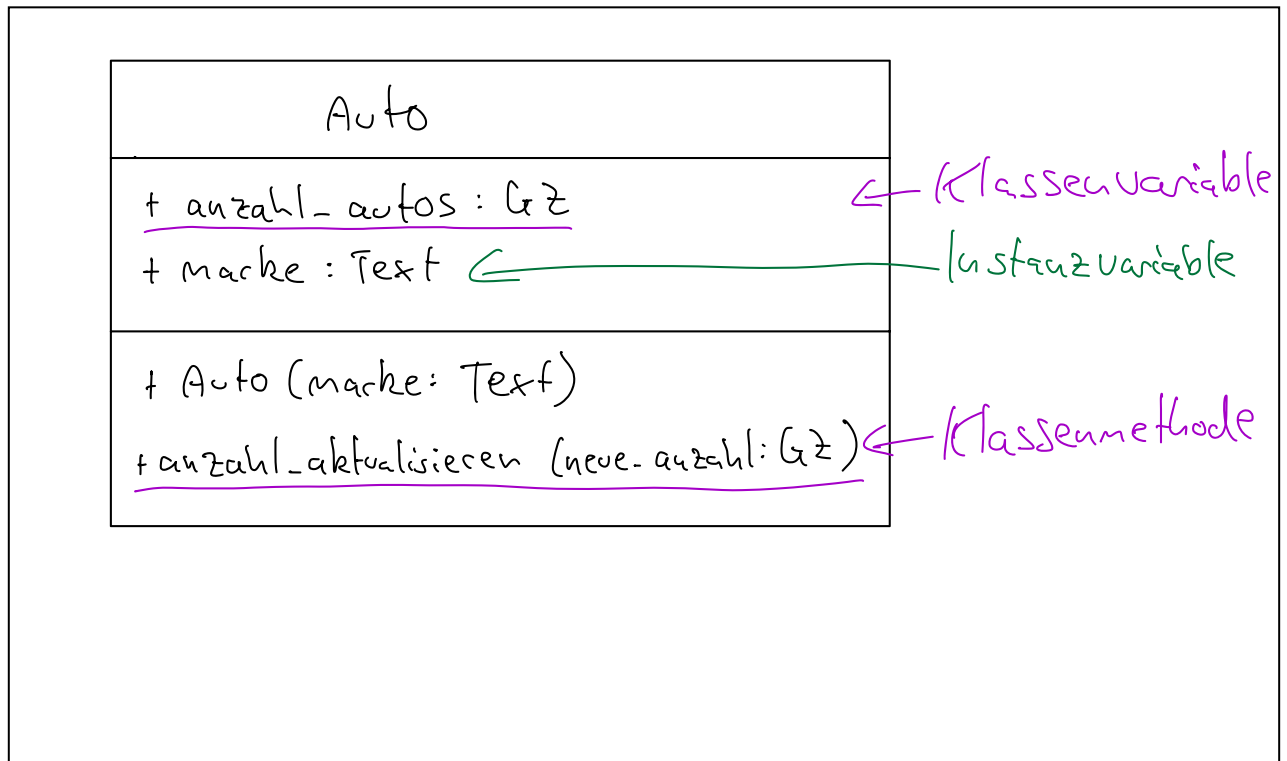
Beispiel: Auto



- Wird in einem Programm ein Objekt geschaffen, dann operieren alle Methoden auf einem eigenen Satz von Daten. Ändert ein Objekt seinen Datenbestand, hat das keine Auswirkung auf die anderen Objekte der gleichen Klasse.
- Manchmal gibt es Situationen für Eigenschaften, die nicht direkt einem individuellen Objekt zugeordnet werden können. Diese Eigenschaften sind Klassen zugeordnet und werden durch **statische Variablen / Methoden** verwirklicht.  
→ **Klassenvariable / Klassenmethode**
- Klassenvariablen sind Variablen, die für alle Instanzen einer Klasse gemeinsam sind. Sie werden **innerhalb der Klasse**, aber **außerhalb jeder Methode** definiert. Sie sind nützlich, wenn man Daten speichern möchte, die für alle Instanzen (= Objekte) gleich sind.
- Klassenmethoden sind Methoden, welche auf die Klasse selbst und nicht auf eine Instanz der Klasse angewendet werden.



Beispiel: Auto mit Klassenvariablen und Klassenmethoden in UML



in Python

```
class Auto():
    anzahl_autos = 0

    def __init__(self, marke):
        self.marke = marke
        Auto.anzahl_autos += 1

    @classmethod
    def anzahl_aktualisieren(cls, neue_anzahl):
        cls.anzahl_autos = neue_anzahl

auto1 = Auto("Toyota")
auto2 = Auto("Honda")
print(Auto.anzahl_autos) # Ausgabe: 2

# Aufruf der Klassenmethode
Auto.anzahl_aktualisieren(5)
print(Auto.anzahl_autos) # Ausgabe: 5
```



## Übungsaufgaben

### Aufgabe 1: Mitarbeiterverwaltung

Es soll eine Klasse in UML skizziert werden, die zur Verwaltung von Mitarbeitern dient. Die Klasse hat folgende Eigenschaften:

- Jeder Mitarbeiter besitzt eine Nummer, einen Vor- und Nachnamen, ein Gehalt und hat eine Abteilung.
- Die Klasse hat einen Zähler integriert, mit dessen Hilfe eine fortlaufende Mitarbeiternummer erzeugt wird.
- Mit Hilfe der statischen Funktion `getNaechsteFreieMitarbeiterNummer()` wird die nächste freie Mitarbeiternummer ermittelt. Der Zähler `mitarbeiterZaehler` wird dabei um eins hochgezählt, damit die Mitarbeiter-Nummern eindeutig bleiben.
- Die Generierung der Mitarbeiternummer soll im Konstruktor vorgenommen werden.
- Beim Erzeugen eines Mitarbeiter-Objektes kann die Mitarbeiternummer mitgegeben werden. Es ist aber auch möglich einen Mitarbeiter ohne eine Nummer zu erzeugen.
- Mit Hilfe einer Funktion `mitarbeiterAusgeben()` sollen alle Informationen zum Mitarbeiter für die Personalverwaltung ausgegeben werden können.



Erstellen Sie das UML Klassendiagramm für diese Klasse. Alle Attribute sollen *private* deklariert sein. Vervollständigen Sie das Klassendiagramm mit allen *getter* und *setter*-Funktionen.

diagrams-net

### Aufgabe 2: Buchverwaltung

In einer Bibliothek gibt es viele Bücher. Jedes Buch hat einen Titel, einen Autor und eine ISBN-Nummer. Die Bibliothek hat eine Klassenvariable `gesamtanzahl_buecher`, die die Gesamtzahl der Bücher in der Bibliothek speichert. Jedes Mal, wenn ein neues Buch zur Bibliothek hinzugefügt wird, wird diese Variable aktualisiert. Es gibt eine Klassenmethode `buch_hinzufuegen`, die ein neues Buch zur Bibliothek hinzufügt und die `gesamtanzahl_buecher` entsprechend erhöht. Beim Erzeugen eines Buch-Objektes sollen alle Instanzvariablen als Parameter übergeben werden.

Erstellen Sie das UML Klassendiagramm für diese Klasse. Alle Attribute sollen *private* deklariert sein. Vervollständigen Sie das Klassendiagramm mit allen *getter* und *setter*-Funktionen.