

Grundlagen der Programmierung

Vorlesung und Übung

06 – Klassen und Objekte

Prof. Dr. Andreas Biesdorf

Wirtschaft
Hauptcampus

H O C H
S C H U L E
T R I E R

Themen bisher:

- Grundlegende Operatoren und Berechnungen
- Kleinere Programme mit einfachen Kontrollstrukturen
- Implementierung einfacherer Suchen
- Bedingte Anweisungen
- Verwendung von Booleschen Operatoren und Vergleichsoperatoren
- Wiederholung von Code-Abschnitten mittels Schleifen
- Funktionen

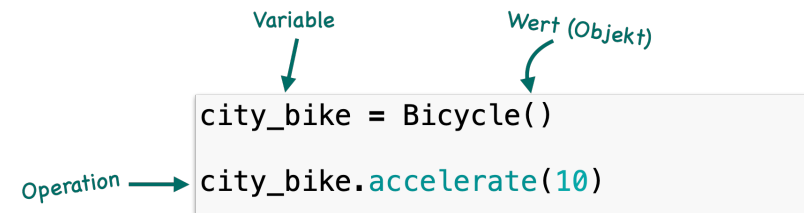
Themen:

- Grundkonzepte der objektorientierten Programmierung (OOP)
- Grundlagen von Methoden, Klassen und Objekten
- Initialisieren von Objekten über Konstruktoren
- Wichtige Begriffe wie Kapselung und Geheimnisprinzip

Themen:

- Eigene Klassen definieren und Objekte anlegen
- Zustände von Objekten verändern
- Methoden mit Parametern und Rückgabewerten deklarieren und aufrufen
- Sichtbarkeit von Datenfeldern und Methoden einschränken
- Wichtige Grundkonzepte der objektorientierten Programmierung wie Kapselung und Geheimnisprinzip erläutern

- Objektorientierte Programmierung ermöglicht es, eigene Datentypen zu definieren und anzulegen
- Können dann in Programmen verwendet werden, um Werte zu speichern und (komplexere) Operationen auszuführen



- Programmcode wird strukturiert (*Modularität*)
- Programmcode kann einfacher wiederverwendet und mit anderen geteilt werden (*Wiederverwendbarkeit*)
- Programmcode wird leichter verstanden (*Wartbarkeit*)
- Programmcode kann leichter um neue Funktionalitäten ergänzt werden (*Erweiterbarkeit*)
- Programmcode kann verteilt entwickelt werden (*Flexibilität*)

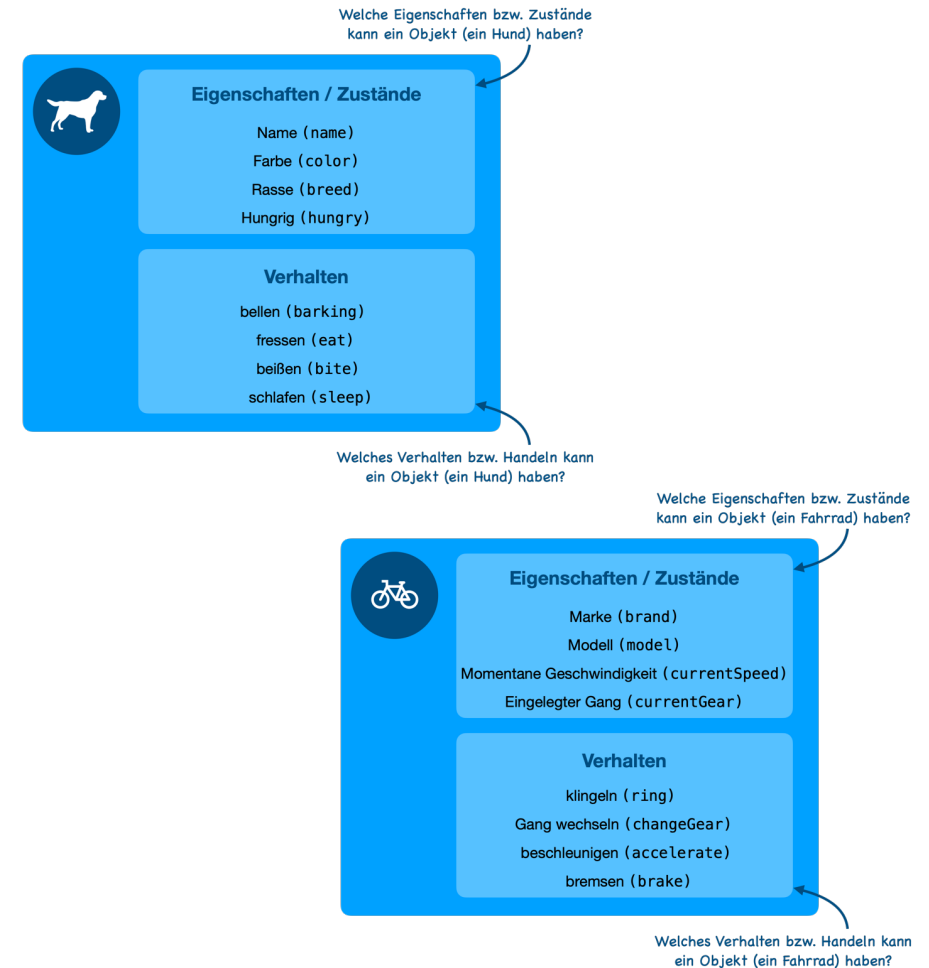
Objekte

Wirtschaft
Hauptcampus

H O C H
S C H U L E
T R I E R

Was ist ein Objekt?

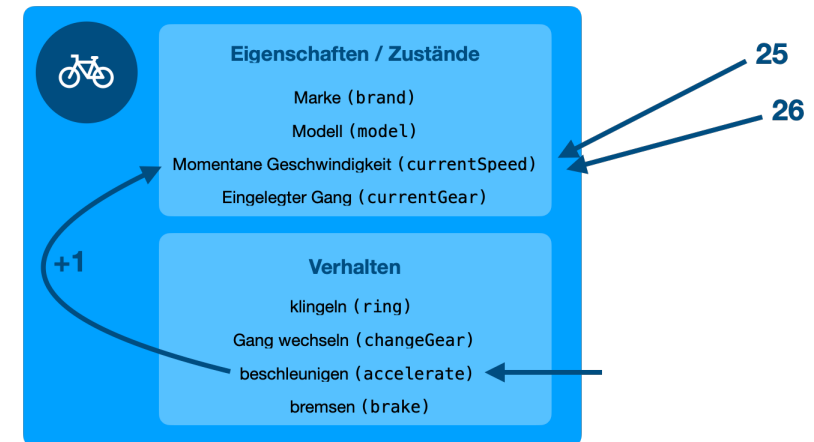
- Objekte repräsentieren Dinge der realen Welt oder eines Problembereichs
 - Beschreiben *Zustand & Verhalten*
- **Beispiele:**
 - „Das rote Auto auf dem Parkplatz“
 - „Der bellende Hund“



Was ist ein Objekt?

Jedes Objekt aus der realen Welt lässt sich in einem Software-Objekt abbilden

- Objekte in Programmen speichern ihre Zustände in *Variablen* (Datenfelder)
- Objekte in Programmen legen ihr Verhalten über *Methoden* offen bzw. lassen sich über Methoden *steuern*
- Methoden können den Zustand *verändern*



Klassen

Wirtschaft
Hauptcampus

H O C H
S C H U L E
T R I E R

Was ist eine Klasse?

In der realen Welt finden sich viele Objekte der gleichen Art

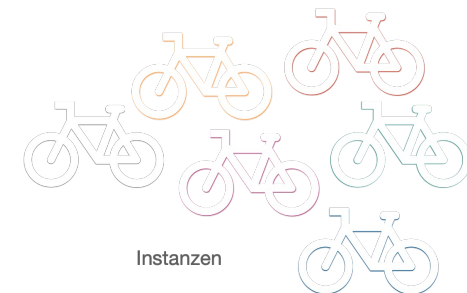
- Beispielsweise viele Fahrräder, von der gleichen Marke und dem gleichen Modell
- Gebaut nach den gleichen Bauplänen und mit den gleichen Komponenten

In der Objektorientierung ist eine Klasse der **Bauplan**, aus dem einzelne Objekte erstellt werden können

Die Objekte werden als "Instanzen" einer Klasse bezeichnet



Klasse



Instanzen

Woraus besteht eine Klasse?

Beschreiben Eigenschaften
bzw. Zustände eines Objekts
(Datenfeld)

Initialisieren ein Objekt,
übernehmen ggf. Anfangswerte

Beschreiben das Verhalten bzw.
Handlungen eines Objekts

Klasse

Instanzvariablen

Konstruktor(en)

Methoden

```
class Rechteck:
```

```
    _kante_a = 0  
    _kante_b = 0  
  
    _name = ""
```

```
    def __init__(self, k_a, k_b):  
        self._kante_a = k_a  
        self._kante_b = k_b
```

```
    def set_kanten(self, k_a, k_b):  
        self._kante_a = k_a  
        self._kante_b = k_b
```

```
    def print_werte(self):  
        print(str(self._kante_a), str(self._kante_b))
```

```
    def update_self_name(self):  
        self._name = str(self._kante_a) + "x" + str(self._kante_b)
```

```
    def __str__(self):  
        self.update_self_name()  
        return self._name
```

Wie erzeuge ich nun ein neues Objekt?

- Klassen sind Vorlagen, aus denen Objekte erzeugt werden können
- Von einer Klasse können mehrere Instanzen [Objekte] erzeugt werden
- Zugriff auf Datenfelder und Methoden via „Dot“-Operator [.]

```
quadrat = Rechteck(15,15)  
quadrat.print_werte()  
print(str(re))
```

```
re = Rechteck(15,20)  
re.set_kanten(10,20)  
re.print_werte()  
print(str(re))
```

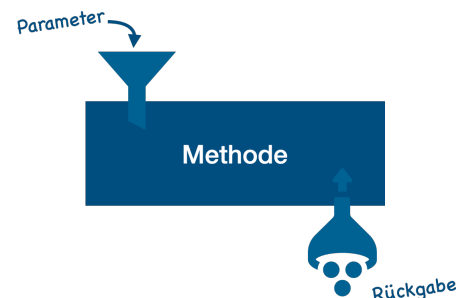
Funktionen und Parameter

Wirtschaft
Hauptcampus

H O C H
S C H U L E
T R I E R

Einführung

- Über Funktionen kann mit Objekten interagiert werden
- Funktionen können Parameter besitzen, über die zusätzliche Informationen übergeben werden können
- Funktionen können via "return" das Ergebnis einer Operation zurückgeben



```
def set_kanten(self, k_a, k_b):  
    self._kante_a = k_a  
    self._kante_b = k_b  
  
def print_werte(self):  
    print(str(self._kante_a), str(self._kante_b))  
  
def update_self_name(self):  
    self._name = str(self._kante_a) + "x" + str(self._kante_b)  
  
def __str__(self):  
    self.update_self_name()  
    return self._name
```

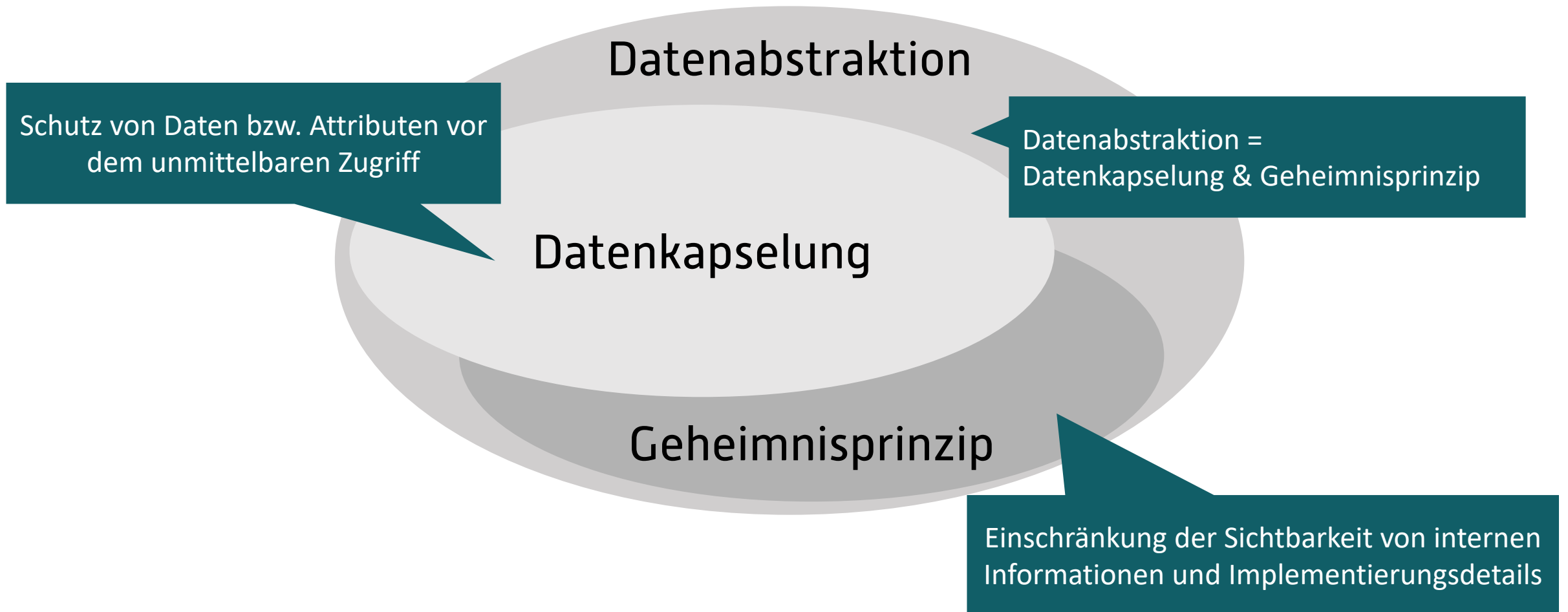
Einführung

- Methoden werden nicht nur zur Objektinteraktion verwendet
- Methoden **erhöhen die Wiederverwendbarkeit** von Programmcode
 - Wiederkehrende Programmteile müssen nicht immer wieder neu programmiert werden, sondern können an zentraler Stelle aufgerufen werden
- Methoden **reduzieren die Komplexität** von Programmen
 - Komplexe Programme können mit Hilfe von Methoden in kleine Teilprogramme zerlegt werden

Datenkapselung und Geheimnisprinzip

Wirtschaft
Hauptcampus

H O C H
S C H U L E
T R I E R



Namen	Bezeichnung	Bedeutung
name	Public	Attribute ohne führende Unterstriche sind sowohl innerhalb einer Klasse als auch von außen les- und schreibbar.
_name	Protected	Man kann zwar auch von außen lesend und schreibend zugreifen, aber der Entwickler macht damit klar, dass man diese Member nicht benutzen sollte. Protected-Attribute sind insbesondere bei Vererbungen von Bedeutung.
__name	Private	Sind von außen nicht sichtbar und nicht benutzbar.

Zusammenfassung

Wirtschaft
Hauptcampus

H O C H
S C H U L E
T R I E R

Einführung

- Klassen modellieren Konzepte der realen Welt
 - Sind der Bauplan für Objekte
 - Aus einer Klasse können mehrere Objekte erzeugt werden (= Instanzen einer Klasse)
- Instanzen können verschiedene Zustände annehmen
 - Zustände werden über Datenfelder (Instanzvariablen) verwaltet
- Methoden implementieren das Verhalten von Objekten

Einführung

- Methoden können via "return" ein Ergebnis (einen Wert) zurückgeben oder nicht
- Über Access Modifier kann der Zugriff auf Datenfelder/Methoden eingeschränkt werden
 - Standardmäßig sollten Datenfelder immer "private" deklariert werden und Werte nur über getter- und setter-Methoden verändert werden
- Konstruktoren können verwendet werden, um Objekte in bestimmten Zuständen zu initialisieren
- Parameter dienen dazu, um einem Konstruktor oder einer Methode Werte zu übergeben

Kontrollfragen

- Was ist der Unterschied zwischen einer Klasse und einem Objekt?
- Wie wird eine neue Instanz einer Klasse (Objekt) angelegt?
- Was ist ein Parameter?
- Was versteht man unter der "Signatur" einer Methode?
- Was ist ein Konstruktor?