

# Objektorientierte Programmierung

Anna-Lena Barner,  
Patrick Jende



# *Stundenfahrplan:*

- Objektorientierung
- UML-Klassendiagramme
- Klassen in Python

# Unser Beispiel-Mii

---





Beenden

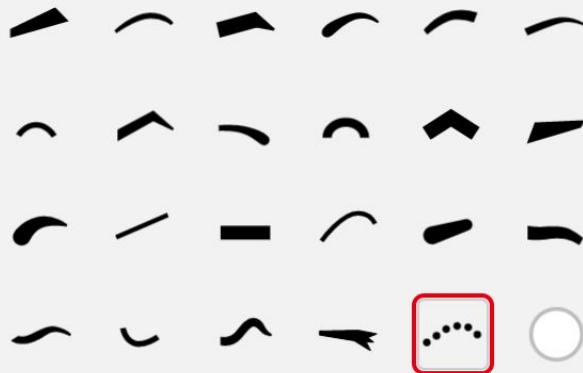
Fertig

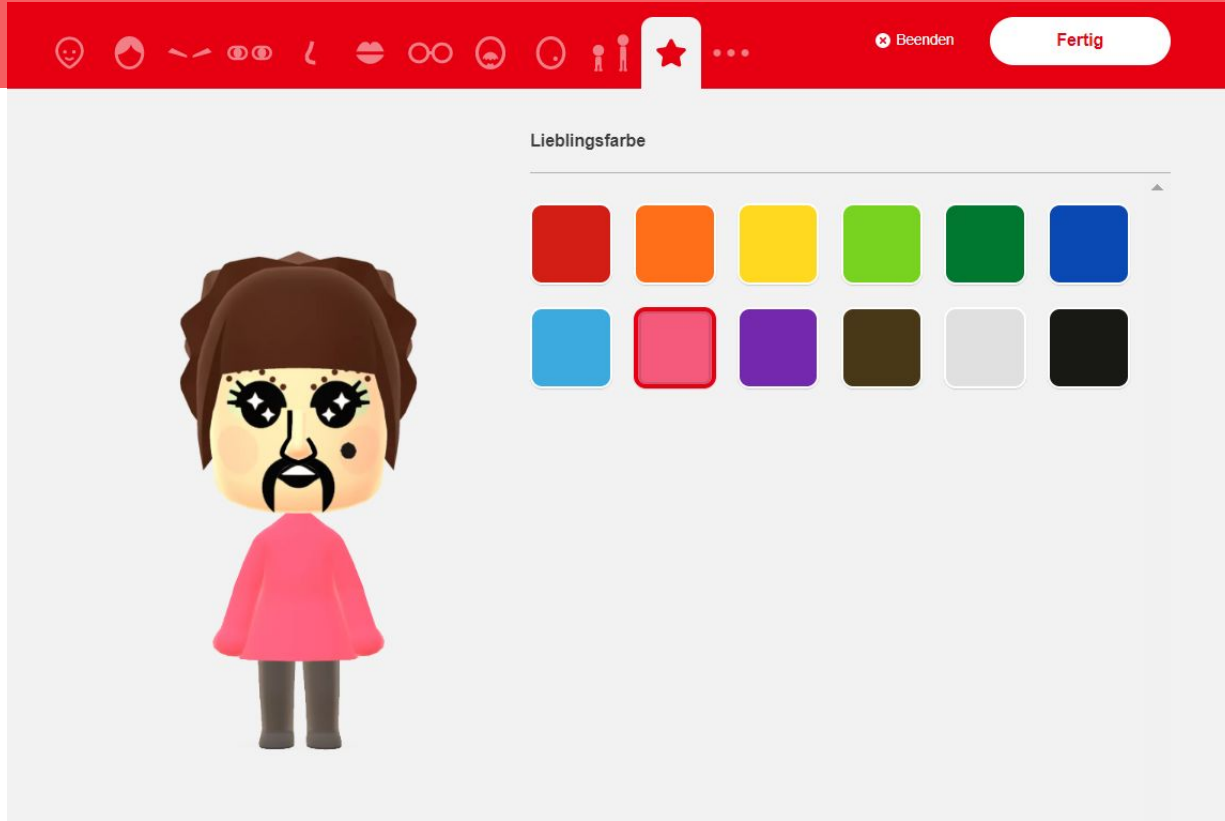
Augenbrauen

Elemente

Farbe

Position





# Viele "Objekte" - eine Klasse



# Aufgabe: 1. Erstelle für folgende Klasse ein UML-Diagramm

Die Klasse soll "Bruch" heißen. Überlege dir, welche Attribute ein Bruch hat und welche Methoden geeignet wären.

2. Überlege dir nun ein eigenes Beispiel aus deinem Alltag und konstruiere ein UML-Diagramm mit mindestens 4 Attributen und 4 Methoden.

Einzelarbeit, 10 min

# Klassen in Python



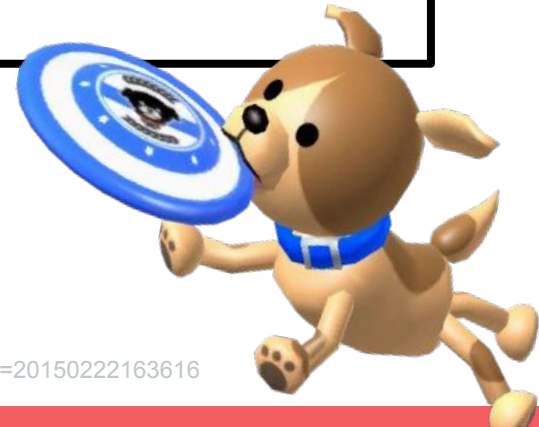
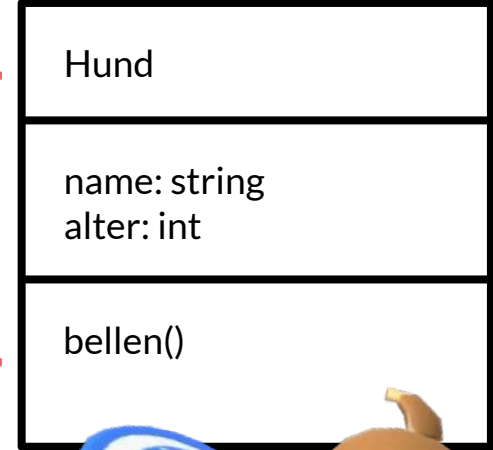


<https://i.ytimg.com/vi/a37cAqxKoBY/maxresdefault.jpg>

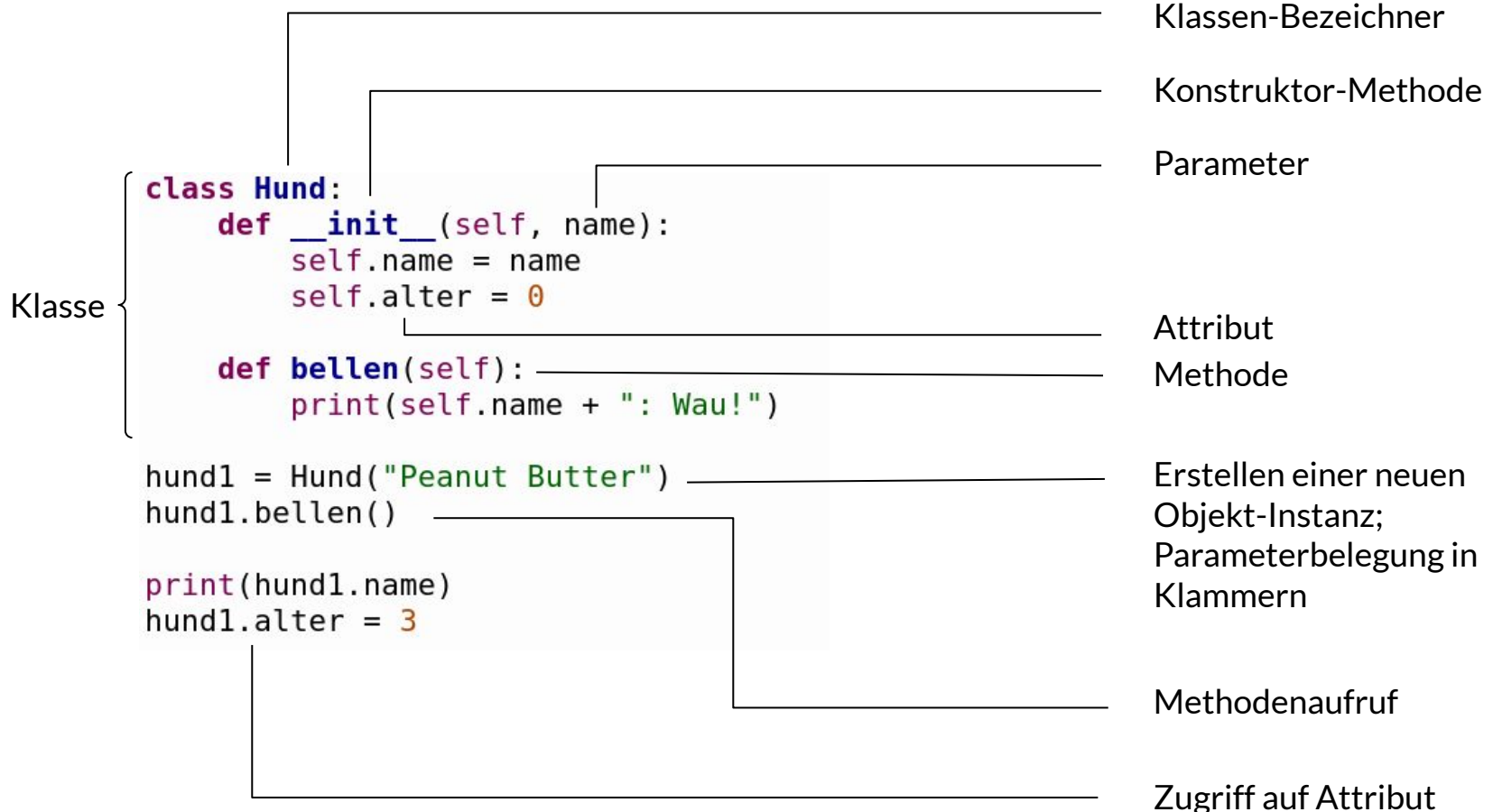
# Python und UML

```
class Hund:
    def __init__(self, name):
        self.name = name
        self.alter = 0

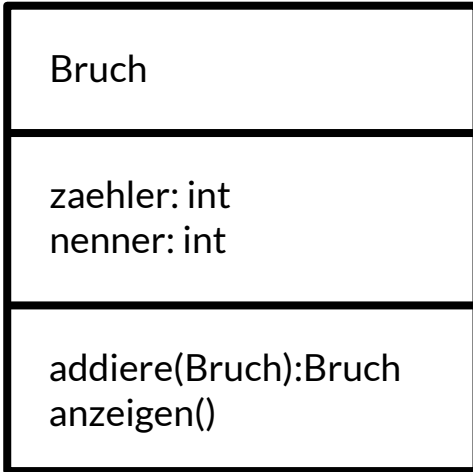
    def bellen(self):
        print(self.name + ": Wau!")
```



# Talk Nerdy to Me



# Rechnen mit Brüchen - Die Klasse Bruch



Erstelle die Klasse Bruch entsprechend des Klassendiagramms.

1. Ergänze den Konstruktor mit den Attributen.
2. Ergänze die Methode **anzeigen()**
3. Implementiere die Methode **addiere()**

Einzelarbeit, 20-25 min

Link: [patjen.de/aufgabe01](https://patjen.de/aufgabe01)

Erkunden

# Klassen und Objekte in Apps

Untersuche ein paar Apps auf deinem Smartphone/Laptop hinsichtlich Objekten, die in dessen Benutzeroberflächen auftreten.

**Wie könnten entsprechende Objekt-Klassen aussehen?**

**Modelliere drei solcher Objektklassen in UML-Notation. Gib zu jeder Klasse 3 Attribute und 3 Methoden und jeweils ein Beispiel-Objekt an.**

Bis zum nächsten Mal, am 6. Dezember.

Link: [patjen.de/folien01](https://patjen.de/folien01)

Bye.

Anna-Lena Barner,  
Patrick Jende

