5 Objektorientierte Programmierung

Projekttage 17 bis 19 Juli 2024

Problemstellung

Beim Darstellen von großen und/oder komplexen Dingen
(z.B. Eine Kreatur in einem Videospiel)

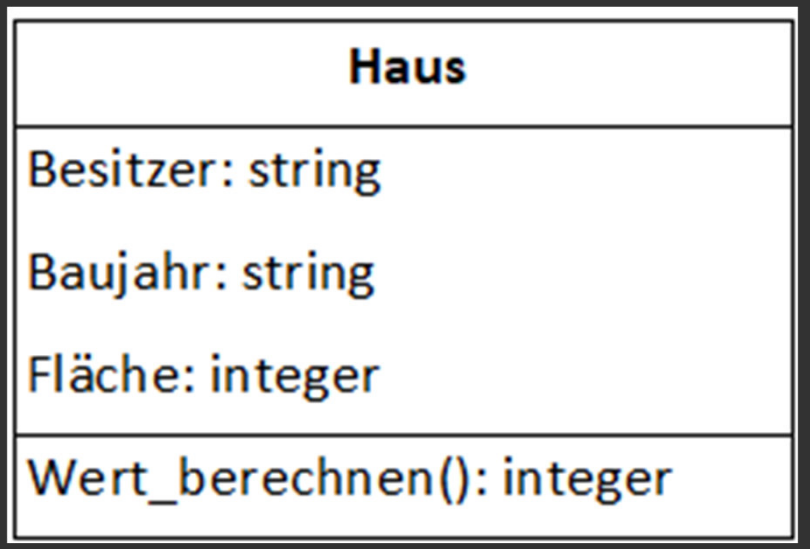
- Viele einzelne Variablen
- Wird schnell unübersichtlich

```
1   Peters_Haus_Besitzer = "Peter Mustermann"
2   Peters_Haus_Baujahr = "1985"
3   Peters_Haus_Flaeche = 50
4   Peters_Haus_Anzahl_Zimmer = 3
5
6   Annas_Haus_Besitzer = "Anna Becker"
7   Annas_Haus_Baujahr = "2005"
8   Annas_Haus_Flaeche = 70
9   Annas_Haus_Anzahl_Zimmer = 4
10
11  def Wert_berechenen(besitzer):
12      if besitzer == Peters_Haus_Besitzer:
13          print("Besitzer ist Peter")
14          # calculate
15      elif besitzer == Annas_Haus_Besitzer:
16          print("Besitzer ist Anna")
17          # calculate
```

Lösung

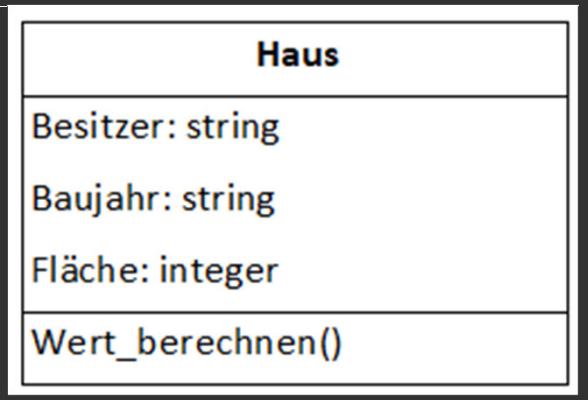
Objekt schreiben, dass alle einzelnen Werte zusammenhält

- übersichtlicher, kompakter
- Einfach zu bedienen



Wichtige Bezeichnungen

- Klasse: Eine ‚Schablone' zum Erstellen eines Objektes. Alle Objekte einer Klasse wurden nach derselben ‚Schablone' erstellt.
  Bspw. zeigt das Bild den Aufbau der Klasse Haus



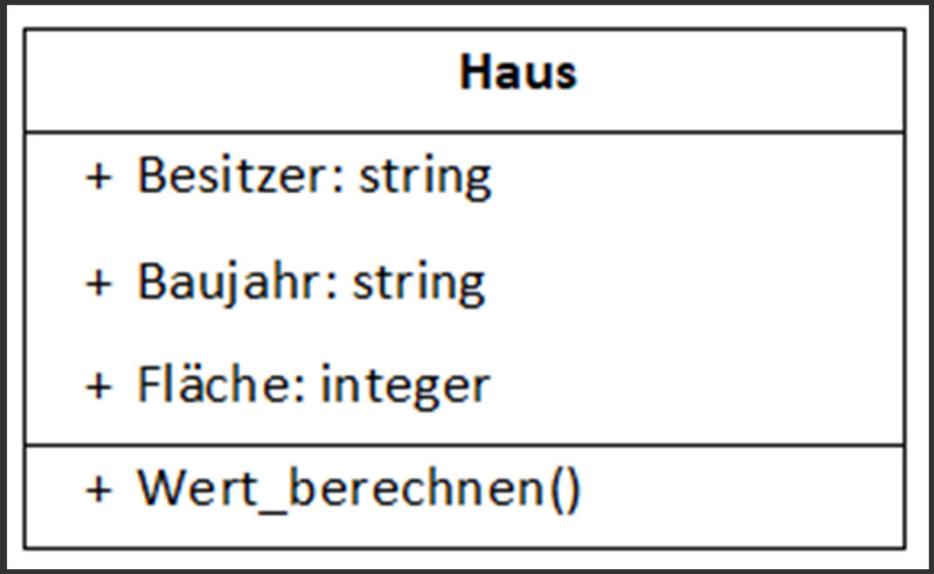
- Instanz: eine Instanz ist das konkrete mit Werten initialisierte ‚Packet', das nach Vorlage einer Klasse erstellt wurde.
  Bspw. ist wäre Peters, oder Annas Haus eine Instanz der Klasse Haus
- Objekte: Objekte bezeichnen im allgemeinen Instanzen (irgend-)einer Klasse. Statt einer **Variable** hat man dann ein **Objekt** im Speicher.

Objekte

Objekte dienen dazu, Werte gesammelt zu speichern und zu verarbeiten.

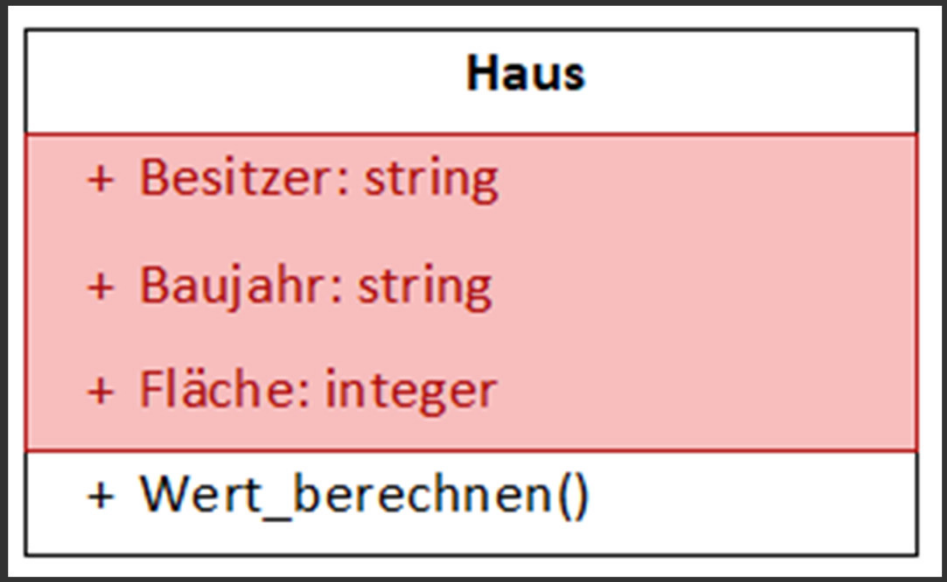
Objekte bestehen grundsätzlich aus zwei Dingen:

- Attributen
- Methoden

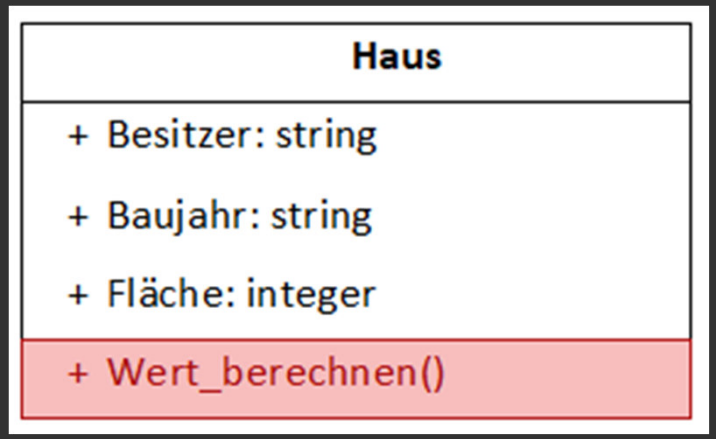


Attribute

- Attribute sind ‚Variablen' in einem Objekt.
- Sie sind wichtig, da nur so Werte innerhalb eines Objektes gespeichert werden können.
- Eine Instanz einer Klasse hat immer die gleichen Attribute



Methoden



- Methoden sind ähnlich wie Funktionen. Sie machen festgelegte Dinge mit ihren Parametern und geben (manchmal) Werte zurück.
- Der Unterschied ist: eine Methode operiert immer auf einer Instanz, die sie als Parameter nehmen muss bzw. automatisch nimmt.
- Aufruf in den meisten Programmiersprachen: Test_Objekt.Methode() (Auch ein Unterschied zur ‚normalen' Funktion)
- Können, ähnlich wie Attribute öffentlich (public) oder privat (private) sein

Operatorüberladung

Um Instanzen einer Klasse vergleichen und addieren zu können, muss man die Vergleichs- und Rechenoperatoren überladen.

`Ergebnis = Wert1 + Wert2` ist identisch mit `Ergebnis = Wert1.__add__(Wert2)`

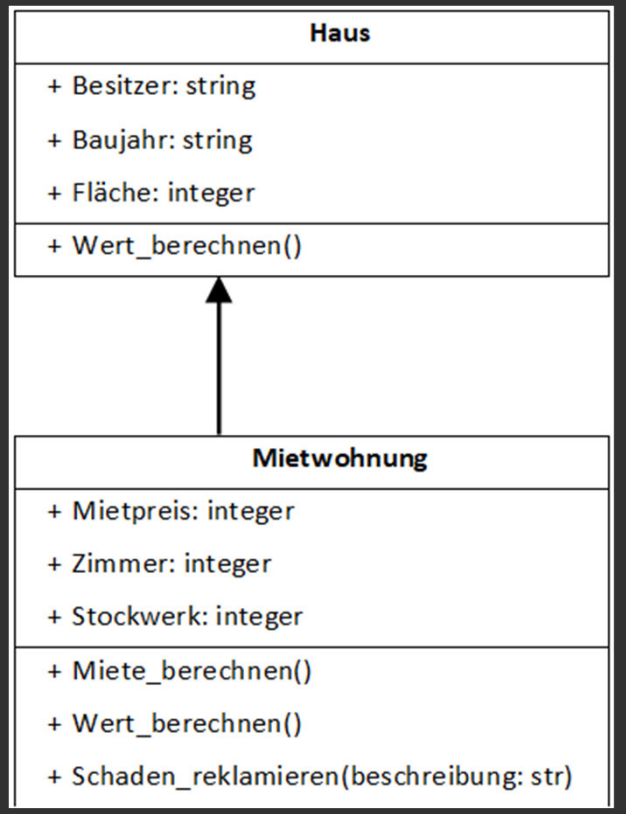
Hat eine Klasse die Methode `def __add__(self, other: 'Objekt')` implementiert, so gelingt auch der Aufruf von `Ergebnis = Test_Objekt1 + Test_Objekt2`

Instanzen der Klasse lassen sich so addieren. Wie genau wird in der Methode `def __add__(self, other: 'Objekt')` definiert.

Das funktioniert mit anderen Operatoren genauso, nur sind die Name der Methoden, die die Klasse haben muss andere.

Vererbung

Eine Klasse kann von einer anderen erben und übernimmt dabei alle öffentlichen Attribute und Methoden. In abgeleiteten Klasse (die, die erbt), kann man weitere Attribute und Methoden ergänzen, oder auch welche, die geerbt wurden überschreiben.



Praxis

Nun schauen wir uns gemeinsam die Implementierung anhand der Beispielklasse Haus in Python an.