

PROPÄDEUTIKUM INFORMATIK

SOSE 2018

Martin Mehlhose

OBJEKTORIENTIERUNG

KLASSEN

- Klassen werden definiert mit `class "Klassenname":`
- Alle Anweisungen und Methoden zur Klasse eingerückt
- Ein Objekt von der Klasse erzeugen wir mit dem Konstruktor `Klassenname()`

KLASSEN

- von Klassen lassen sich konkrete Objekte ableiten
- Objekte können eigene Attribute und Methoden haben
- Attribute beschreiben Eigenschaften des Objektes
- Methoden beschreiben das "Verhalten" des Objektes
- Methoden sind Funktionen, die auf Objektinstanzen gelten

UML

- Zur Darstellung von Klassen und dem Zusammenhand zwischen Klassen nutzen wir UML Diagrammen
- Das UML-Diagramm eine Klasse enthält den Namen, alle Attribute, und alle Methoden
- Hinter Attributen wird der Datentyp vermerkt
- Hinter Funktionen wird der Rückgabotyp vermerkt

UML

- +: bedeutet das Attribut/ die Methode ist public
- -: bedeutet das Attribut/ die Methode ist privat (gibt es in Python nicht)
- Haben Methoden Parameter werden diese im UML auch mit Datentyp angegeben

OBJEKTE

- Zugriff auf Objektattribute und -methoden immer über die Punktnotation
- In sauberer Objektorientierung kein direkter Zugriff auf Variablen sondern nur über getter und setter Methoden

OBJEKTE

- Objekte werden über eine Konstruktormethode initialisiert:
- **`__init__(self, optionale Parameter)`**
- Im Konstruktor sollten alle Attribute initialisiert werden, damit das Objekt danach in einem validen Zustand ist

OBJEKTE

- gibt man ein Object in die print() Funktion wird nur die Hauptspeicheradresse der Instanz ausgegeben
- Mit der String Methode kann man das Verhalten ändern und die gewünschten Informationen ausprinten
- `__str__(self)`

OBJEKTE

- Objekte sind, wie schon die Listen, Referenzdatentypen
- hinter Variable steht also nur die Adresse nicht das Objekt selbst
- Daher wieder aufpassen beim Kopieren von Objekten

KLASSEN

- Es gibt noch Klassenattribute und -funktionen
- Diese gelten für die gesamte Klasse, nicht nur für konkrete Objecte
- Werden im UML durch Unterstreichen gekennzeichnet
- Aufruf wieder mit der Punktnotation auf dem Klassennamen

AGGREGATION UND VERERBUNG

AGGREGATION

- Ein Objekt kann nicht nur einfache Datentypen als Attribute besitzen, sondern auch aus anderen Objekten aufgebaut sein
- besitzt ein Objekt ein anderes Objekt als Attribut nennt man das Aggregation

VERERBUNG

- Wir können aus einer allgemeinen Elternklasse spezielle Kindklassen ableiten
- Das nennen wir Vererbung, da die Kindklasse alle Attribute und Methoden der Elternklasse erhält
- Hinter den Klassennamen schreiben wir in Python in Klammern die Elternklasse

VERERBUNG

- Im Kontruktor der Kindklasse muss der Konstruktor der Elternklasse aufgerufen werden, um alle Attribute zu Initialisieren
- Die Kindklasse kann um neue Methoden / Attribute ergänzt werden
- Es können auch einzelne Methoden neu geschrieben werden
- Es wird von python immer die "speziellste" Methode verwendet

