Einführung in Programmierung und Datenbanken

joern ploennigs

Modellierung mit Objekten

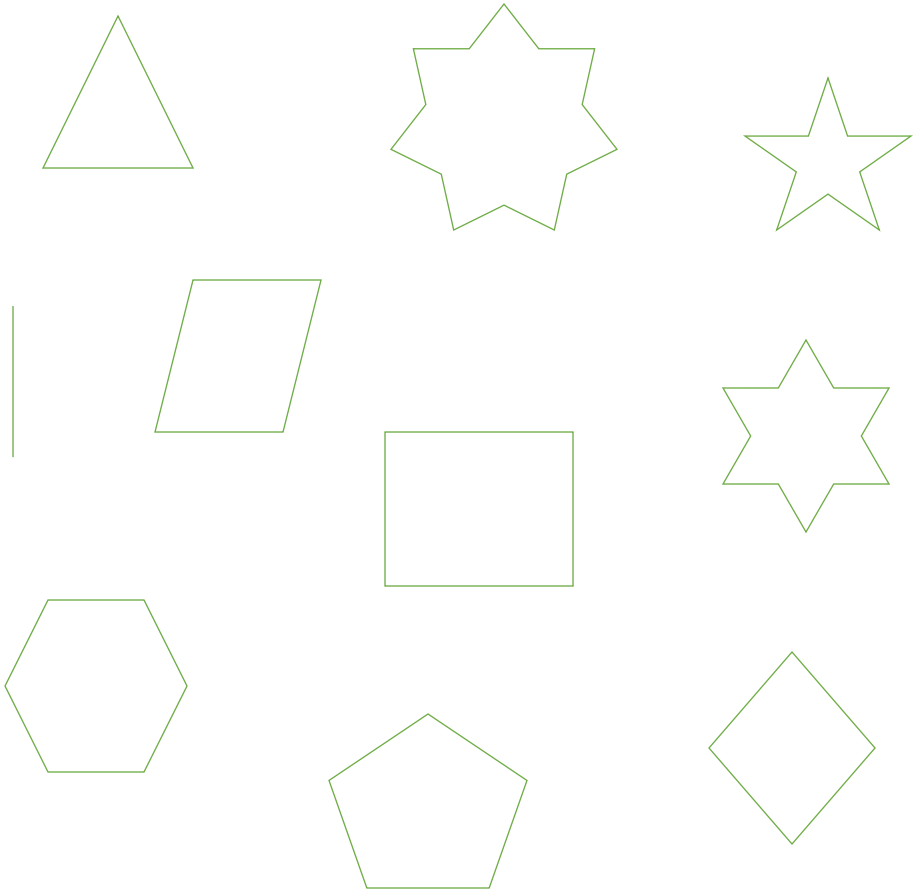
- Unsere Wahrnehmung und unser Denken besteht darauf Objekte zu erkennen. Das setzt voraus, dass wir das Objekt von anderen Objekten trennen können (z. B. Stuhl und Tisch)
- Das Nachbilden dieser Wahrnehmung und dieses Weltverständnisses im Computer, nennt man Modellierung. Ein **Modell** ist eine abstrakte, vereinfachte Abbildung eines Systems (meist aus der Realität). Die Eigenschaft eines **Systems** ist es, dass es immer von seiner Umwelt trennbar ist.
- Da unsere Wahrnehmung auf Objekten basiert, ist es naheliegend für die Modellierung auch Objekte zu nutzen.

- Die echte Welt hat Systeme die interagieren, die Verhaltensweisen müssen wir auch modellieren!

- Das Verhalten wird in Programmen meist durch Funktionen abgebildet, welche Variablen verarbeiten und ändern.

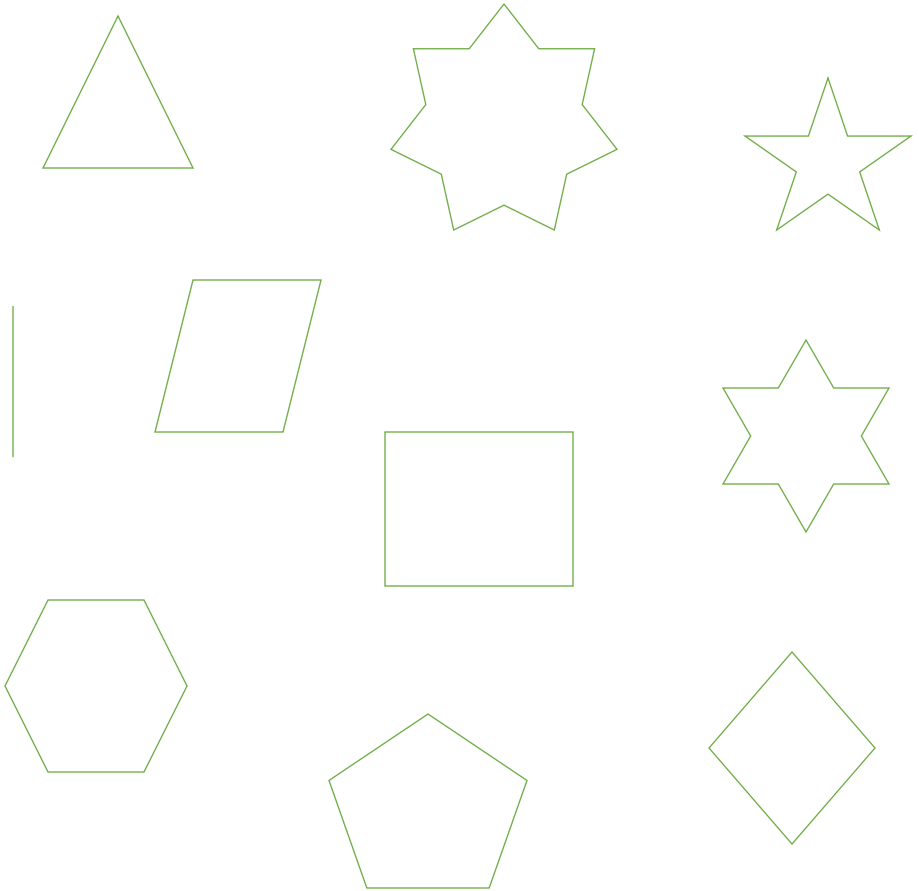
Hörsaalfrage

Was haben diese Elemente gemein?



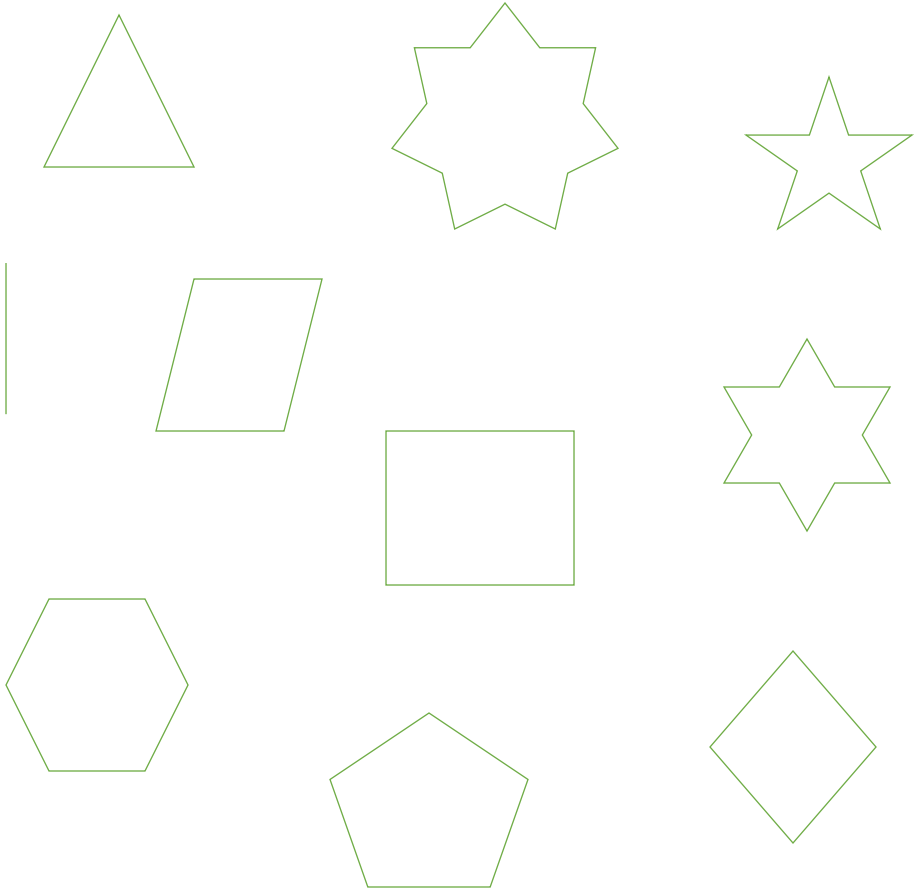
Hörsaalfrage

Woraus bestehen diese Elemente?



Woraus bestehen diese Elemente?

- Das sind alles geometrische Objekte
- Sie bestehen alle aus Punkten, die mit Linien verbunden sind
- Das können wir bei der Modellierung ausnutzen und die Klassen so definieren, dass wir die Gemeinsamkeit in der Konstruktion ausnutzen



Objektorientierter Softwareentwurf

Im objektorientierten Softwareentwurf, wird ein Programm so entworfen, dass es nur aus Objekten besteht. Man modelliert im Entwurf wie:

- diese Objekte in Form von **Klassen** definiert sind,
- welche **Attribute** und **Methoden** sie besitzen,
- wie diese Klassen aufeinander aufbauen (**Vererbung**),
- als auch wie sie miteinander in statischer Beziehung stehen (**Referenzen**)
- und wie sie dynamisch interagieren (**Verhalten**)

Der Softwareentwurf gleicht dabei stark dem Vorgehen von Ingenieuren beim Erstellen von Plänen, z. B. von Bauplänen.

Die übliche Modellierungssprache sind UML Diagramme.

Objektorientierter Softwareentwurf - Referenzen

- In Programmen stehen Objekte meist im Zusammenhang. Z. B. besteht eine Linie aus zwei Punkten.
- Diesen Zusammenhang zweier Objekt-Klassen bezeichnet man als Referenz.
- Um Referenzen in Python zu erstellen, erstellt man einfach ein Attribut vom Datentyp des anderen Objektes.
- So kann man eine Linie als Verbindung zweier Punkte und die Länge definieren

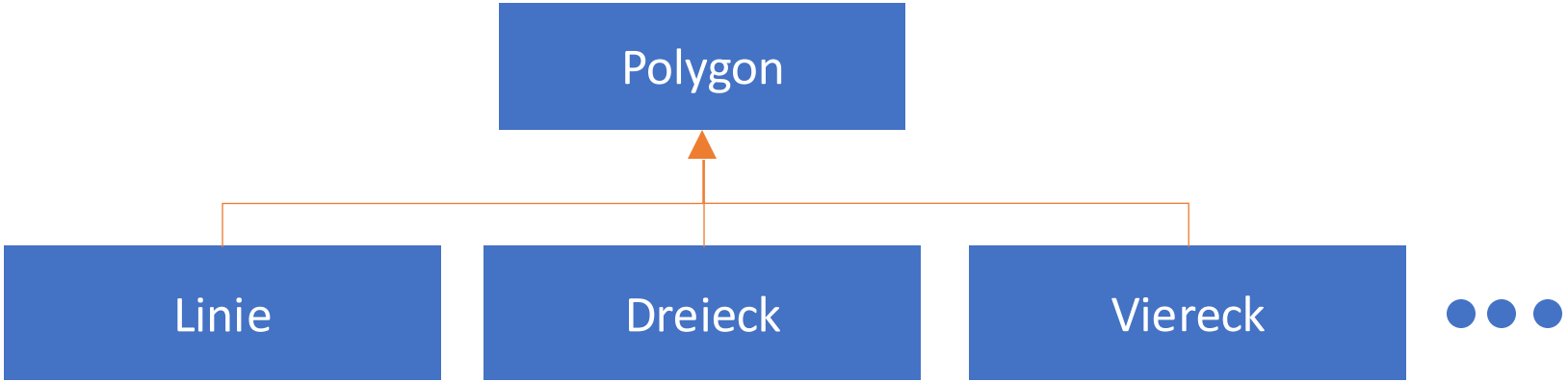
```
class Line:
    def __init__(self, start: Point, end: Point):
        self.start = start
        self.end = end

    def length(self):
        return self.start.distance(self.end)
```

Objektorientierter Softwareentwurf - Vererbung

- Gleiche Objekte teilen oft auch ähnliche Eigenschaften und Methoden. Z. B. bestehen alle geometrischen Elemente aus Punkten die mit Linien verbunden sind
- Um diese Attribute und Methoden nicht jedes mal neu definieren zu müssen dabei ggf. Fehler zu machen, gibt es die Vererbung in der OOP
- OOP erlaubt das Definieren von spezialisierten Unterklassen, bzw. von generalisierten Oberklassen.
- Die Unterklasse (Spezialklasse) besitzt die Methoden und Attribute der generalisierten Oberklasse (Basisklasse).
- Angelehnt an die Genetik auch „Vererbung“ genannt

- Die abgeleitete Klasse erbt Methoden und Attribute der Basisklassen, aber kann auch zusätzliche Methoden und Attribute enthalten.

Beispiel Vererbung



Objektorientierter Softwareentwurf - Polymorphie

- Ein Objekt einer Spezialklasse kann stets auch als Mitglied der Basisklasse betrachtet werden.
- Die Spezialklasse kann die geerbten Methoden/Attribute der Basisklasse überschreiben und somit umdefinieren.

- In statisch typisierten Sprachen gilt außerdem: In einer Variablen die ein Objekt der Basisklasse aufnehmen kann, kann auch ein Objekt einer abgeleiteten Klasse gespeichert werden.

Paradigmen der Objektorientierung - Datenkapselung

- Ziel der Datenkapselung ist es Kontrolle darüber zu haben, welche Attribute lesbar oder beschreibbar sind und wie dies geschehen kann
- Hierfür deklariert man Attribute oder Methoden als privat, protected, und public
  - privat – Nur die Instanz der gleichen Klasse kann zugreifen
  - protected – Nur die Instanz der gleichen Klasse oder Unterklasse kann zugreifen
  - public – Alle können zugreifen
- Ferner kontrolliert man ob Attribute nur lesbar oder auch schreibbar sind. Dafür werden Attribute als privat deklariert und sind dann nur durch Get-Funktionen lesbar und durch Set-Funktionen veränderbar.

Dynamik in Klassen

- In der OOP wird angenommen, dass Klassen statisch sind und es keine anderen als die in der Klasse definierten Attribute und Methoden gibt.
- Python bietet ein höheres Maß an Dynamik als andere Programmiersprachen.
  - Dynamische Typisierung von Variablen
  - Dynamisches Binden von Methoden und Attributen

- Wir können sogar Attribute und Methoden selbst hinzufügen, wenn diese nicht in der Klasse definiert sind.
- So können wir z. B. zur Laufzeit Referenzen zwischen Objekten schaffen die im vordefinierten Datenmodell keinen Bezug haben.

UNIFIED MODELLING LANGUAGE - GRUNDLAGEN

UML (Unified Modelling Language) ist ein ISO-Standard mit dem Softwareentwürfe dokumentiert werden.

- Genutzt für den Entwurf, die Ausschreibung, Implementation und Dokumentation von:
  - Funktionalitäten
  - Strukturen
  - Prozessen und Interaktionen
- Der Standard umfasst 13 Diagrammtypen, die typischsten Diagramme sind:
  - *Klassendiagramme (Für die Klassen- und Datenstruktur)*
  - *Programmablaufplan (Für allgemeine Abläufe und Algorithmen, Siehe Vorlesung 4)*
  - Sequenzdiagramme (Für Interaktionen, https://de.wikipedia.org/wiki/Sequenzdiagramm)
  - Use-Case Diagramme (Für Nutzer und Nutzfälle, https://en.wikipedia.org/wiki/Use_case_diagram)

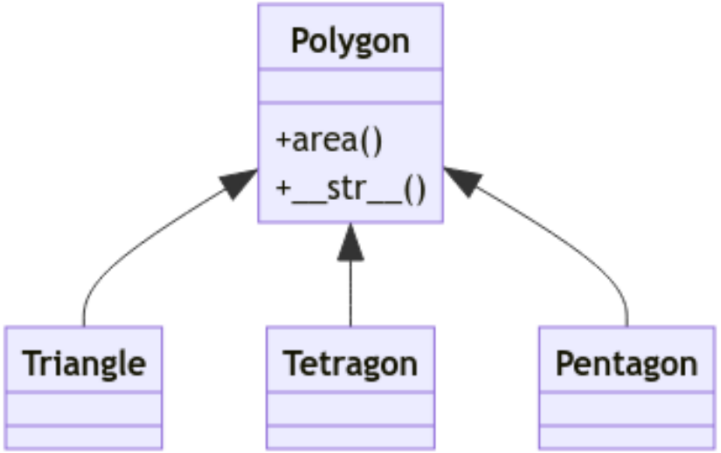
Unified Modelling Language - Klassendiagramm

- Das am häufigsten verwendete Diagramm im objektorientierten Modellieren

- Im Klassendiagramm modelliert werden:
  - Klassen mit
    - Attributen
    - Methoden
    - Beziehungen
  - Hierarchien & Vererbungen
  - …

Klassendiagramm – Vererbung

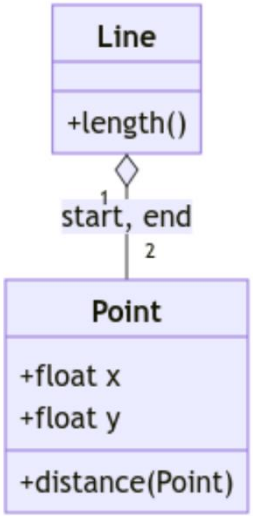
Vererbung wird in UML durch ein Referenz mit einem gefüllten Dreick "▲" bei der Oberklasse gezeichnet. Um zum Beispiel auszudrücken, dass `Triangle`, `Tetragon` und `Pentagon` Unterklassen des Polygons sind, können wir zeichnen.



Quelle: de.wikipedia.org/wiki/Klassendiagramm

Klassendiagramm - Referenzen

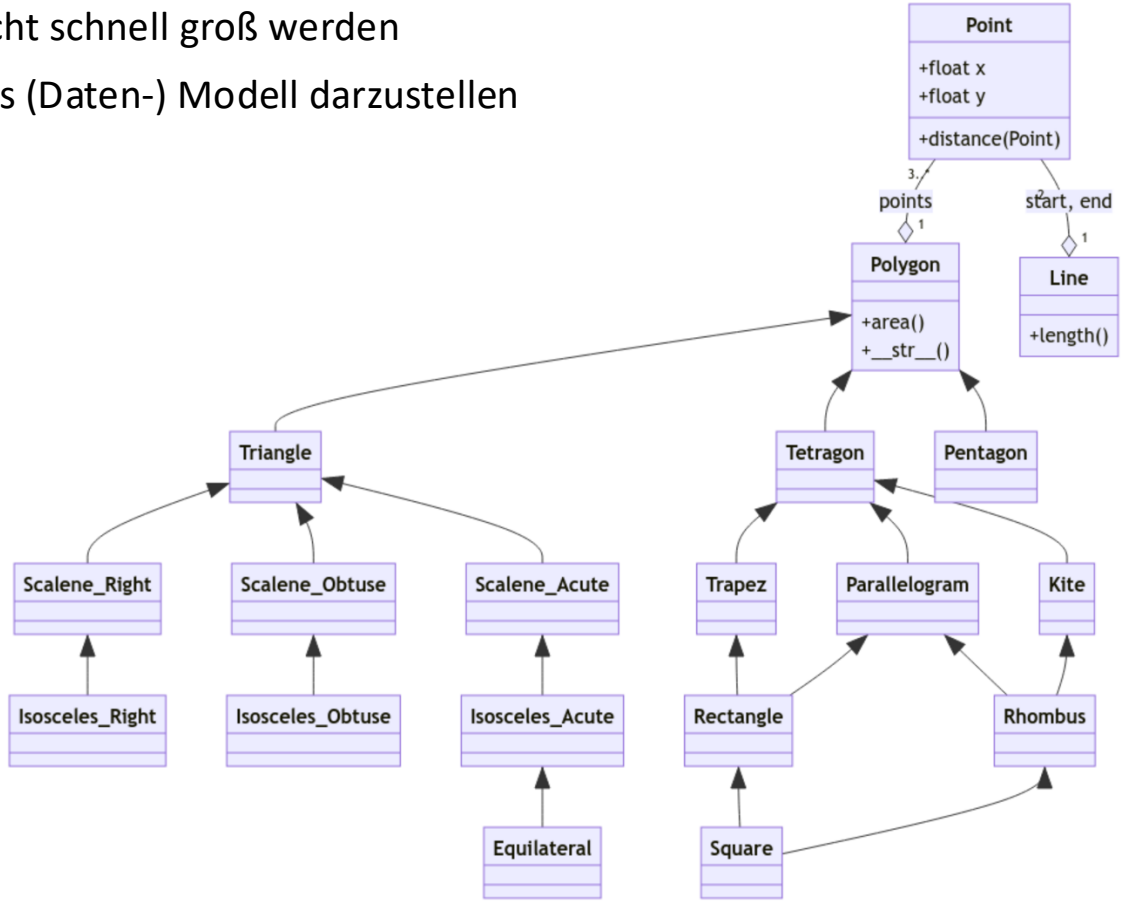
- Referenzen werden in UML mit Linien zwischen Objekten gekennzeichnet. Die Art der Linie und des Pfeils geben den Typ der Referenz an.
- Man unterscheidet in UML die Referenzen in der Art der Besitzverhältnisse, also ob ein Objekt Teil eines anderen ist und ohne ihn existieren kann (Aggregated) oder nicht ohne diesen existieren kann (Composition) oder komplett unabhängig ist (Assoziation)
- Zahlen an den Referenzen geben die Multiplizität an, also wie viele Objekte in dieser Relation im Zusammenhang stehen.



Quelle: de.wikipedia.org/wiki/Klassendiagramm

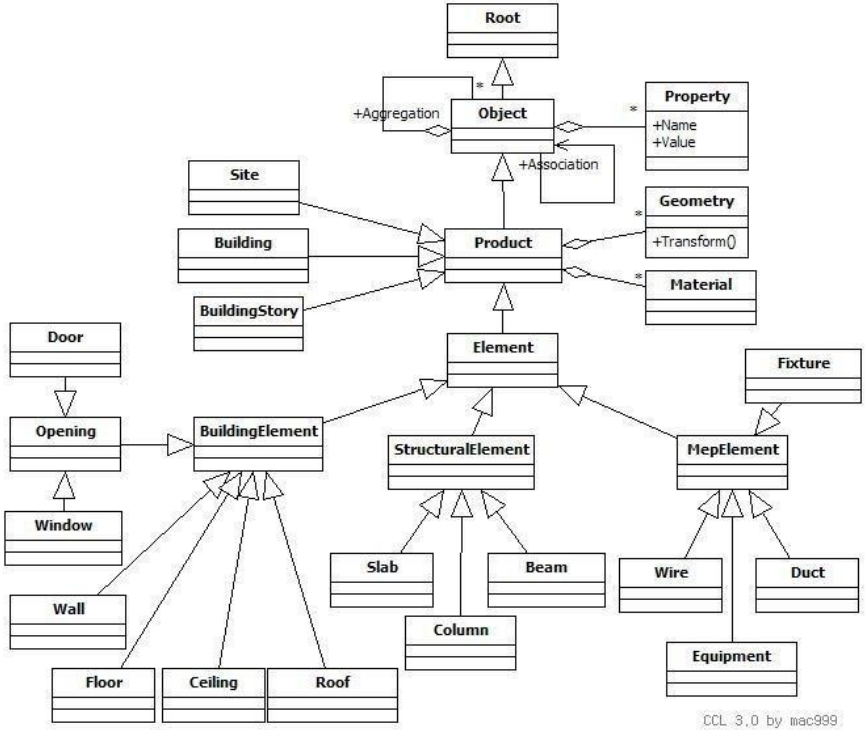
Klassendiagramm - Gesammtbeispiel

- Klassendiagramme können recht schnell groß werden
- Sie eignen sich sehr gut um das (Daten-) Modell darzustellen



Anwendungsbeispiel – BIM (Building Information Models)

- Bauzeichnungen werden heutzutage als BIM (Building Information Model) gespeichert
- IFC ist ein objektorientiertes Model, mit Vererbung, Spezialisierung, Generalisierung und Polymorphismus
- Es wird u.a. in UML dokumentiert



LITERATURHINWEISE

- Technische Einführung:
  https://docs.python.org/3/reference/datamodel.html

- Einsteigerfreundliche Erklärung:
  https://realpython.com/python3-object-oriented-programming/

Hörsaalfrage

FRAGEN?

Midjourney: A psychedelic DJ with a question mark for a head