



Übungen – OOP Grundlagen in Python

Übung 1: Klassen & Objekte in Python: Musikinstrumente

Sie arbeiten für eine digitale Musikplattform, die verschiedene Musikinstrumente verwaltet. Ihre Aufgabe ist es, ein passendes Klassensystem dafür zu erstellen.

Erstellen Sie eine Klasse `Instrument`, mit den **Attributen**:

name: Der Name des Instruments (z. B. "Gitarre", "Klavier")

typ: Der Typ des Instruments (z. B. "Saiteninstrument", "Tasteninstrument")

Fügen Sie einen Konstruktor hinzu, der diese Werte speichert.

Erstellen Sie eine Methode `spiele()`, die **beide Attribute verwendet** und eine Nachricht wie diese ausgibt:

"Die **Gitarre** (ein **Saiteninstrument**) spielt ein ruhiges Lied."

Erstellen Sie mindestens zwei unterschiedliche **Objekte**, z. B. ein `Saiteninstrument` und ein `Tasteninstrument`, und rufen Sie die Methode `spiele()` jeweils auf.



Übungen – OOP Grundlagen in Python

Übung 2: Vererbung in Python: Smart Home Geräte

Sie entwickeln eine Steuerzentrale für ein Smart Home System, das verschiedene smarte Geräte verwaltet. Ihre Aufgabe ist es, ein geeignetes Vererbungskonzept zu programmieren.

Erstellen Sie eine Oberklasse `SmartDevice` mit den folgenden **Attributen**:

name → Der Name des Geräts (z. B. "Lampe", "Thermostat")

status → Ob das Gerät "an" oder "aus" ist (Standard: "aus")

Fügen Sie eine Methode `zeige_status()` hinzu, die den Gerätestatus anzeigt.

Die Ausgabe sollte z. B. so aussehen: "Lampe ist an."

Erstellen Sie eine Unterklasse `SmartLampe`, die zusätzlich das Attribut **helligkeit** besitzt (Wert z. B. 70 für 70 %).

Der Konstruktor von `SmartLampe` soll den Konstruktor der Oberklasse aufzurufen.



Übungen – OOP Grundlagen in Python

Übung 2: Vererbung in Python: Smart Home Geräte

Überschreiben Sie die Methode `zeige_status()` so, dass sie zusätzlich die Helligkeit ausgibt:

"Lampe ist an mit Helligkeit 70 %."

Erstellen Sie ein **Objekt** vom Typ `SmartLampe` mit:

Name: "Lampe"

Status: "an"

Helligkeit: 70

Rufen Sie die Methode `zeige_status()` auf und überprüfen Sie die Ausgabe.



Übungen – OOP Grundlagen in Python

Hilfreiche Ressourcen:

https://www.w3schools.com/python/python_classes.asp

<https://docs.python.org/3/tutorial/classes.html>

<https://www.geeksforgeeks.org/python-classes-and-objects/>

<https://www.python-lernen.de/vererbung-python.htm>

https://www.w3schools.com/python/python_inheritance.asp

<https://www.programiz.com/python-programming/inheritance>