

## Termin 3 (Stichworte)

### Wiederholung

- Präsentationen in *OO-Allgemein* zu **run** und `__main__` zeigen

### Bewertung des aktuellen Schritts (weitere Klassen)

- Präsentation **OO-Python-P01-g Weitere Klassen – Bewertung.pdf**
- Wiederholtes Copy and Paste zeigt, dass man abstrahieren sollte. Termin-01-8

### Abschnitt 2: Abstraktion

- Codeduplizierung ist Hinweis auf Mängel der Modellierung
- Entscheidung für Lösung ohne Kapselung!  
( mit Kapselung geht es hier etwas anders, besser nicht machen ) Termin-02-2
- Projekte in *Moebel-erarbeiten-aus-Anfangsprojekt.zip* , Hinweis auf Präsentation **OO-Python-P02-1 Schritte-zur-Vererbung.pdf** und Bilderserie **Bilder-Moebel-aus-Anfangsprojekt.pdf**
- `tisch.py` → `moebel.py` entsprechend bearbeiten Termin-02-3
- **tisch.py**  
dazu passend bearbeiten
- Zugriff von Methoden besprechen
- Alles funktioniert, aber ist das ok so? Termin-02-4
- besser machen!  
Attribute nur an einer Stelle,  
Konstruktor Oberklasse aufrufen Termin-02-5
- Andere Klassen anpassen, (ggf noch komplette Lösung vorstellen)  
**OO-Python-P02-2 Vererbung-einfache-Loesung** Termin-02-6
- wichtig: **self** als Parameter erläutern!  
Präsentation **OO-Python-P02-a self als Parameter.pdf** Termin-02-7
- Kapselung realisieren (Einzelarbeit/Gruppenarbeit?)  
=> *Raumplaner-Vererbung nicht abstrakt.zip* Termin-02-8
- Termin-02-9
- Termin-02-10
- Termin-02-11

anders  
bei  
Kapselung

### Kohäsion

- Transformiere ausgliedern → Kohäsion **OO-Python-P02-c Kohäsion.pdf**  
Hinweis auf Text **OO-Python-03 Kohäsion und Kopplung.pdf**

### Abstraktion als Eigenschaft

- Es gibt keine Instanz **Moebel** an sich ! Termin-02-10  
Lösungsansätze
  - `GibFigur()` in `Moebel` leer implementieren  
→ sinnloses Initialisieren von `Moebel` bleibt möglich
  - `GibFigur()` in `Moebel` weglassen  
→ Fehler beim Initialisieren von `Moebel`

- GibFigur() in Moebel als abstrakt kennzeichnen
  - → Klasse Moebel selbst muss abstrakt sein und
    - Initialisieren führt zu sinnvoller Fehlermeldung
- Präsentation ***OO-Python-P02-b Moebel\_ist\_abstrakt\_Py3.pdf***  
Umsetzen im Programm und austesten:

```
from abc import ABC, abstractmethod
class Moebel(ABC):
```

### ***Schrittweise Ausführung***

- die Varianten vorstellen ***OO-Python-P02e schrittweise.pdf***
  - timer und Bind
  - externe Klasse Schritte
  - Threads verwenden