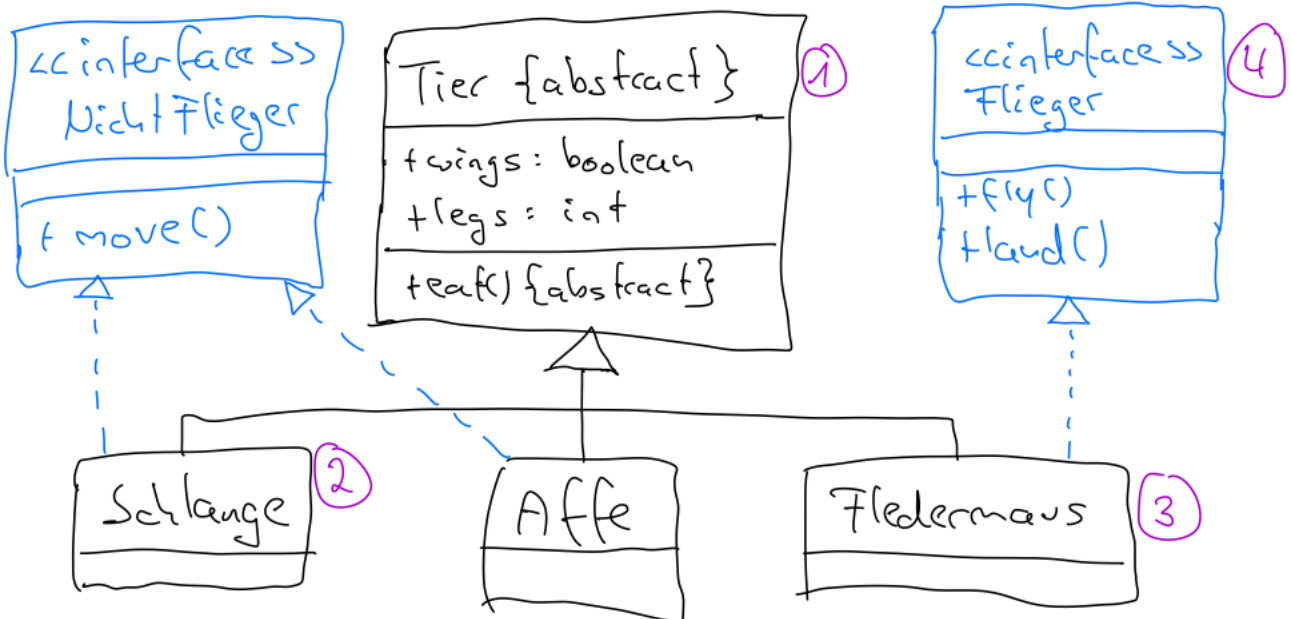




Interfaces = Schnittstellen

Ziel: Spezifikation von Klassen von der Implementierung trennen.

Beispiel: Anwendung mit Tieren



in Python:

```
from abc import ABC, abstractmethod
```

① class Tier(ABC):

...
←

④ class Flieger(ABC):

② class Schlange(Tier, NichtFlieger):

③ class Fledermaus(Tier, Flieger):

- Eine Klasse kann eine beliebige Anzahl an Interfaces implementieren.
- Interfaces ersetzen das Konzept der Mehrfachvererbung.
- keine (aus)implementierten Methoden
- keine Konstruktoren
- Variablen / Konstanten sind erlaubt
- Interfaces können nur von anderen Interfaces erben.



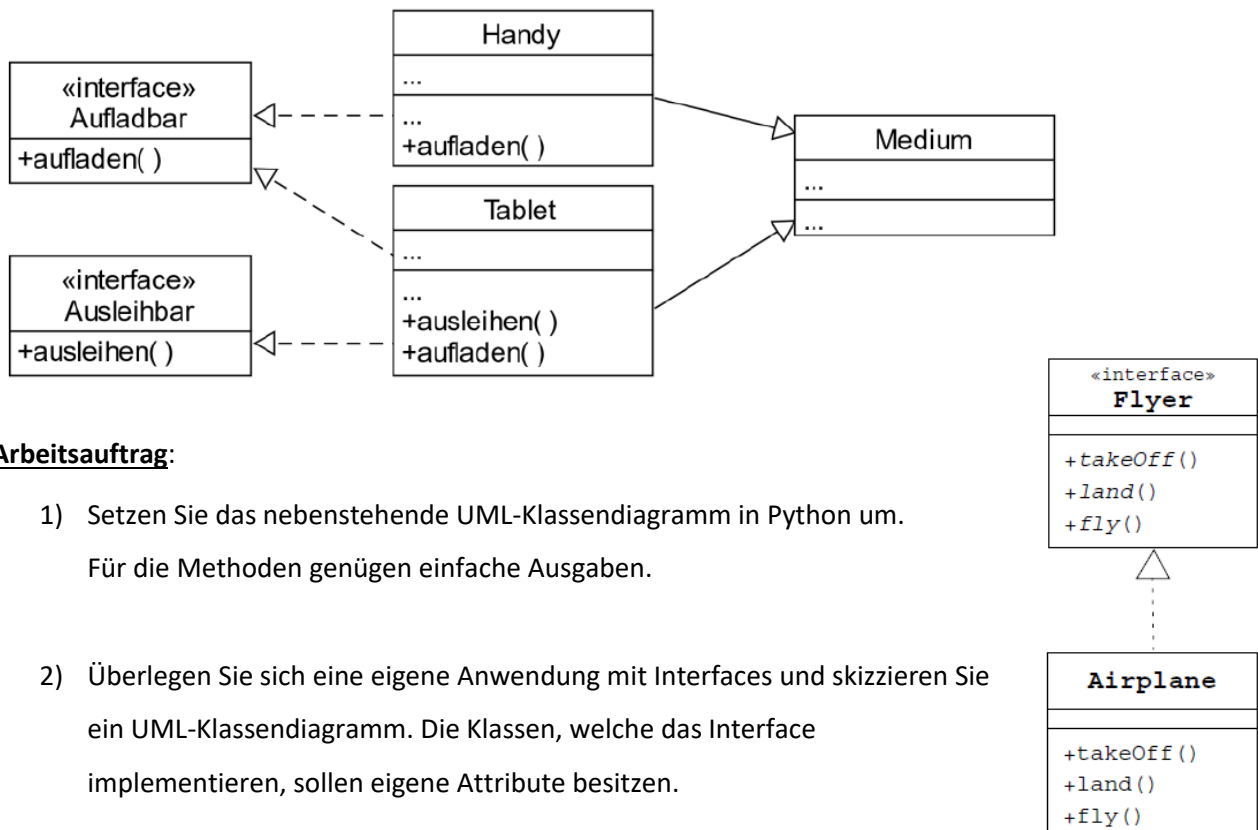
Abstrakte Klassen vs. Interfaces



Klassenhierarchie, um gemeinsame Attribute und Methoden an Unterklasse zu vererben, wo diese spezifisch erweitert, überschrieben (ggf. erst implementiert) werden.

gemeinsames Verhalten von Klassen unterschiedlicher Vererbungshierarchien.

Beispiel aus der Formelsammlung:



Arbeitsauftrag:

- 1) Setzen Sie das nebenstehende UML-Klassendiagramm in Python um.
Für die Methoden genügen einfache Ausgaben.
- 2) Überlegen Sie sich eine eigene Anwendung mit Interfaces und skizzieren Sie ein UML-Klassendiagramm. Die Klassen, welche das Interface implementieren, sollen eigene Attribute besitzen.



Vorteile von Interfaces





