**Interfaces**

- Stellen eine besondere Form von Klasse da

- Definieren lediglich den Aufbau einer Schnittstelle: Name von Methoden

Signaturen der Methoden

Attribute

- Funktionalität wird von Klassen implementiert, die das Interface nutzen wollen

- beinhaltet ausschließlich abstrakte öffentliche Methoden

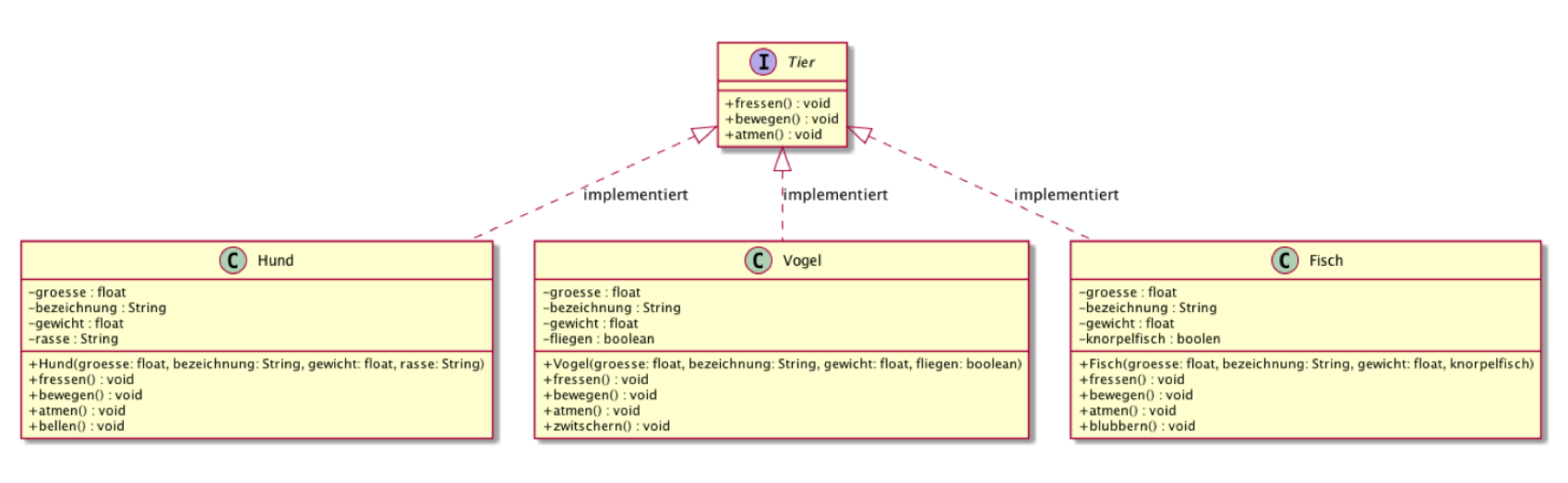
- Alle Attribute sind als Konstanten definiert

- Deklaration durch Schlüsselwort interface

- können von anderen Interfaces mit extends abgeleitet werden

- können als Datentypen für Referenzvariablen benutzt werden

Darstellung in UML



Implementierung

- um einem Interface Funktionalität zu geben, muss eine Klasse das Interface implementieren

class zxy implements abc { }

- Die Klasse muss alle abstrakten Methoden implementieren oder als abstrakte Klasse deklariert werden

- Klassen können mehrere Interfaces implementieren (Mehrfachvererbung ähnlich)

class xyz implements abc, def { }

Polymorphie durch Interfaces

- analog zu Polymophie durch Vererbung

- Unterschied: Vererbung: Objekte der Subklasse werden in Referenzvariable vom Typ der

Superklasse gespeichert

Interfaces: Objekte der implementierenden Klasse werden in Referenzvariablen vom Typ des Interface gespeichert