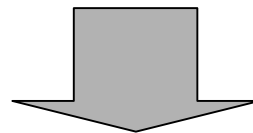


1.3 Charakteristische Eigenschaften von objektorientierten Systemen

- Einkapselung (Encapsulation)
- Geheimhaltungsprinzip (Information / Implementation hiding)
- Persistenz (State retention)
- Objektidentität (Identity)
- Nachrichten (Messages)
- Klassen (Classes)
- Polymorphismus (Polymorphism)
- Vererbung (Inheritance)



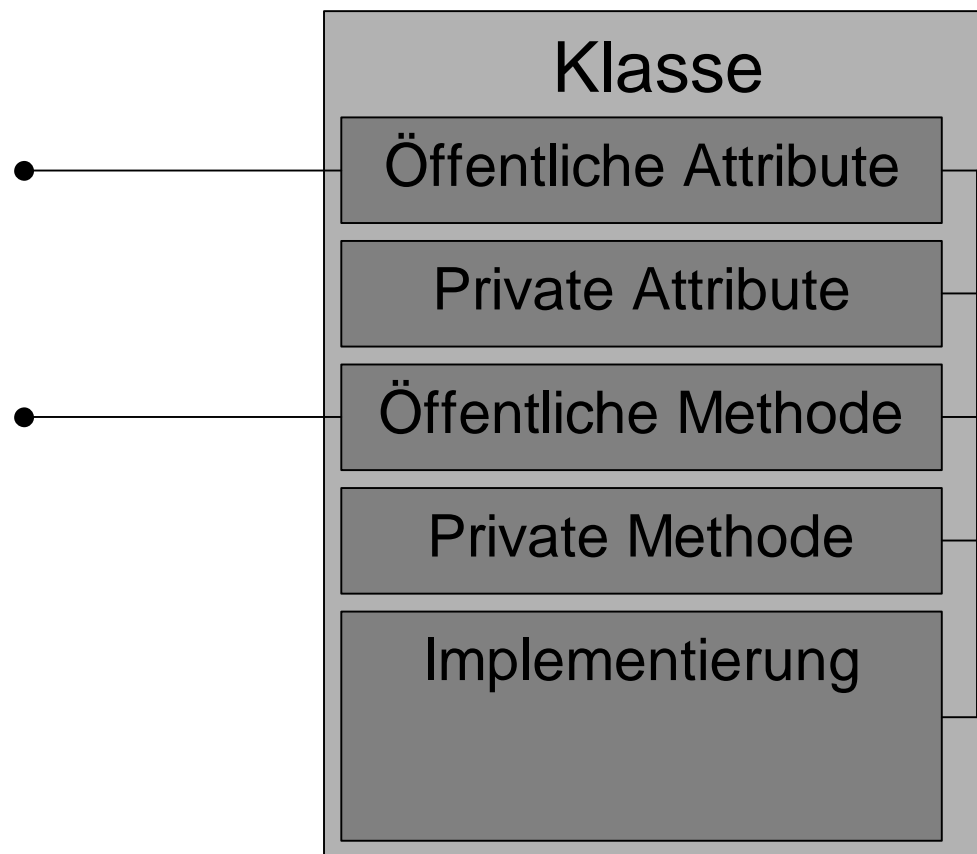
Nur dann echt objektorientiert!

1.3.1 Einkapselung

Encapsulation is the packaging of operations and attributes representing state into an object type so that state is accesible or modifiable only via the interface provided by the encapsulation

- Eine Klasse kann bestimmte Attribute und Operationen, die nur von der Klasse selbst benötigt werden, „verstecken“, so daß sie nach außen nicht sichtbar sind.
- Die Attribute und Operationen, die nach außen sichtbar sind, bilden die **öffentliche Schnittstelle** einer Klasse.
- Die Attribute und Operationen, die nach außen *nicht* sichtbar sind, bilden die **private Schnittstelle** einer Klasse.
- Die *Implementierung* der Operationen einer Klasse sind immer eingekapselt, d.h. von außen nicht sichtbar bzw. zugänglich.

- Öffentliche und private Attribute und Schnittstellen



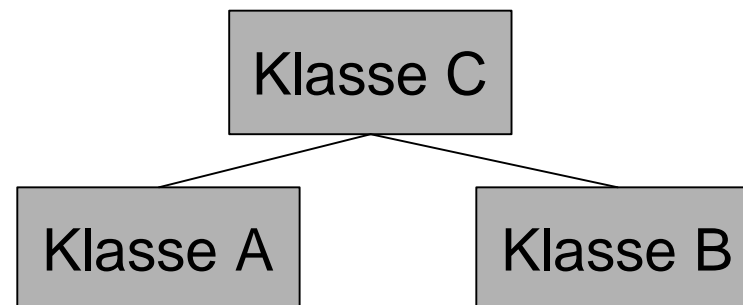
1.3.2 Geheimhaltungsprinzip (Information/Implementation hiding)

Information/Implementation hiding is the use of encapsulation to restrict from external visibility certain information or implementation decisions that are internal to the encapsulation structure.

- Ein Objekt erscheint wie eine *Black Box* für einen äußeren Beobachter
- Der externe Beobachter hat das genaue Wissen über das **was** ein Objekt tut aber nicht **wie** es es tut und wie es intern strukturiert ist.
- z.B.: autonomer Roboter bewegt sich in der realen Welt
 - Richtungsangaben an den Roboter erfolgen in Grad
 - Interne Rechnung in beliebigem Koordinatensystem

1.3.3 Vererbung

- Klassen haben große Ähnlichkeit mit anderen Klassen, d.h. sie haben eine Anzahl von gleichen Attributen und Operationen.
- Diese strukturelle Ähnlichkeit zwischen zwei Klassen A und B kann ausgenutzt werden.
- es wird eine Klasse C gebildet, die die gemeinsamen Merkmale von A und B enthält von dieser Klasse können die Klassen A und B erben, d.h. sie bekommen sämtliche Attribute und Operationen von C und müssen dann nur noch um ihre spezifischen Merkmale ergänzt werden.

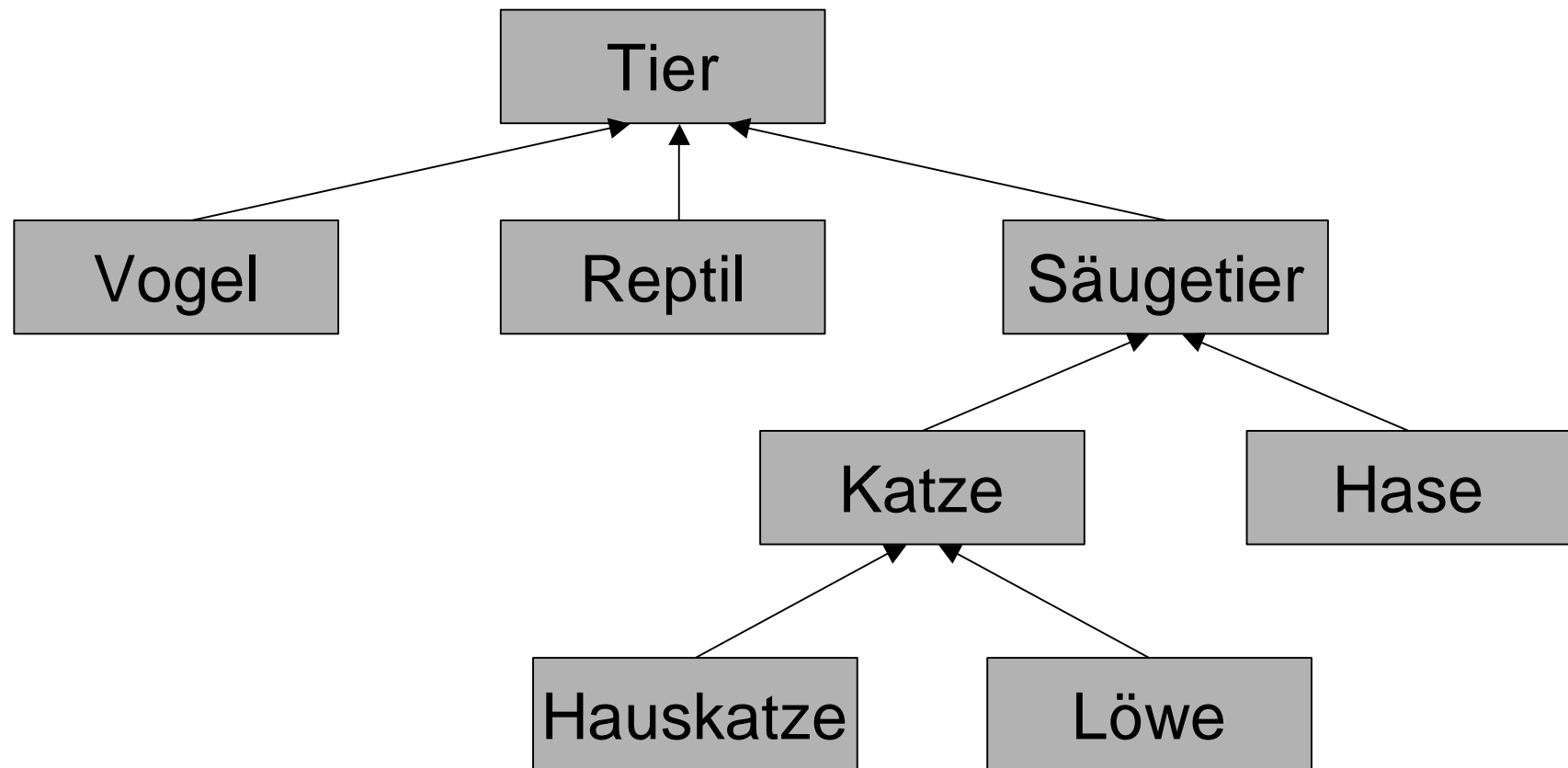


Beispiel für Vererbung

- Die Klasse *Student* hat die Attribute Name, Adresse und Matrikelnummer
- Die Klasse *Angestellter* hat die Attribute Name, Adresse, Sozialversicherungsnummer, Gehalt
- Die Klasse *Person* hat die Attribute Name, Adresse
- Für die Klassen *Student* und *Angestellter* müssen nicht alle diese Attribute definiert werden; sie können von der Klasse *Person* erben, so daß
 - *Student* nur um die Matrikelnummer ergänzt werden muß
 - *Angestellter* nur um die Attribute Sozialversicherungsnummer und Gehalt ergänzt werden muß

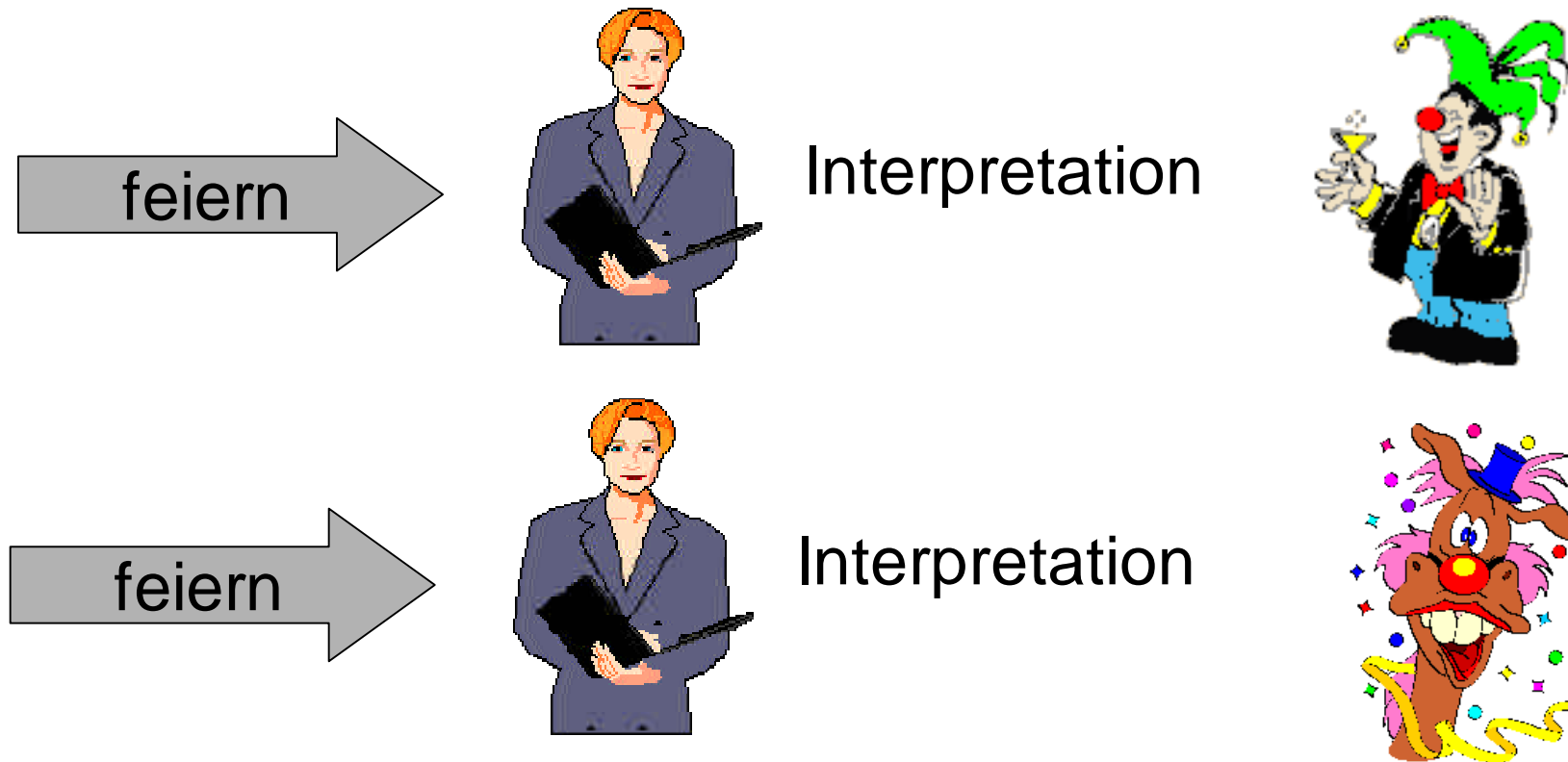
Noch ein Beispiel für Vererbung

- Tierwelt



1.3.4 Polymorphie

- Die Auslösung verschiedener Algorithmen durch die gleiche Nachricht wird als Polymorphismus bezeichnet



1.3.5 Persistenz

- Ein Objekt hat Kenntnis über seine Vergangenheit
- Die Werte für Attribute die ein Objekt hält werden über die gesamte Lebenszeit des Objekts „gespeichert“
- z.B.: Der Roboter behält die Kenntnis über seinen Ort und die Richtung oder der Student über das Semester

Paula
Geburtsdatum: 17.11.77 Einkommen: Stipendium Semester: 3.
lernen feiern

1.3.6 Objektidentität

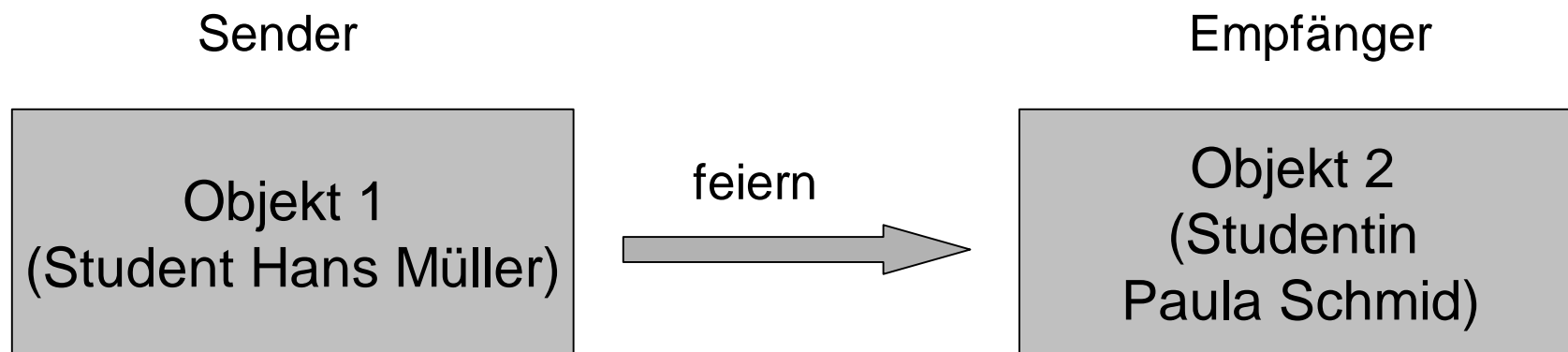
Object identity is the property by which each object (regardless of its class or current state) can be identified and treated as a distinct software entity

- Objekte sind einzigartig und unterscheidbar
 - Durch ein *handle*
 - Durch den Ort, an dem sie sich physikalisch im Adressbereich des Rechners befinden.

Adresse		
134789	Objekt 1	<i>Handle id 23</i>
234355	Objekt 2	<i>Handle id 5</i>
345000	Objekt 3	<i>Handle id 45</i>

1.3.7 Nachrichten

*A message is the vehicle by which a sender object **obj1** conveys to a target object **obj2** a demand for object **obj2** to apply one of its methods.*



- Nachrichten können Argumente haben
- Es gibt unterschiedliche Formen von Nachrichten
 - Informative, interrogative, imperative messages