



University
of Basel

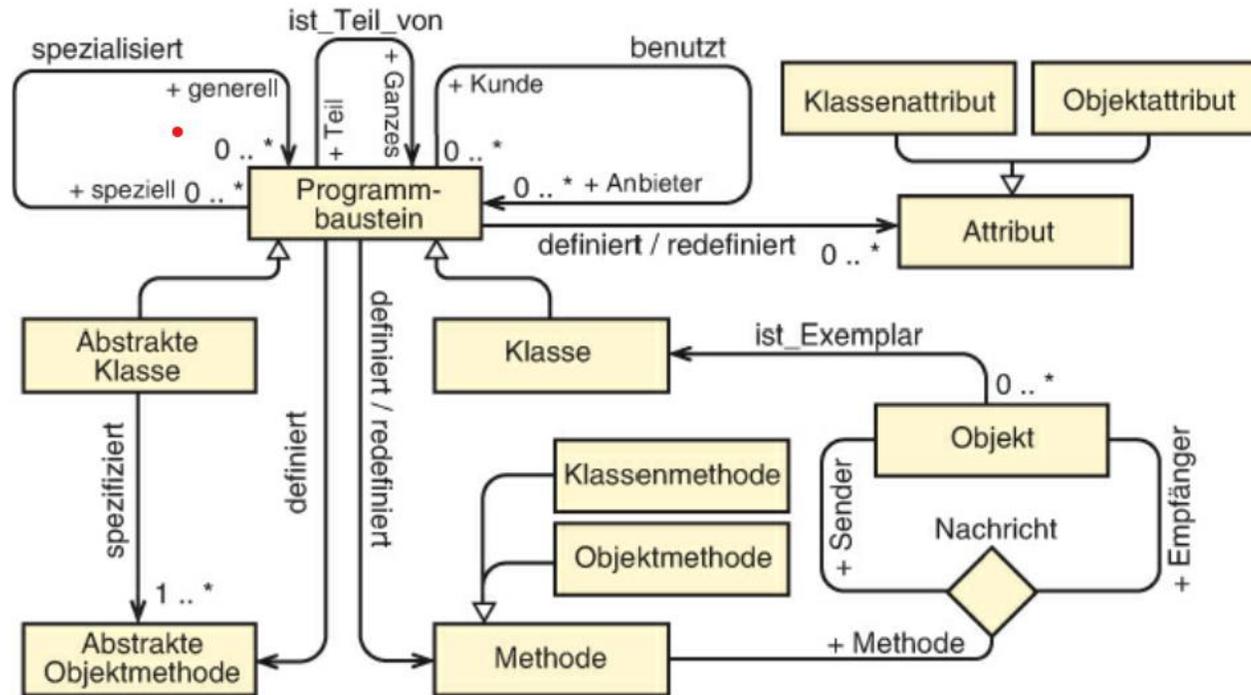
Software Engineering

Marcel Lüthi, Universität Basel

Objektorientiertes Design

Objektorientiertes Design

Entwurf eines Systems, unter Verwendung der objektorientierten Modellierungskonzepte



Quelle: Ludwig & Lichter, Software Engineering (Abb 17-10)

Dynamische Bindung und Polymorphismus

- **Polymorphismus:** Variablen können zur Laufzeit an verschiedene (verwandte) Typen gebunden werden
- **Dynamische Bindung** (late binding): Zur Laufzeit wird Methode entsprechend dem Objekttyp verwendet

```
Abstract class Dog {  
    public void bark();  
}  
  
class Poodle extends Dog {  
    public void bark() {  
        System.out.println("woof");  
    }  
}  
  
class GermanShepherd extends Dog {  
    public void bark() {  
        System.out.println("wau wau");  
    }  
}
```

Kann Poodle
Oder GermanShepherd sein

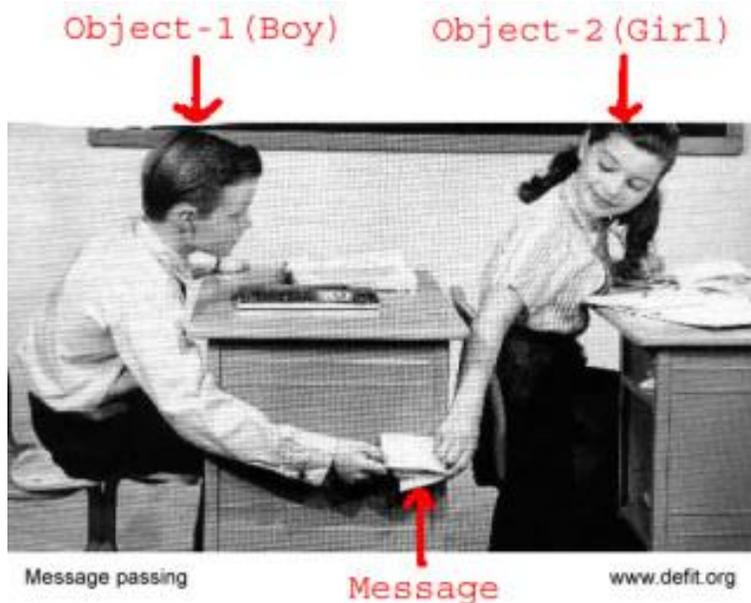
```
void bark(Dog dog) {  
    dog.bark();  
}
```

Methodenselektion je
nach Typ von dog

Objektorientierung

“OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things.”

Allan Kay



Quelle: www.defit.org/message-passing

Information Hiding: Zustand wird in Objekt gekapselt und von Zugriff von aussen geschützt

```
class Vector2D {  
    private double angle;  
    private double magnitude;  
  
    double getX() { // ... }  
    double getY() { // ... }  
    double getMagnitude() { // ... }  
    double getAngle() { // ... }  
}
```

Design und Programmiersprache

Wichtig beim OO Design sind Prinzipien (Information Hiding, Modularisierung, late binding), nicht die Programmiersprache.

OO Sprachen

- OO-Prinzipien und Elemente des OO-Designs werden auf Programmiersprachenkonstrukte abgebildet

Nicht OO Sprachen

- OO-Prinzipien und OO-Designs verlangen Mehraufwand und Hilfskonstrukte
-