

# Referenzattribute

Ein Attribut heißt **Referenzattribut**, wenn der Datentyp des Attributs eine andere schon bestehende Klasse ist (Gegenteil: primitives Attribut bei einem „normalen“ Datentyp). Es **referenziert/verweist** damit auf ein eigenständiges Objekt einer anderen Klasse.

Oft ist ein Referenzattribut ein Bauteil/Bestandteil: z. B. Klasse HAUS hat eine **Tür**, ein **Dach**, eine **Wand**,...

## Deklaration:

allgemein: **Klassenname Attributsname;**

Beispiele: **KREIS sonne;**  
**DACH dach;**

## Initialisierung:

Aufruf eines Konstruktors der Klasse mit dem **new**-Operator (inkl. Werte für die Parameter falls vorhanden)

allgemein: **Attributsname = new Klassenname();**

Beispiele: **sonne = new KREIS();** oder **sonne = new KREIS(3.5);** (KREIS besitzt mehrere Konstruktoren; beim zweiten kann man direkt den Radius des Kreises übergeben.)

**dach = new DACH();**

```
public class UNTERSCHIEDE{  
    //Deklaration  
    int x;    //primitives Attribut  
    KREIS k; //Referenzattribut  
    public UNTERSCHIEDE(){  
        //Initialisierung  
        this.x = 5;  
        this.k = new KREIS();  
  
        //nachträgliche Veränderung  
        this.x = 10;  
        this.k.setzeRadius(2);  
    }  
}
```

# Objektdiagramme

Objekte werden – wie bereits bekannt – durch **Objektkarten** veranschaulicht. Sie beinhalten neben dem Objektnamen und der dazugehörigen Klasse auch die Attribute sowie deren Werte zu einem bestimmten Zeitpunkt. Wir unterscheiden zwischen einfachen und erweiterten Objektkarten:

Beispiele:

einfache Objektkarte

**ball:KREIS**

erweiterte Objektkarte

**ball:KREIS**

*radius = 2.5*  
*farbe = "rot"*

Zwischen Objekten können (Referenz-) **Beziehungen** bestehen. Ein Objektdiagramm stellt diese Beziehungen grafisch dar. Falls einen nur die Beziehungen interessieren können die primitiven Attribute durch einfache Objektkarten weggelassen werden.

Beispiel mit erweiterten Objektkarten:

