Berufskolleg Technik Remscheid

Assoziation – Beziehung von Klassen

Unter einer Assoziation versteht man eine **Beziehung zwischen zwei Klassen A und B**.

Dabei kann Klasse A öffentliche Methoden (oder auch Variablen) von Klasse B verwenden. Gegebenenfalls ist dies auch umgekehrt der Fall.

Beispiel:

Eine Klasse Kunde kann eine Methode einer Klasse Ware aufrufen, um zum Beispiel den Namen der Ware oder den Preis der Ware ausgeben zu lassen.Eventuell kann der Kunde auf mehrere verschiedene Waren zugreifen.

Gegebenenfalls ist es auch sinnvoll, dass die Klasse Ware auf Elemente der Klasse Kunde zugreifen darf, zum Beispiel um den Namen eines Käufers zu erfragen.

Ein Objekt einer Klasse A kann bei einer Assoziation also auf ein oder mehrere Elemente von einer oder mehreren anderen Objekten einer zweiten Klasse B zugreifen.

Umsetzung des Beispiels in Java

Zunächst benötigt man ein Hauptprogramm, in dem die benötigten Objekte der verschiedenen Klassen erzeugt werden. Dies geschieht in der Regel in einer weiteren Klasse.

```
public class Kaufhaus {

        public static void main(String[] args) {

                Ware spielzeug = new Ware();   // Erzeugen der Objekte
                Kunde meier = new Kunde();

                meier.wareZuweisen(spielzeug);// Speichern der Adresse von
                                                // spielzeug in meier.
                meier.ausgabe();                // Ausgabe aller Daten
        }
}
```

Damit das Objekt meier (Typ Kunde) auf öffentliche Variablen und/oder Methoden des Objekts spielzeug (Typ Ware) zugreifen kann, muss die Klasse Kunde einen Verweis (engl. reference) auf ein Objekt der Klasse Ware speichern.

Zum setzen dieses Verweises muss eine entsprechende Methode in der Klasse Kunde vorhanden sein. Diese Methode wird mit meier.wareZuweisen(spielzeug); aufgerufen. Dabei wird die Adresse von spielzeug übergeben.

Vorhandene Konstruktoren können den Verweis explizit (ausdrücklich) initialisieren. Andernfalls wird die virtuelle Maschine den Verweis automatisch mit null vorbelegen.

```
public class Ware {
        String name;
        double preis;
        ...
        // weitere Methoden
}

public class Kunde {

        private Ware gekaufteWare;      // Das ist der Verweis auf ein Objekt
                                        // der Klasse Ware.
        // weitere Variablen

        public Kunde()                  // Konstruktor
        {
                gekaufteWare=null;   // Initialisierung der Referenz
                ...
        }

        public void wareZuweisen(Ware dieWare)
        {
                this.gekaufteWare=dieWare;  // Hier erst wird die konkrete
                                            // Adresse des Warenobjekts
                                            // gespeichert.
        }
```

Verweist nun die Referenz auf spielzeug, so kann über diesen Verweis eine öffentliche Methode des Objekts spielzeug aufgerufen werden. Zum Beispiel kann das Objekt meier den Preis der gekauften Ware über die Ausgabemethode ausgeben lassen.

Beispiel:

```
        public void ausgabe()       // Methode in der Klasse Kunde
        {
                System.out.println("Preis der Ware: "+gekaufteWare.getPreis());
        }
```

Weitere Anmerkungen:

1. Damit die Klasse Ware auch auf Methoden der Klasse Kunde zugreifen kann, muss ein Objekt der Klasse Ware einen Verweis auf die Klasse Kunde speichern.

2. Soll ein Objekt der Klasse Kunde auf mehr als ein konkretes Objekt der Klasse Ware zugreifen können, so müssen mehrere Verweise in geeigneter Art und Weise gespeichert werden. Dies geschieht zum Beispiel mit Hilfe von Arrays oder Listen. Von Ware zu Kunde lässt sich das ebenfalls mit diesen Mitteln realisieren.

3. An die Methode wareZuweisen wird ein Verweis auf ein Objekt übergeben. Bisher waren alle Übergabeparameter an Methoden von einem primitiven Datentyp.
   Wie bereits bekannt wird der Verweis – also die Adresse auf das andere Objekt – kopiert.

   Da es sich hierbei aber um eine Adresse im Hauptspeicher handelt, ergeben sich etwas andere Konsequenzen als bei der Übergabe von primitiven Datentypen. Näheres erläutert das Infoblatt „IB_ParameterübergabeAnMethoden_Verweise.pdf“