

Algorithmen am Beispiel „Java“ II

Klassenmethoden (s. auch Skript „Java“ I: S. 2 f.)

In objektorientierten Programmiersprachen werden häufig benutzte Algorithmen (Sequenzen von Befehlen) zu so genannten **Methoden** zusammengefasst. Methoden werden unter Nennung ihres Namens und ggfs. mit Übergabe von Parametern aufgerufen werden (bei imperativen Programmiersprachen spricht man von **Prozeduren**)

a) Methoden sind entweder *objektbezogen* oder *klassenbezogen*. Eine mit dem Schlüsselwort **static** versehene Deklaration einer Klassenvariablen führt eine klassenbezogene Variable ein, welche in allen Methoden dieser Klasse sichtbar ist.

b) Die Zugriffsrechte werden über so genannte *Modifier* geregelt: Durch den Modifier **private** wird der Zugriff nur von der eigenen Klasse ermöglicht, durch den Modifier **public** darf von einer beliebigen anderen Klasse zugegriffen werden.

Gibt eine Methode einen Wert zurück, so steht sein Typ vor dem Methodennamen, andernfalls steht dort **void**.

Die Kombination aus Methodennamen und den formalen Parametern bezeichnet man auch als *Signatur*. Beispiel: **public static void main**

```
/* ***** Methoden.java ***** */
import AlgoTools.IO;

public class Methoden {
    public static void bitte() {
        IO.println("Bitte Eingabe: ");
    }

    public static void sterne(int k) {
        int i;
        for (i=0; i < k; i++)
            IO.print('*');
        IO.println();
    }

    public static int zweihoch(int n) {
        int i, h = 1;
        for (i=0; i < n; i++) h = h * 2;
        return h; // liefere Ergebnis ab
    }

    public static int ggt(int a, int b) {
        while (a != b)
            if (a > b) a=a-b; else b=b-a;
        return a;
    }

    public static void main (String [] argv) {
        bitte();
        sterne(3*5+7);
        IO.println("2 hoch 7 ist " + zweihoch(7));
        IO.println("ggt(28,12) = " + ggt(28,12));
    }
}

/** Klassen-Methoden
 * mit und ohne formale Parameter
 * mit und ohne Rueckgabewert
 */
// Methode ohne Rueckgabewert
// und ohne Parameter
// Methode ohne Rueckgabewert
// mit einem Integer-Parameter
// lokale Variable
// zeichne k Sterne
// Methode mit Rueckgabewert
// mit einem Integer-Parametern
// lokale Variablen
// berechne 2 hoch n
// Methode mit Rueckgabewert
// zwei Integer-Parameter
// solange Zahlen verschieden
// ziehe kleinere von groesserer ab
// liefere Ergebnis zurueck
// aktuelle Parameter unveraendert
// Methode ohne Rueckgabewert
// oeffentlich aufrufbar
// rufe bitte auf
// rufe sterne auf
// rufe zweihoch auf
// rufe ggt auf
```

Java ist objektorientiert (s. auch Skript „Java“ II: S. 1 ff.), das heißt:

1. Die informatische Modellierung der **Realwelt** erfolgt durch **Objekte**.
2. Danach werden alle Objekte nach ihren entscheidenden Merkmalen in so genannten **Klassen** zusammengefasst.
3. Zu einer Klasse und ihren Objekten gehören deshalb auch entsprechende **Datenfelder** mit den Daten der betreffenden Objektmerkmale.
4. Diese die Objekte bestimmenden Daten werden Methoden genannt. Sie bestehen aus algorithmischen Anweisungsfolgen, die die besonderen Bestimmungen der Objekte einer Klasse definieren.
5. Die Anwendung einer bestehenden Methode erfolgt im Sinne von „Objekt A ruft Methode M von Objekt B auf“.

Klassen

Zur besseren Verwaltung gleichartiger Objekte bedient sich auch Java des Konzepts der Klasse. Klassen sind vergleichbar mit Vorlagen, aus denen Objekte erzeugt werden. Im Programm werden also nicht einzelne Objekte, sondern nur eine Klasse gleichartiger Objekte definiert.

Man kann sich die Erzeugung von Objekten aus einer Klasse vorstellen wie die Produktion von Autos aus dem Konstruktionsplan eines bestimmten Fahrzeugtyps. Klassen als „Konstruktionspläne“ für Objekte bestimmen alle Eigenschaften, die ein Objekt als zugehörig zu einer Klasse ausweisen.

Beispiel: Die allgemeine Klasse „Auto“ legt fest, dass jedes individuelle Auto vier Reifen, x Türen, einen Motor und y Sitze etc. hat.

Die Klasse entspricht in etwa einem komplexen Datentyp. Sie legt nicht nur die Datentypen fest, aus denen die mit Hilfe der Klassen erzeugten Objekte bestehen, sie definiert zudem die Algorithmen, die auf diesen Daten operieren. Während also zur Laufzeit eines Programms einzelne Objekte miteinander interagieren, wird das Grundmuster dieser Interaktion durch die Definition der einzelnen Klassen festgelegt.

Das aus der Klasse „Auto“ erzeugte Objekt „Automodell1“ hat schließlich vier Reifen mit dem Durchmesser 60 cm und der Breite 20 cm, fünf rote Türen, einen Motor mit 150 kW und fünf Ledersitze. Ein weiteres Objekt „Automodell2“ hat vier Reifen mit dem Durchmesser 40 cm und der Breite 15 cm, fünf blaue Türen ...

Beide Objekte sind unterschiedlich, gehören aber zu der gemeinsamen Klasse Auto.

Methoden

Die einer Klasse von Objekten zugeordneten Algorithmen bezeichnet man auch als Methoden. In Java unterscheidet man dabei u.a. in folgende Zugriffsarten:

1. Private Methoden (**private**) können nur von anderen Methoden derselben Klasse aufgerufen werden.
2. Öffentliche Methoden (**public**) dürfen von allen Klassen aufgerufen werden.

→ Fortsetzung in Arbeit ...