

Einstieg in die Programmierung von Computern – Teil III

Im Teil 3 der mehrteiligen Serie „Einstieg in die Programmierung von Computern“ lernen wir nun die Programmiersprache Java genauer kennen und befassen uns in diesem Teil auch mit Datentypen, Variablen, Klassen und Objekten. Unser erstes kleines Java-Programm im Teil 2 werden wir als Basis verwenden und das neu Gelernte dann schrittweise einbauen.

Java-Datentypen

Datentypen sind eine Art **Bauplan** für die sogenannten **Variablen**. Man weist einer Variablen einen bestimmten Datentyp zu. Damit wird im Arbeitsspeicher (bzw. Hauptspeicher) Speicherplatz für die Variable reserviert. Alle Variablen eines Datentyps haben die gleiche Anzahl von Speicherzellen. Dabei können verschiedene Variablen verschiedene Datentypen zugewiesen bekommen und belegen somit unterschiedlich großen Arbeitsspeicher. *Es gilt: Ein Datentyp definiert die Art der möglichen Inhalte einer Variablen und legt deren Wertebereich und Verwendung fest. Datentypen und Variable sind somit unzertrennbar und bilden gemeinsam eine Einheit.*

Eine Variable ist demnach ein reservierter Speicherbereich und belegt eine feste Anzahl von Byte. Es gibt **einfache Datentypen** wie

diesen nicht mehr ändern kann, **ist Java eine streng typisierte Sprache** (siehe Tabelle).

Es gilt: Die numerischen Datentypen byte, short, int, long und Fließkommazahlen sind vorzeichenbehaftet.

Was bedeutet Unicode?

Prinzipiell werden in Java geschriebene Programme im sogenannten **Unicode-Zeichensatz** dargestellt. Dieser ist ein international standardisierter Zeichensatz mit 16 Bit breiten Zeichen. Das Ziel des **Unicode-Konsortiums** ist es, alle Zeichen der Welt mit dem Unicode-Standard zu kodieren und dem Zeichen somit einen eindeutigen Codepoint zu geben. Der bekannte ASCII-Zeichensatz ist eine Untermenge von Unicode und die Zeichen des ASCII-Zeichensatzes werden in Unicode mit demselben Codewert dargestellt.

Typname Variablenname;

Es können auch mehrere Variablen gleichen Typs in einem Schritt deklariert werden. Diese müssen allerdings mit einem Komma getrennt werden:

Typname Variablenname1, Variablenname2, Variablenname3, ...;

Alphanumerische Zeichen

Der alphanumerische Datentyp char (von engl. Character: Zeichen) ist 2 Byte groß und nimmt ein Unicode-Zeichen auf. Ein char ist nicht vorzeichenbehaftet. Die Literale für Zeichen werden in einfache Hochkommata gesetzt. Sprachersteiger verwechseln häufig die einfachen Hochkommata mit den Anführungszeichen der Zeichenketten (Strings). Die einfache Merkregel: Ein Zeichen – ein Hochkomma, mehrere Zei-



© Dmitry Shironosov

hinzu. Dazu öffnen Sie mit dem Texteditor „Notepad++“ die Datei „HelloWorld.java“ und geben die Anweisungen ein. Das Ergebnis sehen Sie im Kasten 2. *Es gilt: Zeichenketten werden in Java mit einem „+“ zusammengefügt!*

Referenzvariablen

Wie weiter oben bereits erwähnt, gibt es neben den einfachen Datentypen die Referenztypen. Die Variablen dieser Referenztypen nennt man auch „Referenzvariablen“. Sie ermöglichen den Zu-

Schlüsselwort	Länge in Byte	Wertebereich
boolean	1	true oder false
char	2	16-Bit Unicode Zeichen (0x0000...0xffff)
byte	1	-2 ⁷ bis 2 ⁷ - 1 (-128...127)
short	2	-2 ¹⁵ bis 2 ¹⁵ - 1 (-32768...32767)
int	4	-2 ³¹ bis 2 ³¹ - 1 (-2147483648...2147483647)
long	8	-2 ⁶³ bis 2 ⁶³ - 1 (-9223372036854775808...9223372036854775807)
float	4	1,40239846E-45f...3,40282347E+38f
double	8	4,94065645841246544E-324...1,79769131486231570E+308

Tabelle: Die einfachen Java-Datentypen und deren Wertebereiche.

z.B. verschiedene Integer, Gleitkommatypen, ... und sogenannte **Referenztypen** (für Referenzvariablen) wie z.B. eine Klasse. Die elementaren Datentypen haben in Java eine feste Größe und einen genau vorgeschriebenen Wertebereich. Ein weiteres Merkmal der einfachen Datentypen ist, dass sie im Gegensatz zu den Referenztypen nicht als Objekte verwaltet werden. Da jede Variable einen festen Datentypen hat und

Deklaration von Variablen einfacher Datentypen im Programm

In den Variablen lassen sich Daten speichern, mit dem das Programm dann zur Laufzeit arbeiten kann. Bevor die Variablen verwendet werden können, müssen sie vorher deklariert werden. Man kann auch sagen, dass die Variablen definiert werden. In Java geschieht das in folgender Weise:

chen – zwei Hochkommata (Gänsefüßchen). Im Kasten 1 sind verschiedenen Beispiele für Variablendeklarationen als korrekte Java-Anweisungen aufgeführt.

Es gilt: Alle Anweisungen müssen in Java mit einem Semikolon abgeschlossen werden!

Erweitern Sie nun in unserem Java-Programm aus Teil 1 die Methode „main“ und fügen die Anweisungen aus Kasten 1 direkt nach der geöffneten Klammer „{“

ANZEIGE

64 Bit ... mit XML-Modul

LABOR EXPRESS

Die Abrechnungs-Software für das Dental-Labor

Der Service stimmt!

CD anfordern!
kostenlos testen

Jetzt informieren:
Telefon: 02744 / 920837

BEYCODENT

Wolfsweg 34 • D-57562 Herdorf

www.dental-laborsoftware.de

Kasten 1: Deklaration von Variablen.

```

/*
 * Methode zum Starten eines Java-Programms.
 */
public static void main(String[] args)
{
    // Einige Variablen deklarieren
    String  meinName;      // Mein Name
    int     meinAlter;     // Mein Alter
    double  auszahlung;    // Auszahlung
    char    geschlecht;    // Geschlecht ('f' oder 'm')
    boolean istZahntechniker; // Ist Zahntechniker (true oder false)

    // Mehrere Variablen auf in einem Schritt deklarieren
    int zahl3, zahl4, zahl5;

    // Variablendeklaration mit Wertinitialisierung
    int zahl6 = 10;

    // Ein Zeichen und dann eine Zeichenkette deklarieren
    char  einZeichen = 'a';
    String zeichenkette1 = "Ich";
    String zeichenkette2 = "bin", zeichenkette3 = "Zahntechniker!";

    // Ausgabe von drei zusammengeführten Zeichenketten mit Leerzeichen
    System.out.println(zeichenkette1 + " " + zeichenkette2 + " " + zeichenkette3);

    HelloWorld helloworld = new HelloWorld(); // Ein Objekt erzeugen
    helloworld.showText(); // Ruft die Methode showText() auf
}

```

Kasten 2: Verschiedene Variablendeklarationen in der Methode „main“.

griff auf Objekte, die im sogenannten **Heap-Speicher** abgelegt sind. Der Heap ist ein Speicherbereich speziell für dynamische Variablen, den sogenannten Objekten (siehe nächstes Kapitel). Die

Objekte

Mit dem **new-Operator** können in Java Objekte erzeugt und im Heap-Speicher temporär gespeichert werden. Ein Objekt stellt

```
// Datei: HelloWorld.java zum Ausgeben des Textes Hello World!
public class HelloWorld
{
    private int zahl1; // Eine Instanzvariable
    private static int zahl2; // Eine Klassenvariable (wird nur einmal angelegt)

    /*
     * Methode zum Starten eines Java-Programms.
     */
    public static void main(String[] args)
    {
        int zahl = 10; // lokale Variable
        String kurzerText = "Hello World!"; // lokale Variable

        HelloWorld helloworld = new HelloWorld();
        helloworld.showZahl(zahl); // Variable zahl wird der Methode übergeben
        helloworld.showText(kurzerText); // Variable text1 wird übergeben
    }

    /*
     * Methode zum Anzeigen eines Textes.
     */
    private void showText(String text)
    {
        System.out.println(text); // Gibt einen Text in der Konsole aus
    }

    /*
     * Methode zum Anzeigen eines Wertes.
     * Der Übergabeparameter ist int zahl.
     */
    private void showZahl(int zahl)
    {
        System.out.println(zahl); // Gibt 10 in der Konsole aus
    }
}
```

Kasten 3: Die unterschiedlichen Arten von Variablen in Java.

Referenzvariablen können entweder auf Objekte oder auf nichts (null) zeigen. Wenn eine Referenzvariable auf **null** zeigt, dann ist das eine **null-Referenz**. Die Referenz null ist in Java eine vordefinierte Referenz. Man kann in Java auch jeder Referenzvariablen die Referenz null zuweisen. *Es gilt: Ist ein Objekt vom Typ „Klassenname“, so wird auch eine Referenzvariable vom Typ „Klassenname“ benötigt, soll die Referenzvariable auf dieses Objekt zeigen. Das heißt, dass der Datentyp der Referenzvariablen und des Objektes, auf das die Referenzvariable zeigt, identisch sein müssen.*

Definition einer Referenzvariablen in Java:

Klassenname referenzName;

Beispiele:
 HelloWorld helloWorld;
 oder
 KlasseA klasseA, klasseB, klasseC; // Drei Referenzvariablen in einem Schritt definieren

Es gilt: Eine Referenzvariable ist eine Verknüpfung zu einem Objekt im Heap-Speicher. Sie bekommt einen Namen wie eine Variable eines einfachen Datentyps. Referenzvariablen gibt es nur für Objekte und nicht für die Variablen einfacher Datentypen.

Die unterschiedlichen Arten von Variablen in Java

In Java kennt man insgesamt drei unterschiedliche Arten von Variablen: **Klassenvariablen**, **Instanzvariablen** und **lokale Variablen**. Eine Klassenvariable wird

für eine bestimmte Klasse nur einmal im Programm angelegt. Instanzvariablen existieren in einem Programm für jede erzeugte Instanz einer Klasse. Die lokalen Variablen existieren nur in einer Methode und sind außerhalb der Methode nicht bekannt. **Übergabeparameter** kann man als spezielle lokale Variablen ansehen, die Methoden, Konstruktoren (kommt in Teil 4) und sogenannte Ausnahmebehandlungen (catch-Konstrukte) als Variablen übergeben werden können. Diese sind dann dort lokale Variablen. Wir ändern nochmals unser Programm, wie im Kasten 3 zu sehen ist.

Es gilt: Eine Klassenvariable, die auch bei mehreren Objekt-Instanzen der Klasse immer nur einmal existiert, muss mit static definiert werden.

Fazit

Wir haben in diesem Teil Variablen mit einfachen Datentypen und Referenzvariablen kennen-

gelernt. Wir wissen nun, dass Objekte dynamische Variablen sind, die auf dem Heap abgelegt werden und mittels Referenzvariablen angesprochen werden. Außerdem haben wir die unterschiedlichen Arten von Variablen und ihre Sichtbarkeit kennengelernt. Klassen, wie z.B. unsere Klasse „HelloWorld“, sind eine Art Bauplan und beschreiben die Eigenschaften. Die erzeugten Objekte besitzen dann diese Eigenschaften der Klasse. Die Methoden einer Klasse stellen die Kommunikationsschnittstelle nach außen dar und können Übergabeparameter mit übergeben bekommen.

Ausblick zum Teil 4 der Serie

Im nächsten Teil 4 lernen wir den Objekttyp „Array“ kennen, beschäftigen uns mit Ausdrücken und Operatoren, bedingte Anweisungen, Kontrollstrukturen und vertiefen das Thema „Objekt orientiertes Programmieren“. ZT

ZT Autor



Thomas Burgard entwickelt Applikationssoftware und professionelle Internetauftritte für Unternehmen.

ZT Adresse

Thomas Burgard Dipl.-Ing. (FH)
 Softwareentwicklung & Webdesign
 Bavariastraße 18b
 80336 München
 Tel.: 089 540707-10
 Fax: 089 540707-11
 info@burgardsoft.de
 www.burgardsoft.de

ANZEIGE

Eierschalenprovisorium - Gefräst mit dem CAD/CAM System 5-TEC

EIERSCHALENPROVISORIUM

SOFTWARE-MODUL CAD/CAM

CAD/CAM System 5-TEC:
 Computergesteuertes
 5+1-Achsen-Simultan-Fräsgesät M5
 und vollautomatischer
 optischer Scanner S600 ARTI

Vielfältiges Indikationsspektrum:

Nur mit 5+1 Achsen fräsbar

Käppchen	Kronen in Prettau®	Brücken in Prettau®	Inlays	Onlays	Eierschalenprovisorium	Aufbiss-schienen	Teleskop-kronen	Abutment mit Klebebasis	Vollzirkonabutment	Verschraubung	Geschlebe	Steg	Brücken mit Unterschnitten u. Übergerenzen	Totalprothese	Metallgerüst	Airbrush	Schrift-laser	Mass-Schleif-funktion
----------	--------------------	---------------------	--------	--------	------------------------	------------------	-----------------	-------------------------	--------------------	---------------	-----------	------	--	---------------	--------------	----------	---------------	-----------------------

Zirkonzahn Worldwide - Südtirol - T +39 0474 066 680 - www.zirkonzahn.com - info@zirkonzahn.com