

Einstieg in die Programmierung von Computern – Teil II

Mit der neuen mehrteiligen Serie „Einstieg in die Programmierung von Computern“ soll ein fundamentales Verständnis für die Funktionsweise der computerisierten Welt geschaffen werden. Eine moderne und leicht verständliche Programmiersprache wie z. B. „Java“ zu erlernen, ist dafür bestens geeignet. Im zweiten Teil geht es nun konkret in die Java-Programmierung.

Bevor wir in die Java-Grundlagen so richtig einsteigen, sollten zuerst die Voraussetzungen für die Programmierung und den Start eines Java-Programms auf dem Computer geschaffen werden. Ich werde mich in diesem Einsteigerseminar lediglich auf das Betriebssystem Windows beschränken. Folgende Software (alles Open Source und somit kostenfrei) wird benötigt:

Damit ist die Java Standard Edition Java SE auf dem Rechner installiert.

Getestet werden kann die Installation und Einstellung der Umgebungsvariablen folgendermaßen:

- Starten der „Windows Eingabeaufforderung“, auch Windows-Konsole genannt. Zu finden ist Eingabeaufforderung unter: **Start → Alle Programme**

installiert. Nun kann mit der Java-Programmierung und einem ersten kleinen Beispiel begonnen werden.

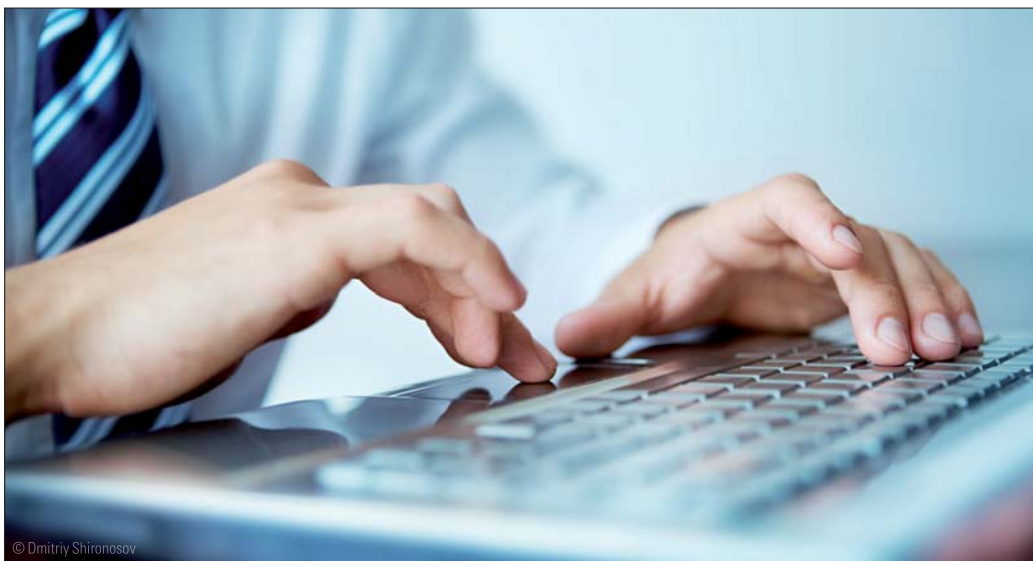
Das erste Java-Programm

Mit dem folgenden ersten Java-Programm wird kennengelernt, wie ein startfähiges Java-Programm aussieht, wie und wo

- **Tipps für Notepad++:** Im Hauptmenü von Notepad++ den Menüpunkt **Erweiterungen** öffnen und den Untermenüpunkt **Explorer** auswählen. Im Notepad++ auf der linken Seite ist nun ein sogenanntes „Explorer-Fenster“ zu sehen, indem bequem die Datei „HelloWorld.java“ im gespeicherten Verzeichnis auf dem Laufwerk ausgewählt werden kann. Die

abhängigen Maschinen- bzw. Bytecode und kann direkt ausgeführt werden.

5. Nun kann das erste selbst geschriebene Java-Programm ausgeführt werden: Starten der „Windows Eingabeaufforderung“ und Kommando `java HelloWorld` eingeben. Die Windows-Konsole zeigt wie gewünscht die Zeichenkette „Hello World!“ an!



Analyse der Datei HelloWorld.java

Im Folgenden soll nun die Source-Datei HelloWorld.java genau analysiert werden.

Die Zeile `// Datei: HelloWorld.java zum Ausgeben des Textes Hello World!` stellt in Java eine Kommentarzeile dar und wird vom Java-Compiler übergangen. Der Programmierer kann sich damit Notizen, Beschreibungen und Merker an gewünschten Programmstellen in der Java-Quell-Datei machen.

Info: Man darf Kommentare in Java an jeder gewünschten Stelle in der Java-Quell-Datei schreiben.

Beispiele für Java-Kommentare

1. Kommentar in einer Zeile
`// Das ist eine Kommentarzeile!`
oder
`/* Das ist eine Kommentarzeile! */`
2. Kommentar über mehrere Zeilen
`/*
 * Methode zum Starten eines
 * Java-Programms
 *
 */`
3. Kommentar am Ende einer Java-Anweisung

`helloworld.showText(); // Ruft die Methode showText() auf.`

Die Zeile `public class HelloWorld` deklariert eine „öffentliche“ (kann von außerhalb instanziiert werden) Java-Klasse mit dem Namen „HelloWorld“, die eine beginnende und am Ende eine geschlossene geschweifte Klammer stehen hat. In die beiden geschweiften Klammern werden die Eigenschaften und Methoden der Klasse geschrieben. Auf Java-Klassen und was dabei zu beachten gilt, wird im kommenden Teil III noch genauer eingegangen.

Die Funktion `main` ist eine statische Funktion und muss in einem Java-Programm insgesamt einmal vorhanden sein. Sie stellt den Einstiegspunkt in die Ausführung einer Java-Anwendung

1. Herunterladen der Java Standard Edition Java SE

Es wird die Java-Plattform JDK (Java Development Kit) mit integrierter JRE zum Programmieren in Java und zum Starten und Ablauf des Java-Programms benötigt (Download unter www.oracle.com/technetwork/java/javase/downloads/, Aktuelle Datei: `jdk-7u9-windows-i586.exe`).

2. Installation der Java Standard Edition Java SE

Nach erfolgreichem Herunterladen wird mit Doppelklick auf die Datei die Installation der Java-Plattform JDK gestartet. Nach erfolgreicher Installation muss noch die Java-Umgebungsvariable eingestellt werden. Unter Windows geht man dafür folgendermaßen vor:

- **Schritt 1:** Aufruf von **Start → Systemsteuerung → System und Sicherheit → System → Erweiterte Systemeinstellungen**
- **Schritt 2:** Klick auf **Erweitert → Umgebungsvariablen**
- **Schritt 3:** Auswahl der „Path-Variable“ im unteren Kasten und Klick auf Bearbeiten.
- **Schritt 4:** Mit dem Cursor „ganz nach rechts“ zum Ende des Textes gehen und ein Semikolon „;“ eingeben.

Nach dem Semikolon nun den Pfad des Java bin Ordners eingeben. Auf meinem Rechner ist das `C:\Program Files\Java\jdk1.7.0_09\bin`.

Besonders für Anfänger ist zusätzlich die Installation der „Demos und Samples“ (Demo-Programme und Beispiele) zu empfehlen.

→ **Zubehör Eingabeaufforderung**

- Eingabe des Kommandos: `java -version`

Zu sehen sollte dann folgende Ausgabe sein:

`java version „1.7.0_09“
Java™ SE Runtime Environment (build 1.7.0_09-b05)
Java HotSpot™ Client VM (build 23.5-b02, mixed mode, sharing)`

3. Installation eines Texteditors zum Schreiben von Java-Programmen

Was weiterhin zum Programmieren von Java-Programmen benötigt wird, ist ein spezieller Texteditor für Programmiersprachen, der die Programmiersprachen-Syntax auch optisch in verschiedenen Farben anzeigt. Hierfür kann man den ebenfalls kostenfreien Open Source Texteditor **Notepad++** verwenden, der für die Java-Programmierung optimal geeignet ist (Download: www.notepad-plus-plus.org/download/v6.2.html, Datei: `Notepad++-Installer für Windows`). Nach erfolgreichem Herunterladen wird mit einem Doppelklick auf die Datei die Installation von Notepad++ gestartet. Es wird automatisch die deutsche Sprache ausgewählt und

ein Java-Programm gespeichert wird, wie es „kompiliert“ wird und zum Schluss natürlich, wie es gestartet wird.

Gestartet wird mit dem berühmten „Hello World“-Programm. Es hat nur wenige Zeilen Code, aber es stellt ein vollständig kompilier- und startfähiges Java-Programm dar. Der Java-Code ist im Kasten zu sehen.

Vorgehensweise

1. **Notepad++** starten.
2. Den im Kasten angezeigten **Java-Code** vollständig im großen weißen editierbaren Bereich eingeben (Achtung: Java unterscheidet Groß- und Kleinschreibung)
3. Datei unter dem Namen `HelloWorld.java` abspeichern.
 - **Vorgehen:** Neues Verzeichnis mit dem Namen `Javaseminar` (z. B. `C:\Javaseminar`) im Windows Explorer anlegen. Dann im Notepad++ im Hauptmenü auf **Datei → Speichern unter ...** das vorher angelegte Verzeichnis auswählen und den Dateinamen „HelloWorld.java“ eingeben. Dann im Dialogfenster auf **Speichern** klicken. Jetzt sollte die Datei unter `C:\Javaseminar\HelloWorld.java` auf dem Rechner gespeichert sein.

ANZEIGE

www.zwp-online.info

FINDEN STATT SUCHEN.

www.zwp-online.at

Auch in Österreich und der Schweiz

www.zwp-online.ch

dar und muss die Signatur `public static void main(String[] args)` besitzen.

- **public**, da die Methode „main“ von außerhalb der Klasse aufgerufen wird.
- **static**, da sie aufgerufen wird, ohne vorher ein Objekt der Klasse „HelloWorld“ gebildet bzw. instanziiert zu haben.
- **void**, da die Methode „main“ keinen Rückgabewert besitzt.
- der Parameter `String[] args`, ist ein sogenannter Array, das die vom Aufruf entgegengenommenen Kommandozeilenparameter enthält. Ein Array kann man sich als eine Art Datenbehälter im Hauptspeicher des Rechners vorstellen (der Java-Datentyp „Array“ wird später noch genauer erklärt).

Jede Methode in Java hat einen Namen und zwei geschweifte Klammern „{ }“, in denen die Anweisungen der Methode geschrieben werden. Die beiden Anweisungen in der Methode `main` sind:

```

HelloWorld helloworld = new HelloWorld();
helloworld.showText(); // Ruft die Methode showText() auf.
    
```

Die Anweisung `HelloWorld helloworld = new HelloWorld();` bewirkt folgendes: Der `new`-Operator dient zum Erzeugen

eines neuen Exemplars einer Klasse. Man kann auch sagen, dass mit `new` eine neue Instanz der Klasse erzeugt wird, im Beispiel von der Klasse „HelloWorld“. Durch seine Verwendung wird ein neuer Speicherbe-

reicht belegt, der das neue Objekt dann aufnimmt. Die neue Instanz der Klasse „HelloWorld“ wird in der Variablen „helloworld“ gespeichert. Mit dieser erzeugten Variable wird die neue Instanz in der Methode „main“ verwendet. Konkret wird mit `helloworld.showText();` die Me-

thode `showText()` der Klasse „HelloWorld“ aufgerufen. Die Variable `helloworld` beinhaltet die erzeugte Instanz der Klasse „HelloWorld“ und kann dazu verwendet werden, Eigenschaften und Methoden einer Klasse privaten Methode „showText“ (siehe zweite Anweisung innerhalb von `main`) problemlos. Mit `private` schützt man sozusagen Eigenschaften und Methoden einer Klasse. Von außen kann man auf diese nicht ohne Weiteres zugreifen. Mit der Anweisung `System.out.println("Hello World!");` wird der Text „Hello World!“ in der Windows-Konsole ausgegeben.

Fazit

Nun wurden die Voraussetzungen für die Java-Programmierung geschaffen und das erste kleine Java-Programm mit dem Namen „HelloWorld“ mit einem speziellen Texteditor „Notepad++“ entwickelt, anschließend mit dem Java-Compiler kompiliert und das Programm zur Ausführung gebracht. Es wurde gelernt, wie die Befehle für die Kompilierung und Start eines Java-Programms heißen und wo das Programm unter Windows aufgerufen werden kann. Weiterhin, dass jedes Java-Programm mindestens eine Klasse besitzen und es eine Methode „main“ als Startpunkt für die Java-Anwendung geben muss. Außerdem wurden die ersten Java-Syntax-Regeln und Anweisungen zum Verständnis für das kleine Programm „HelloWorld“ kennengelernt.

Ausblick auf Teil III der Serie

Teil III beschäftigt sich mehr mit den grundlegenden Java-Datentypen, was „objektorientierte Softwareentwicklung“ bedeutet und wie sie in der Java-Programmiersprache umgesetzt ist. **ZT**

```

//Datei: HelloWorld.java zum Ausgeben des Textes Hello World!
public class HelloWorld
{
    /*
     * Methode zum Starten eines Java-Programms.
     */
    public static void main(String[] args)
    {
        HelloWorld helloworld = new HelloWorld();
        helloworld.showText(); // Ruft die Methode showText() auf.
    }

    private void showText()
    {
        System.out.println("Hello World!");
    }
}
    
```

thode `showText()` der Klasse „HelloWorld“ aufgerufen. Die Variable `helloworld` beinhaltet die erzeugte Instanz der Klasse „HelloWorld“ und kann dazu verwendet werden, Eigenschaften und Methoden einer Klasse

privaten Methode „showText“ (siehe zweite Anweisung innerhalb von `main`) problemlos. Mit `private` schützt man sozusagen Eigenschaften und Methoden einer Klasse. Von außen kann man auf diese nicht ohne Weiteres zugreifen. Mit der Anweisung `System.out.println("Hello World!");` wird der Text „Hello World!“ in der Windows-Konsole ausgegeben.

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

...mehr Ideen - weniger Aufwand

microtec • Inn. M. Nolte
 Rohrstr. 14 • 58093 Hagen
 Tel.: ++49 (0) 2331 8081-0 • Fax: ++49 (0) 2331 8081-18
 info@microtec-dental.de • www.microtec-dental.de

Machen Sie friktionslose Edelmetall-Teleskopkronen wieder fit!

Bei NEM-Kronen empfehlen wir unser Telerep oder Frisoff-System

bohren

Gewinde schneiden

eindrehen, einstellen

abtrennen... fertig

Bitte kreuzen Sie an:

Bitte senden Sie mir ein kostenloses Funktionsmuster*
*Nur einmal pro Labor/Praxis.

Bitte senden Sie mir das Quick-rep Starter-Set zum Sonderpreis von 125,00 €**
Inhalt des Starter-Sets: 14 Friktionselemente + Einbauwerkzeuge
 **Nur einmal pro Labor/Praxis. / zzgl. ges. MwSt. / versandkostenfrei
 Der Sonderpreis gilt nur bei Bestellung innerhalb Deutschlands.

per Fax an 02331 / 8081 - 18

Kostenlose Hotline (0800) 880 4 880

ANZEIGE