Java Tutorial

Part 2 - Elementare Java Grundlagen

1. Vorraussetzungen

Im folgenden setze ich voraus, dass die Grundlagen aus Part 1 beherrscht werden, ein Grundverständnis für logische Operationen vorhanden ist und die Mathematik der Mittelstufe beherrscht wird.

Zunächst installieren Sie bitte das JDK (Java Development Environment). Den Download finden sie auf der Seite von Oracle.

Um nun die Java Komponenten leichter erreichen zu können sollten sie die PATH Variable Ihres Systems setzen, falls das noch nicht bereits getan wurde. Informationen dazu finden Sie hier.

Optional können Sie bereits die Eclipse IDE oder eine andere Java IDE installieren. Ich empfehle Ihnen aber die Nutzung der Eclipse IDE weil ich diese auch in diesem Tutorial benutze.

2. Nun geht's los…

2.1 Allgemeine Grundlagen

In ihrer Kommandozeile sollte nun falls sie nur „java“ oder „javac“ eingeben eine Hilfeausgabe erscheinen, ist dies nicht der Fall, wurde die PATH Variable ihres Systems nicht richtig gesetzt. Erstellen Sie nun bitte eine neue Datei mit dem Namen „helloworld.java“. Der Inhalt der Datei ist folgender:

```
public class Helloworld {

	public static void main(String[] args) {
		System.out.println("Hello world!");
	}

}
```

Gehen Sie nun mithilfe von „cd“ zum Ordner, in dem die soeben erstellte Datei liegt. Als nächstes führen Sie folgenden Befehl aus: „javac helloworld.java“ und daraufhin, sofern kein Fehler erscheint „java helloworld“. Beim ausführen von javac wurde eine Datei namens helloworld.class erstellt, das ist eine kompilierte Klassendatei im Bytecode, die zur Ausführung der Klasse nötig ist. Mit java führen wir nun die Klasse aus.

2.2 Datentypen

Da Java eine streng typisierte Programmiersprache ist, sollten wir uns ganz genau klar machen welche Datentypen uns als „Werkzeug“ direkt von der Sprache mitgegeben werden, was diese Datentypen auf sich haben und welche weiteren Möglichkeiten wir haben.

2.4. Primitive Datentypen

Java kennt genau acht primitive Datentypen, zu diesen gehören unter anderem die Ganzzahltypen **byte**, **short**, **int**, **long**, die Fließkommazahltypen **float** und **double**, der Zeichentyp **char** und der Wahrheitswertetyp **boolean**.
Einige Typen sind Ihnen eventuell aus anderen Programmiersprachen wie C++ bekannt, doch eventuell haben diese dort einige Eigenheiten, die sie in Java nicht aufweisen.

Angefangen bei boolean, booleans fassen die Werte „true“ und „false“ und werden im Programm durch ein Bit dargestellt, dass jeweils 1 (= true) oder 0 (= false) ist.
Wichtig: In Java können Sie, im Gegensatz zu z.B. C++, für true/false nicht 1/0 verwenden. Wie gesagt, Java ist dort etwas eigen und extrem stark typisiert.

Weiter geht es nun bei den Ganzzahltypen. Dazu am besten eine kleine Tabelle:

Datentyp	Größe in Byte	kleinste Zahl	größte Zahl
byte	1	-128	+127
short	2	-32.768	+32.767
int	4	-2.147.483.648	+2.147.483.647
long	8	-9.223.372.036.854.775.808	+9.223.372.036.854.775.807

Wie man sieht, hat man in Java eine große Auswahl, wie man seine ganzen Zahlen darstellt. Hier möchte ich aber anmerken, dass man immer einen möglichst passenden Datentyp wählt.
Einerseits sollte man mit dem vorhandenen Speicher sparsam umgehen, auch wenn dieser in der heutigen Zeit extrem günstig zu bekommen ist, andererseits sollte man sich bewusst sein, dass ein zu klein gewählter Datentyp Konsequenzen nach sich zieht, wenn man die Grenzen über- bzw. unterschreitet.
Überschreitet man das Maximum um z.B. 1, so landet man direkt bei der kleinsten Zahl. Unterschreitet man das Minimum um z.B. 1, so landet man bei der größten Zahl. Man kann sich das ganze als Rad vorstellen. Es gibt kein direktes Ende bei den Datentypen, wenn man am einen Ende rauskommt, kommt man am anderen Ende rein. Das hat den schönen Aspekt, dass das Programm bei einem Over- bzw. Underflow nicht terminiert, andererseits ist genau dieser positive Aspekt auch, bei Nichtbeachtung dieser Tatsache, ein massives Problem, dass später bei Schleifendurchläufen oder Berechnungen zu fatalen Problemen führen kann.

Nun folgen auch schon die Fließkommazahlen. Float wird durch 4 Byte dargestellt, Double durch 8 Byte. Die Datentypen wurden gemäß IEEE 754 realisiert. Auf der verlinkten Seite finden Sie u.a. auch die Grenzen des Datentyps, ich möchte nicht weiter darauf eingehen, da dies einen tieferen Griff in die Informatik nach sich ziehen würde, der den Rahmen dieses Tutorials sprengen würde.

Zu guter Letzt haben wir noch den Zeichentyp, dieser wird mit char bezeichnet und wird mit 2 Byte dargestellt. Diese zwei Byte werden im Einerkomplement dargestellt, das heißt im Gegensatz zum short belaufen sich die Grenzen der Zahl von 0 bis +65.535. Ein Zeichen wird in char durch die Zahl die ihr im Unicode (UTF-16) zugewiesen wurde repräsentiert. Das heißt natürlich auch, dass wir einen Char zu einem int oder größer umwandeln können. Das ganze funktioniert in folgender Reihenfolge:
byte -> short -> int -> long -> float -> double
bzw. char -> int -> long -> float -> double
Diese Umwandlungen funktionieren in Java problemlos, sie werden implizit vom Compiler durchgeführt, d.h. wir müssen im Code nicht extra darauf hinweisen, es wird automatisch gemacht.