

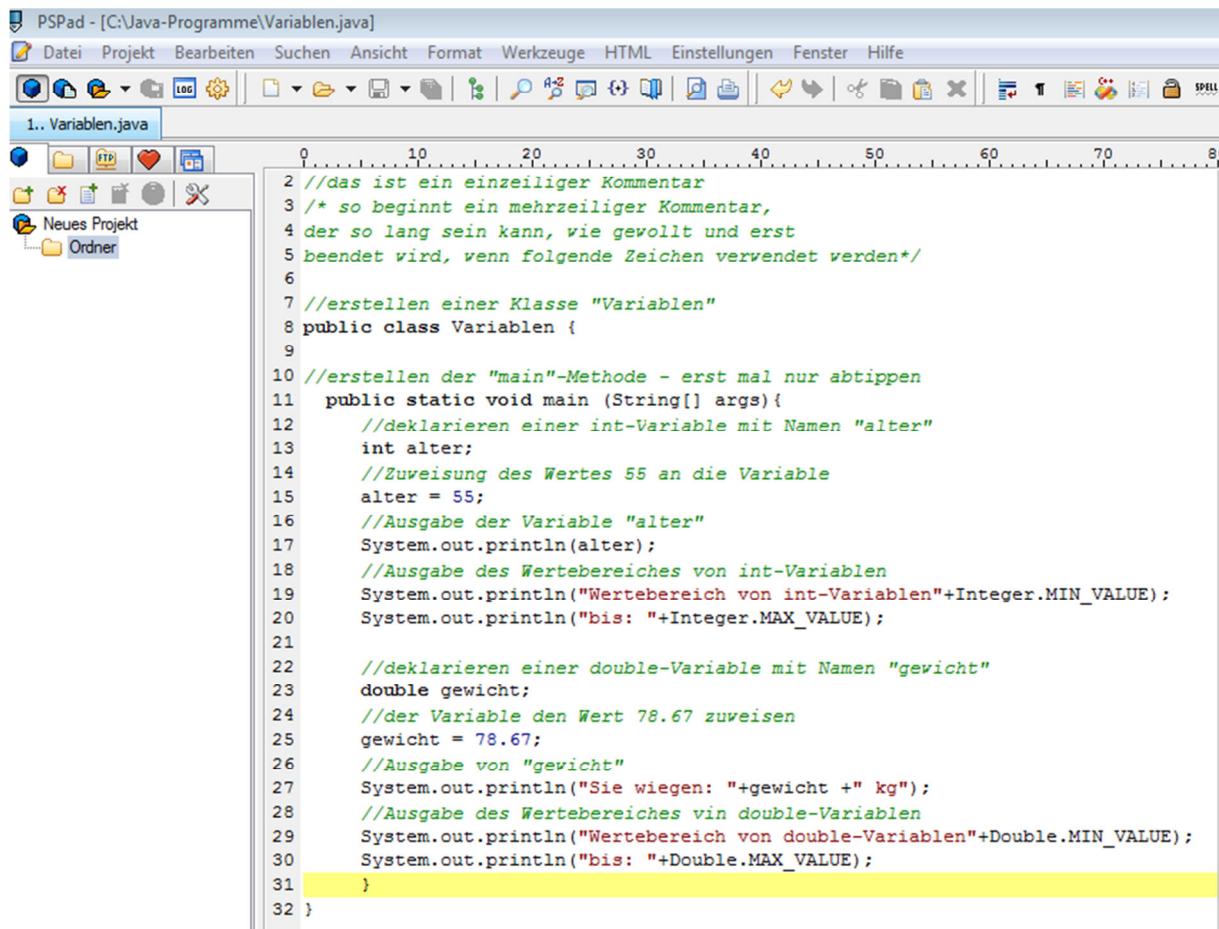
Das erste Programm-Teil 2

Programmierer legen großen Wert auf Kürze und sind ständig am Sparen. Ob Zeilen im Quellcode oder Speicherplatz, alles halten sie so kurz wie nur möglich, als ob das was kosten würde. So ist es möglich, Deklaration und Wertzuweisung in eine Zeile zu packen, in unserem Fall also:

```
int alter = 55;
```

Jeder muss für sich selbst entscheiden, wie er arbeitet, bei der späteren Programmierarbeit kommt diese Methode sowieso selten vor, denn eigentlich ist unsere Variable alles andere als variabel.

Zur Verarbeitung von Fließkomma-Zahlen benutzen wir den Variablentyp der „double-Variable“. Diese belegt 64 Bit unseres Arbeitsspeichers, den Wertebereich lassen Sie sich bitte im Listing-2 ausgeben, das ist mir zu viel Getippe und es schaut sich eh keiner so genau an.



The screenshot shows the PSPad text editor interface with a Java file named '1.. Variablen.java' open. The code demonstrates various Java variable declarations and assignments. The code is as follows:

```
1 //das ist ein einzeiliger Kommentar
2 /* so beginnt ein mehrzeiliger Kommentar,
3  * der so lang sein kann, wie gewollt und erst
4  * beendet wird, wenn folgende Zeichen verwendet werden */
5
6
7 //erstellen einer Klasse "Variablen"
8 public class Variablen {
9
10 //erstellen der "main"-Methode - erst mal nur abtippen
11 public static void main (String[] args){
12     //deklarieren einer int-Variable mit Namen "alter"
13     int alter;
14     //Zuweisung des Wertes 55 an die Variable
15     alter = 55;
16     //Ausgabe der Variable "alter"
17     System.out.println(alter);
18     //Ausgabe des Wertebereiches von int-Variablen
19     System.out.println("Wertebereich von int-Variablen"+Integer.MIN_VALUE);
20     System.out.println("bis: "+Integer.MAX_VALUE);
21
22     //deklarieren einer double-Variable mit Namen "gewicht"
23     double gewicht;
24     //der Variable den Wert 78.67 zuweisen
25     gewicht = 78.67;
26     //Ausgabe von "gewicht"
27     System.out.println("Sie wiegen: "+gewicht +" kg");
28     //Ausgabe des Wertebereiches von double-Variablen
29     System.out.println("Wertebereich von double-Variablen"+Double.MIN_VALUE);
30     System.out.println("bis: "+Double.MAX_VALUE);
31 }
32 }
```

Listing 1-2

Zeile 23

Hier wird die Variable vom Typ Double Deklariert und „gewicht“ genannt.

Zeile 25

Der Variable „gewicht“ wird der Wert 78.67 zugewiesen.

Zeile 27

Ausgabe der Variable „gewicht“.

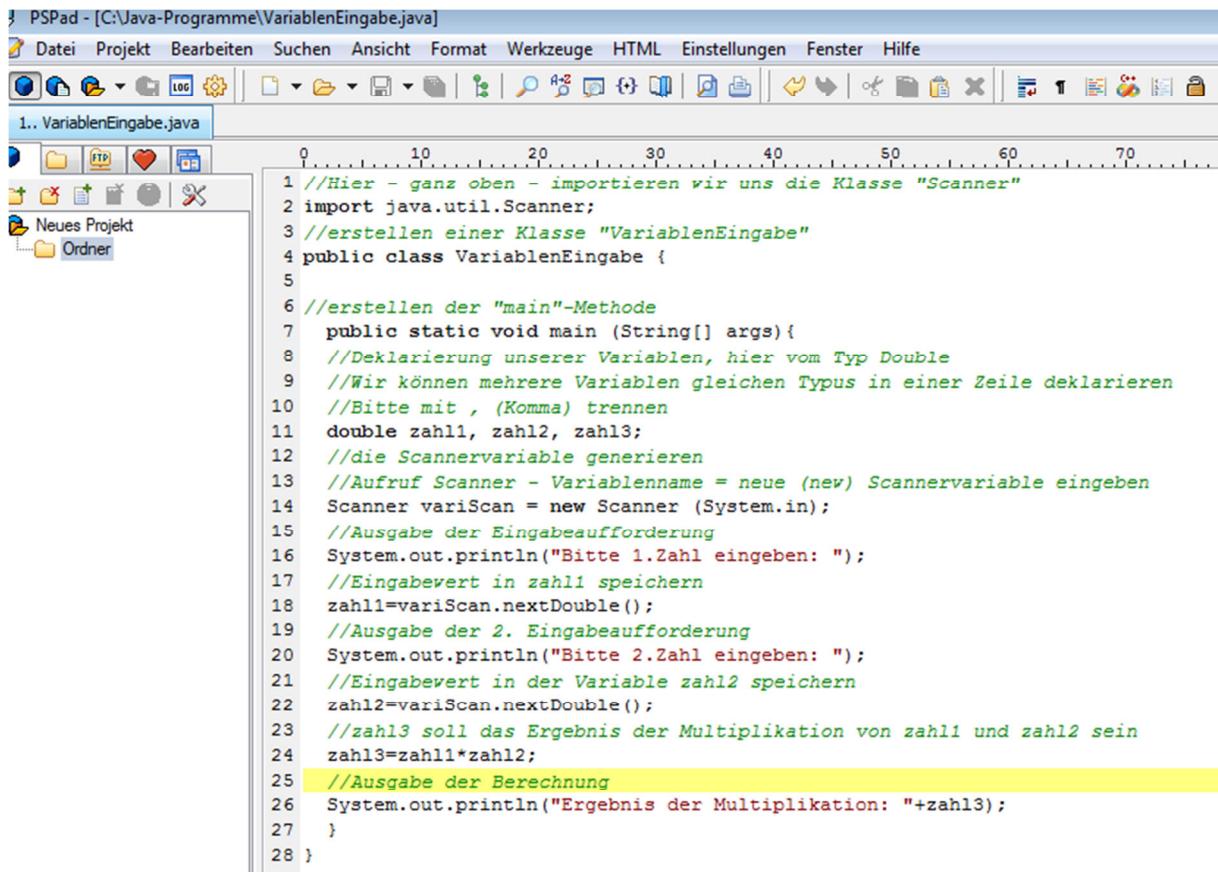
Zeilen 29 und 30

Ausgabe des Wertebereiches der Double-Variablen.

Da Sie auch hier mit der Höhe Ihrer Gewichtsangabe mogeln dürfen, beachten Sie bitte, dass die Abtrennung des Vorkomma-Teiles vom Nachkomma-Teil durch den Punkt erfolgt, nicht durch das Komma! Wird bei der Eingabe der Dezimalpunkt weggelassen, wird die Variable trotzdem mit der Nachkommastelle dargestellt, in dem Fall mit „.0“.

An dieser Stelle, noch bevor wir die anderen Variablen-Typen behandeln, greife ich im Thema nach vorn, damit das Arbeiten in Java mehr Spaß macht. Es geht um die Eingaben. Meiner Meinung nach, ist eine Variable erste dann eine, wenn ihr Wert auch variabel ist, also durch Eingabe verschiedener Zahlen und Zeichen Bewegung in die Programme kommen. Und letztlich ist ein Programm erst eins, wenn der Prozess aus Eingabe, Verarbeitung und Ausgabe nachvollziehbar wird und den eigentlichen Sinn sich dadurch erschließt. Und wie gesagt: Es macht auch mehr Spaß.

Java bietet uns von Haus aus Klassen an (Klassen behandeln wir dann später), die wir für unsere Zwecke benutzen können. Manche davon werden automatisch geladen und laufen im Hintergrund bereits mit. Wir benutzten Sie bereits (z.B. System.out.println()), andere müssen geladen, bzw. importiert werden, damit sie uns zu Diensten sein können. Für die Eingaben benutzen wir die Klasse „Scanner“. Den unteren Quellcode bitte neu erstellen, die Datei unter dem Namen „VariablenEingabe“ speichern.



The screenshot shows the PSPad code editor with the following details:

- File Path:** C:\Java-Programme\VariablenEingabe.java
- Toolbar:** Datei, Projekt, Bearbeiten, Suchen, Ansicht, Format, Werkzeuge, HTML, Einstellungen, Fenster, Hilfe.
- Left Sidebar:** Shows a project tree with "Neues Projekt" and "Ordner".
- Code Area:** Displays the following Java code:

```
1 //Hier - ganz oben - importieren wir uns die Klasse "Scanner"
2 import java.util.Scanner;
3 //erstellen einer Klasse "VariablenEingabe"
4 public class VariablenEingabe {
5
6     //erstellen der "main"-Methode
7     public static void main (String[] args){
8         //Deklarierung unserer Variablen, hier vom Typ Double
9         //Wir können mehrere Variablen gleichen Typus in einer Zeile deklarieren
10        //Bitte mit , (Komma) trennen
11        double zahl1, zahl2, zahl3;
12        //die Scannervariable generieren
13        //Aufruf Scanner - Variablenname = neue (nev) Scannervariable eingeben
14        Scanner variScan = new Scanner (System.in);
15        //Ausgabe der Eingabeaufforderung
16        System.out.println("Bitte 1.Zahl eingeben: ");
17        //Eingabewert in zahl1 speichern
18        zahl1=variScan.nextDouble();
19        //Ausgabe der 2. Eingabeaufforderung
20        System.out.println("Bitte 2.Zahl eingeben: ");
21        //Eingabewert in der Variable zahl2 speichern
22        zahl2=variScan.nextDouble();
23        //zahl3 soll das Ergebnis der Multiplikation von zahl1 und zahl2 sein
24        zahl3=zahl1*zahl2;
25        //Ausgabe der Berechnung
26        System.out.println("Ergebnis der Multiplikation: "+zahl3);
27    }
28 }
```

Listing 2

Zeile 2

Import der Java-Klasse „Scanner“.

Zeile 4

Erstellen der Klasse „VariablenEingabe“.

Zeile 7

Erstellen unserer „main-Methode“; Beginn des Programms.

Zeile 11

Deklarieren von 3 Variablen vom Typ Double. Hier mal in einer Zeile, gleich hintereinander. Die einzelnen Variablen durch Komma trennen.

Zeile 14

Aufruf der Klasse Scanner, Deklarierung einer (Scanner)Variable mit dem Namen „variScan“. Dies soll eine neue Variable der Klasse Scanner sein, also new Scanner, und im Gegensatz zum bereits benutzten System.out erfolgt hier ein System.in – also eine Eingabe.

Zeile 16

Ist nichts anderes, als eine Eingabeaufforderung, man könnte sie weglassen, aber dann weiß man ja nicht, was der blinkende Cursor von einem erwartet.

Alle Eingaben werden mit „Enter“ abgeschlossen.

Zeile 18

Hier wird der Eingabewert in die Variable zahl1 geschrieben und gespeichert. „variScan“ kann neu beschrieben werden.

Zeile 20

Eingabe der 2. Zahl

Zeile 22

Eingabewert in zahl2 schreiben und speichern.

Zeile 24

Hier wird's aufregend, wir multiplizieren zahl1 mit zahl2, speichern das Ergebnis in zahl3.

Zeile 26

Ausgabe des Ergebnisses mit System.out.println.

Ende des Programmes – bitte speichern!

Nun Compilieren!

Eingabeaufforderung starten. (Es empfiehlt sich, die Editoren, die Eingabeaufforderung u.dgl. an der Task-Leiste anzuheften, von dort können sie mit einem einfachen Klick und ohne Fenster zu minimieren oder zu schließen jederzeit geöffnet und gestartet werden.

Unterverzeichnis „java-programme“ aufrufen, also:

C:\.....>cd\ „Enter“

C:\> cd java-programme „Enter“

C:\> javac VariablenEingabe.java „Enter“

(Daumen drücken!!!!) Wenn ohne Fehler:

C:\>java VariablenEingabe „Enter“

Ab geht's!

```
C:\ Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Marco Rauner>cd\
C:\>cd java-programme
C:\Java-Programme>javac VariablenEingabe.java
C:\Java-Programme>java VariablenEingabe
Bitte 1.Zahl eingeben:
12
Bitte 2.Zahl eingeben:
45
Ergebnis der Multiplikation: 540.0
C:\Java-Programme>
```