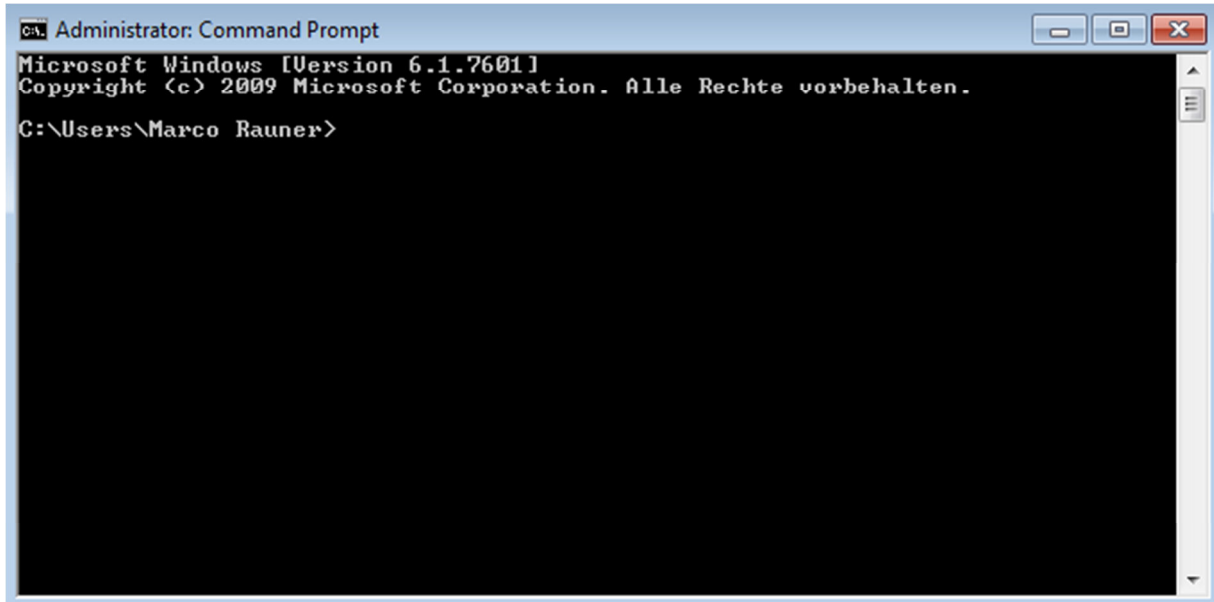


## Der Anfang

Laden wir uns den „Command Prompt“ auch „Eingabeaufforderung“ genannt auf den Desktop. Das sollte dann so aussehen:

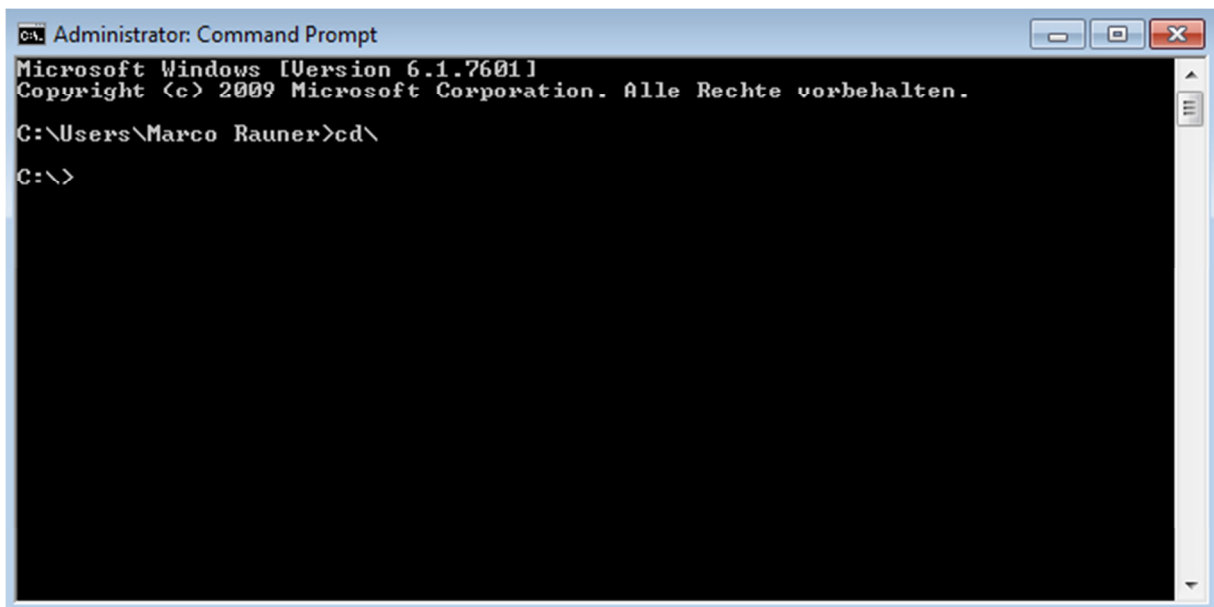


```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\Marco Rauner>
```

Erreichen können wir das, indem wir nach Anklicken des Windows-Button in das Eingabefeld „Programme/Dateien durchsuchen“ die Zeichenkette „cmd“ eingeben und das angebotene Programm bestätigen.

Natürlich mit dem entsprechenden Namen Ihrer Einstellungen hinter dem User. Nach Eingabe von `cd\`

(\ = alt gr+ß) und zur Bestätigung „Enter“, sollte es so aussehen :



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

Jetzt haben wir es geschafft! Wir sind ganz oben: C:\>\_ ist die Spitze unsere Computers, höher geht es nicht. C:\>\_ ist die oberste Ebene unserer Festplatte, alles andere liegt darunter. Aus Jux geben wir jetzt den Befehl „tree“ ein, drücken „Enter“ und haben was zum Gucken. Anhalten kann man das mit der Taste: „Pause“.

Das Schauspiel war aber beeindruckend, oder? Was haben wir gesehen? Der DOS-Befehl „tree“ (Baum) hat uns die Verzeichnisstruktur

(Ordner/Unterordner/Unterunterorder/Unterunterunterorder usw.)

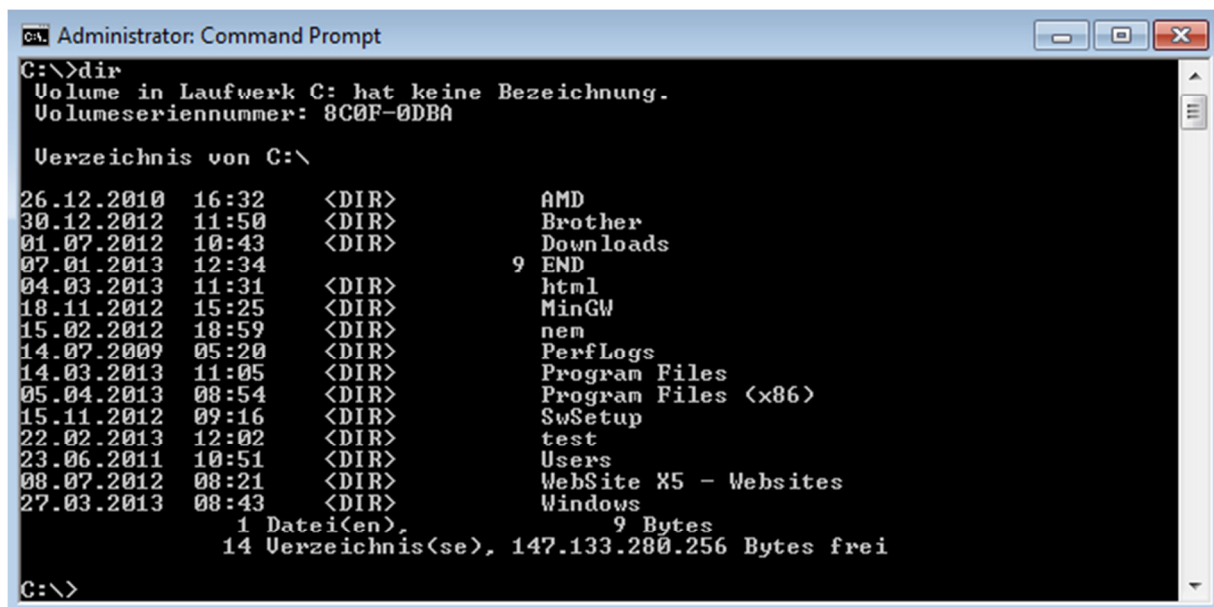
in Form eines „Baumes“ aufgelistet. Heute macht das der Explorer von Windows grafisch, aber am Ende dieses Kapitels sind wir in der Lage, Ordner und Unterordner anzulegen und zu löschen, und wir bewegen uns zwischen diesen Ordnern hin und her, ohne einen Maus und Rechts-Klick. Ja, das geht.

Und mit einem kühnen „cls“ reinigen wir unseren Bildschirm.

Keine Angst, wir werden jetzt nicht alle DOS-Befehle durchgehen und ausprobieren, aber wenn Euch die paar Anweisungen das Verständnis für die Datei- und Verzeichnis-Struktur im PC vertiefen hilft, wird uns das später helfen, mit Java-Programmen zu arbeiten und diese auszuführen.

Mit dem Befehl „dir“ werden uns alle Dateien und Unterverzeichnisse, die sich direkt unter C: befinden, angezeigt. Bitte tun Sie das.

Unterverzeichnisse (Ordner) sind mit <DIR> gekennzeichnet. Die Dateien stehen mit Größenangaben dazwischen, rechts lesen wir die Bezeichnungen.



```
Administrator: Command Prompt
C:\>dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volumenseriennummer: 8C0F-0DBA

Verzeichnis von C:\

26.12.2010  16:32    <DIR>          AMD
30.12.2012  11:50    <DIR>          Brother
01.07.2012  10:43    <DIR>          Downloads
07.01.2013  12:34    9 END
04.03.2013  11:31    <DIR>          html
18.11.2012  15:25    <DIR>          MinGW
15.02.2012  18:59    <DIR>          nem
14.07.2009   05:20    <DIR>          PerfLogs
14.03.2013  11:05    <DIR>          Program Files
05.04.2013  08:54    <DIR>          Program Files (x86)
15.11.2012  09:16    <DIR>          SwSetup
22.02.2013  12:02    <DIR>          test
23.06.2011  10:51    <DIR>          Users
08.07.2012  08:21    <DIR>          WebSite X5 - Websites
27.03.2013  08:43    <DIR>          Windows
          1 Datei(en),          9 Bytes
          14 Verzeichnis(se), 147.133.280.256 Bytes frei

C:\>
```

Mit dem „cd“- Befehl können wir das entsprechende Verzeichnis ansprechen und erreichen. Hier: „cd test“ und Enter – wir befinden uns im Verzeichnis „test“ – im Windows-Explorer wäre das der Ordner „test“.

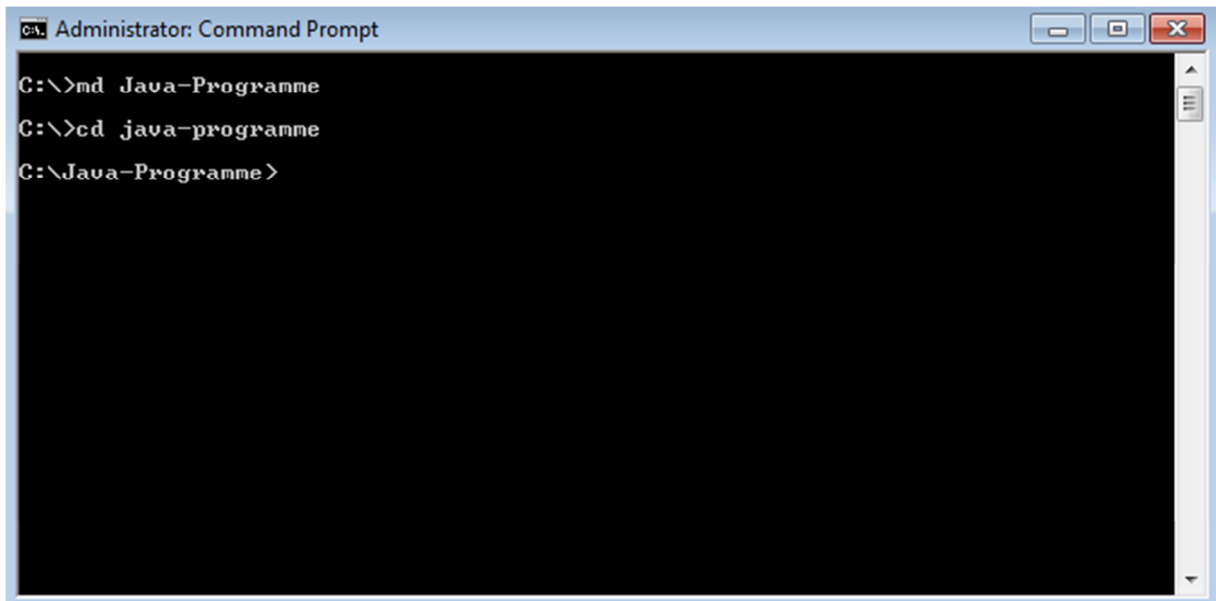
Mit „cd..“ gehen wir einen Schritt zurück, das „cd\“ bringt uns immer zurück zu „C:\>\_“.

Jetzt eine kleine Übung. Wir erstellen das Unterverzeichnis „Java-Programme“ unter „C:\>\_“.

Der Befehl dafür: „md Java-Programme“ und Enter. Kontrolle mit „dir“ und dann mit „cd Java-Programme“ dahin wechseln. Ok? Dann zurück mit „cd..“.

Ein „rd Java-Programme“ würde das gerade angelegte Unterverzeichnis von der Festplatte löschen (wenn sich darin keine Dateien oder weitere Unterverzeichnisse befinden).

Unter „Java-Programme“ wollen wir später unsere Programme speichern und ausführen.



```
Administrator: Command Prompt
C:\>md Java-Programme
C:\>cd java-programme
C:\Java-Programme>
```

So wie hier oben, sollte der Bildschirm aussehen. Den können wir jetzt schließen oder minimieren.

Jetzt brauchen wir JAVA und die nötigen Entwicklungswerkzeuge (JDK). Mit der JDK laden wir also Compiler, Interpreter sowie Debugger etc. herunter. Was das im Einzelnen ist, erfahren wir noch.

Unter:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

steht uns das ganze kostenlos zur Verfügung und kann von dort auf den Rechner geladen werden. Nach dem Aufruf der Oracle-Seite sollte das so aussehen:

## Java SE Downloads

Latest Release    Next Release (Early Access)    Embedded Use    Previous Releases



Java Platform (JDK) 7u17    JavaFX 2.2.7    JDK 7 + NetBeans

Wir klicken auf den linken Button und schauen:

### Java SE Development Kit 7u17

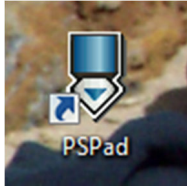
You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement     Decline License Agreement

Product / File Description	File Size	Download
Linux x86	106.65 MB	<a href="#">jdk-7u17-linux-i586.rpm</a>
Linux x86	92.97 MB	<a href="#">jdk-7u17-linux-i586.tar.gz</a>
Linux x64	104.78 MB	<a href="#">jdk-7u17-linux-x64.rpm</a>
Linux x64	91.71 MB	<a href="#">jdk-7u17-linux-x64.tar.gz</a>
Mac OS X x64	143.78 MB	<a href="#">jdk-7u17-macosx-x64.dmg</a>
Solaris x86 (SVR4 package)	135.39 MB	<a href="#">jdk-7u17-solaris-i586.tar.Z</a>
Solaris x86	91.67 MB	<a href="#">jdk-7u17-solaris-i586.tar.gz</a>
Solaris SPARC (SVR4 package)	135.92 MB	<a href="#">jdk-7u17-solaris-sparc.tar.Z</a>
Solaris SPARC	95.32 MB	<a href="#">jdk-7u17-solaris-sparc.tar.gz</a>
Solaris SPARC 64-bit (SVR4 package)	22.97 MB	<a href="#">jdk-7u17-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	17.59 MB	<a href="#">jdk-7u17-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	22.61 MB	<a href="#">jdk-7u17-solaris-x64.tar.Z</a>
Solaris x64	15.02 MB	<a href="#">jdk-7u17-solaris-x64.tar.gz</a>
Windows x86	88.75 MB	<a href="#">jdk-7u17-windows-i586.exe</a>
Windows x64	90.42 MB	<a href="#">jdk-7u17-windows-x64.exe</a>

Wir accepten natürlich the License Agreement und laden die zu unserem System passende Datei herunter und führen sie dann aus. Das sollte alles keine größeren Probleme machen.

Nun brauchen wir noch einen Editor, in dem wir unsere Programme eintippen können. Ich empfehle für die ersten Übungen und die paar kleinen Programme „PSPad“.



<http://www.pspad.com/de/>

Nach meiner Erfahrung übt das ständige Tippen und Wiederholen der Anweisungen und Befehle Hand und Auge und vertieft das Verständnis für den Programmablauf und den Syntax. Später werden wir mit der Entwicklungsumgebung von Eclipse weiterarbeiten, aber das ist ein Kapitel für sich, am Anfang ziemlich komplex und damit eher verwirrend. Jeder wird das für sich entscheiden müssen, wie und womit er arbeiten will. Ich selbst bin von Eclipse zurück zu PSPad gegangen, wenn es sich um reine Java-Themen handelte. Mit einem tieferen Verständnis für die Zusammenhänge (kompilieren, Apps bauen usw. usf.) ist die Arbeit mit Eclipse aber ein Hochgenuss. Überhaupt wird es sich im Lauf meiner Ausführungen herausstellen, dass sich die Themen verzweigen, überlappen und damit auch „verfummeln“ und zerfransen. Irgendwann wird sich jeder die Frage stellen müssen, welches Thema ihn begeistert und wo er weiter und tiefer graben möchte.

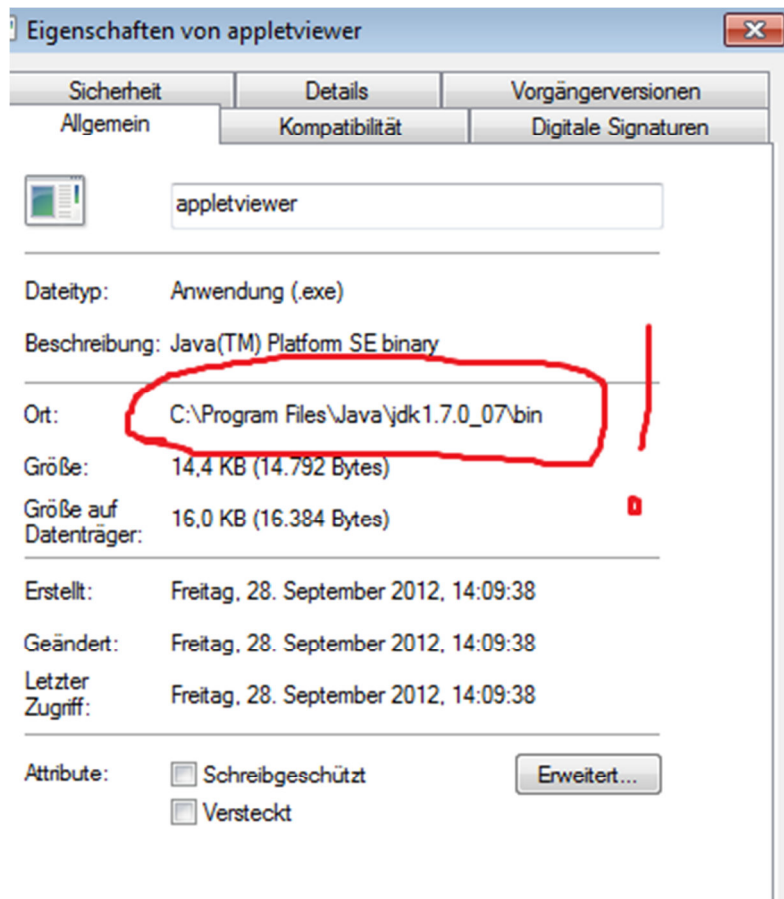
Wahrscheinlich werden wir alle keine Profi-Programmierer, Web-Entwickler udgl. werden, aber wir werden Freude und Stolz empfinden, wenn wir unseren Freunden, Söhnen und Töchtern, den Computer-Neffen und Schwiegersöhnen mal eben das „tree“ vorführen, abschätzig nickend und kurz stöhnend den Kopf schüttelnd um dann zu erklären, wie man einem Computer beibringt, das Ergebnis von 1+1 auszurechnen und dann noch auszugeben.

Ruhm sei Euch gewiss!

Ohne das jetzt Folgende im Detail zu verstehen, sind folgende Taten noch zu vollbringen, um das Ganze funktionieren zu lassen. Denn wir müssen unseren Computer noch mitteilen, was er tun muss, um Java-Programme zu übersetzen und dann auch ausführen zu können.

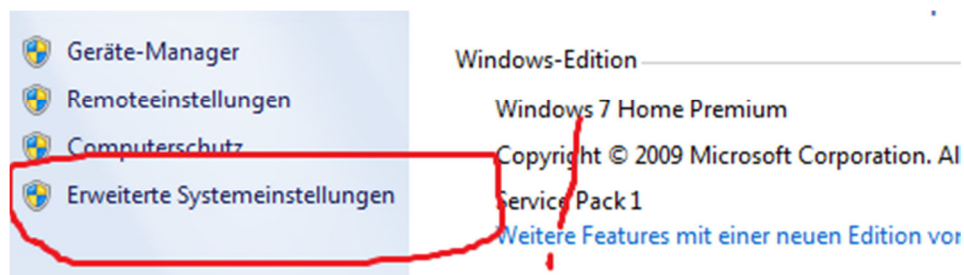
1. Windows-Button anklicken
2. Computer anklicken
3. Festplatte C: doppelklicken (jetzt sehen wir alle Ordner wieder, die wir vorhin im Command-Prompt gesehen haben)
4. den Ordner „Programme“ öffnen (doppelklicken)
5. und dort sollten wir den Ordner „Java“ vorfinden (wenn nicht, ist hier jetzt Schluss)
6. den Ordner „Java“ öffnen
7. den Ordner „JDK 1.7\_07“ öffnen
8. den Ordner „bin“ öffnen
9. Rechtsklick auf eine Anwendungsdatei in diesem Verzeichnis
10. „Eigenschaften“ öffnen

Dieses Fenster sollten wir jetzt sehen:

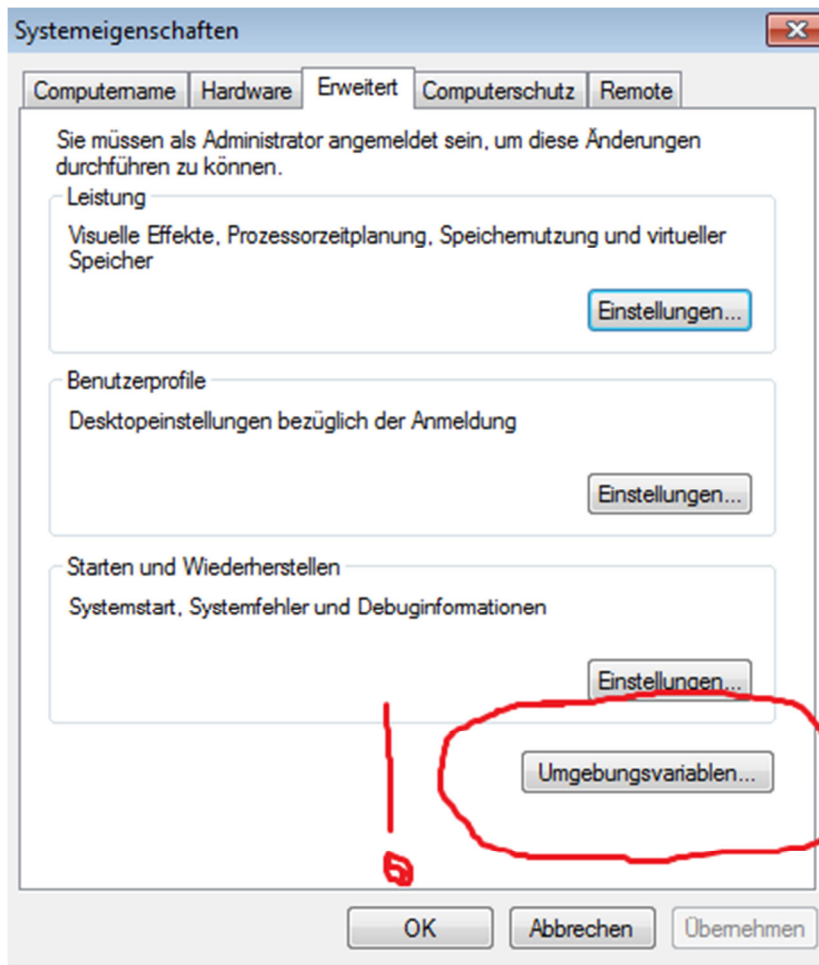


Den Pfad unter Ort markieren wir, und mittels der rechten Maustaste kopieren wir ihn uns in die Zwischenablage.

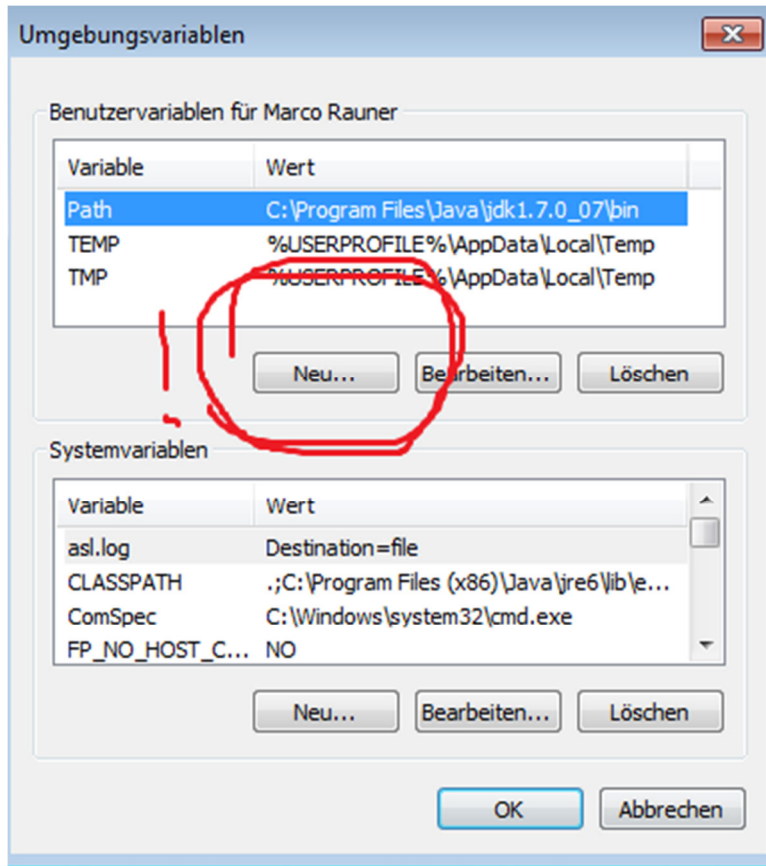
1. im Start-Menü auf „Computer“ klicken, rechte Maustaste drücken – auf „Eigenschaften“ klicken
2. „Erweiterte Systemeinstellungen“ öffnen (siehe Bild unten)



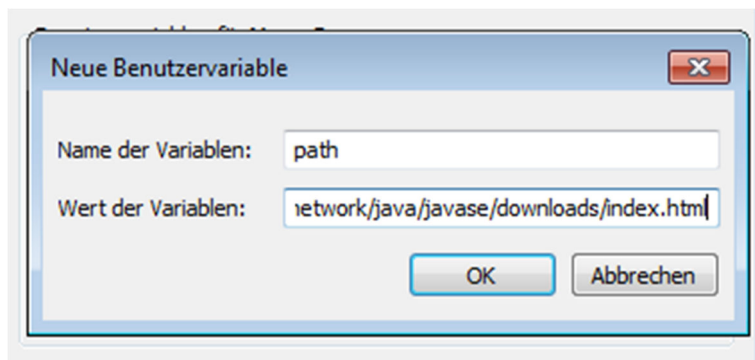
3. den Button „Umgebungsvariablen“ anklicken



- den Button „Neu“ anklicken



- „Name der Variablen:“ eingeben (path) und bei „Wert der Variable“ den Pfad aus der Zwischenablage einfügen



- alle Fenster mit den Button „OK“ schließen

Geschafft!!!



Geben wir jetzt in unserem Command-Prompt hinter C:>\_ den Befehl „javac“ ein, muss dringend folgendes erscheinen:



```
Administrator: Command Prompt
C:\>javac
Usage: javac <options> <source files>
where possible options include:
  -g                Generate all debugging info
  -g:none           Generate no debugging info
  -g:{lines,vars,source}  Generate only some debugging info
  -nowarn           Generate no warnings
  -verbose          Output messages about what the compiler is doing
  -deprecation      Output source locations where deprecated APIs are used
  -classpath <path> Specify where to find user class files and annotations processors
  -cp <path>        Specify where to find user class files and annotations processors
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>   Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -proc:{none,only} Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>... ] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path> Specify where to find annotation processors
  -d <directory>    Specify where to place generated class files
  -s <directory>    Specify where to place generated source files
  -implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files
  -encoding <encoding> Specify character encoding used by source files
  -source <release> Provide source compatibility with specified release

  -target <release>  Generate class files for specific VM version
  -version           Version information
  -help             Print a synopsis of standard options
  -Akey[=value]     Options to pass to annotation processors
  -X               Print a synopsis of nonstandard options
  -J<flag>          Pass <flag> directly to the runtime system
  -Werror           Terminate compilation if warnings occur
  @<filename>       Read options and filenames from file

C:\>
```

Wunderbar.