

Einführung in die Informatik 1 - Javainstallationsanleitung

Die Tutoren

Stand: 19. Oktober 2009

Inhaltsverzeichnis

1	Einführung in Java	1
1.1	Installation von Java unter Mac	1
1.2	Benutzung von Java	1
1.2.1	Verzeichniswechsel im Terminal	1
1.2.2	Kompilieren von Java-Programmen mit "Terminal"	2
1.2.3	Java-Programme mit "Terminal" starten	4
1.2.4	Schreiben von Java-Programmen	4
1.3	Installation von Java unter Linux	4
1.4	Installation von Java unter Windows XP	6
1.4.1	Vorabüberprüfung bereits installierter Java-Komponenten	6
1.4.2	Installation des JDK 1.6	6
1.5	Installation von Java unter Windows Vista	7
1.6	Installation von Java unter Windows 7	7
1.6.1	Überprüfung bereits installierter Komponenten	7
1.6.2	Installation des JDK 1.6	7
1.7	Benutzung von Java unter Windows	10
1.7.1	Erstellen einer Java Datei	10
1.7.2	Sichtbarmachen der Endung	10
1.7.3	Eingabeaufforderung öffnen	14
1.7.4	Kompilieren und Ausführen	14
1.8	Das erste Java-Programm: Hello World!	14

1 Einführung in Java

1.1 Installation von Java unter Mac

Um Java-Programme kompilieren und testen zu können, benötigen Sie das Java Development Kit (JDK). Das JDK ist bei Mac OS X Leopard vorinstalliert. Sie sollten nur überprüfen, ob die bei Ihnen installierte Version auch die Neueste ist. Wählen Sie dafür in der Menüzeile "Apfel" → "Softwareaktualisierung ...". Im Rechnernetz der TU-Berlin wird momentan die Version 1.5 (auch 5.0 genannt) von Java verwendet. Damit Ihre Programme einschränkungslos im Rechnernetz lauffähig sind, sollte genau diese Version von Java auf Ihrem Mac installiert sein. Das entsprechende Auswahlmü befindet sich unter "Programme" → "Dienstprogramme": "Java-Einstellungen.app". Starten Sie das Programm. In Abbildung 1.1 ist das entsprechende Fenster dargestellt.

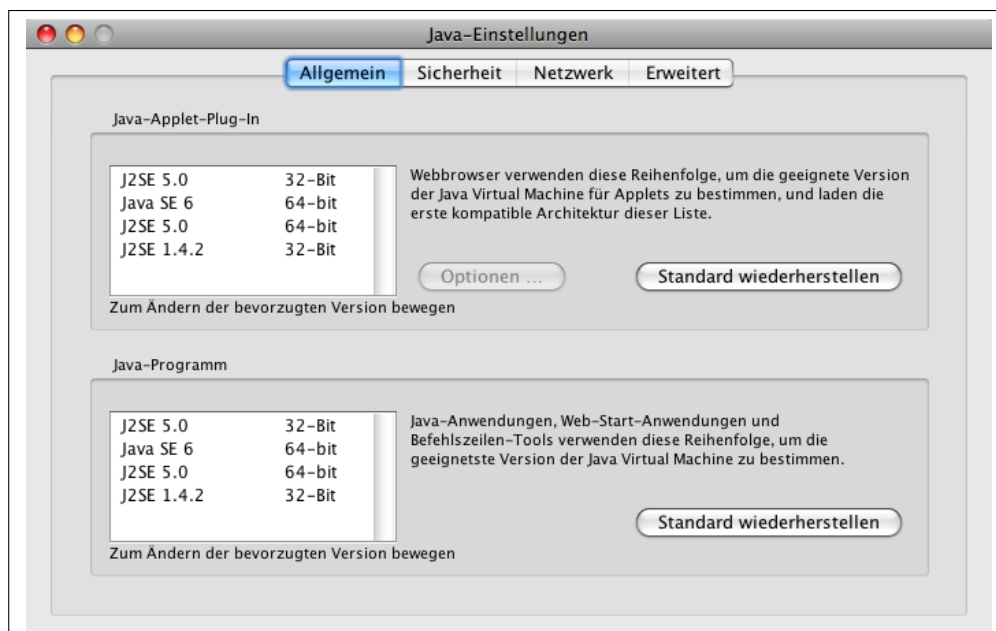


Abbildung 1.1: Java-Einstellungen.app nach dem Starten.

In den Listen "Java-Applet-Plug-In" und "Java-Programm" sollte "J2SE 5.0 32-bit" in der obersten Zeile stehen. Ist dies nicht der Fall, ändern Sie diese Einstellungen, indem Sie die Einträge verschieben. Die übrigen Registerkarten von "Java-Einstellungen.app" haben hier keine Relevanz. Beenden Sie das Konfigurationsprogramm.

1.2 Benutzung von Java

Die folgenden Schritte zeigen, wie Sie die fertige Java-Installation nutzen können.

1.2.1 Verzeichniswechsel im Terminal

In dieser Beschreibung wird "Terminal.app" (vorinstalliert) benutzt. Sie können aber ebenfalls "X11.app" oder andere Konsolen benutzen.

Vor dem Starten eines Java-Programms muss dieses erst mit dem Java-Compiler kompiliert werden, bevor es ausführbar ist. Starten Sie dazu "Terminal.app", welches sich unter "Programme" → "Dienstprogramme" befindet. Das Programm lässt sich auch starten, indem Sie in "Spotlight" (Lupe, oben rechts in der Mac OS X-Menüzeile) »Terminal« eingeben. Beim Ausführen öffnet sich ein Fenster wie in 1.2.

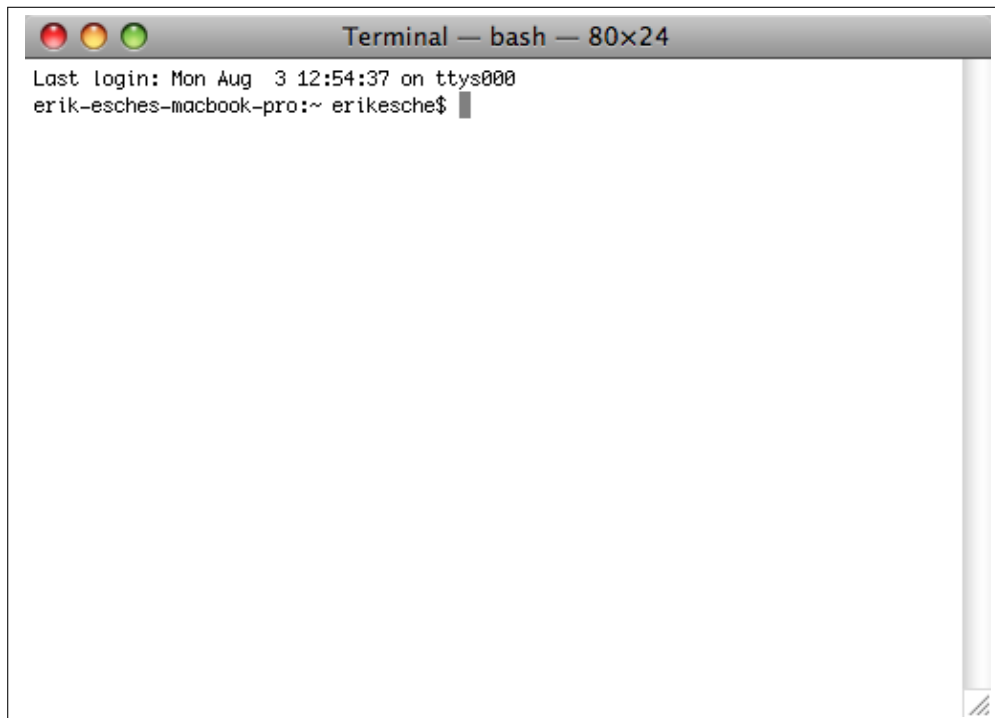


Abbildung 1.2: "Terminal.app".

Geben Sie den Befehl »ls« ein. Dieser Befehl gibt eine Liste der Verzeichnisse und Dateien des aktuellen Verzeichnisses aus (siehe Abbildung 1.3). Es wird im Weiteren davon ausgegangen, dass sich Ihre Java-Dateien in einem Verzeichnis "Inftech" auf dem Desktop (Schreibtisch) befinden.

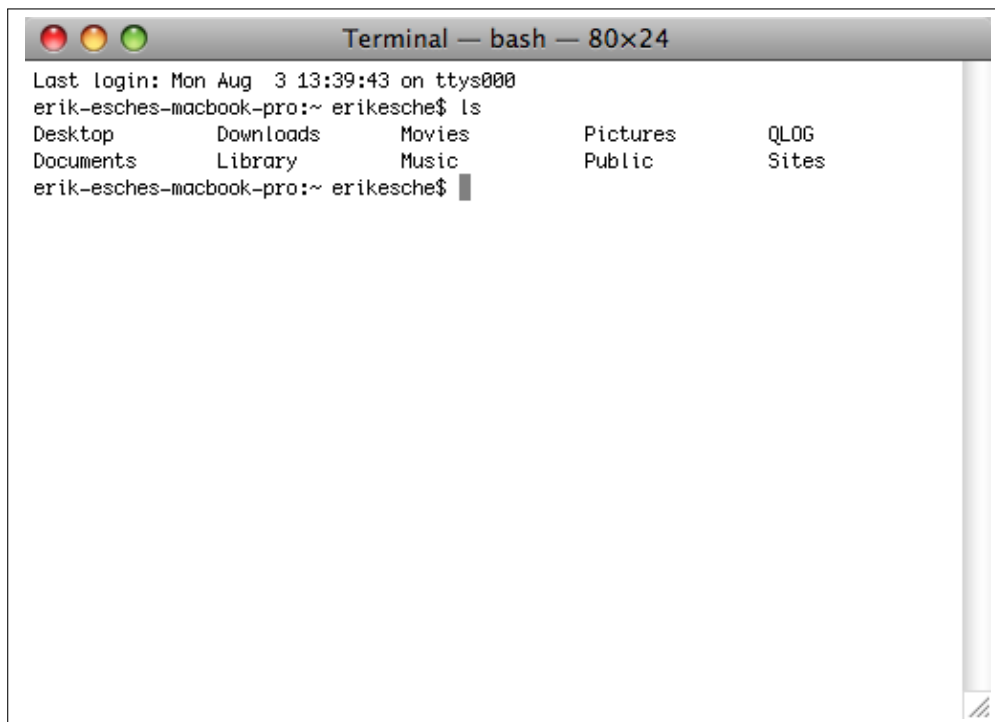
»ls« zeigt Ihnen, dass Sie sich in dem Desktop übergeordneten Verzeichnis befinden.

Wechseln Sie nun mit »cd Desktop« zum Desktopverzeichnis.

Mit »ls« wird der Inhalt des Desktops angezeigt und mit dem Befehl »cd Inftech« kommen Sie in das Verzeichnis, in dem sich die Java-Dateien befinden. Diese Ein- und Ausgaben sind in Abbildung 1.4 dargestellt. *Die hier gezeigten Befehle funktionieren natürlich in jedem beliebigen Verzeichnis. D.h. mit "Terminal" können Sie durch »cd /Users/<username>/Desktop/Inftech« in jedes gewünschte Verzeichnis wechseln. An Stelle von »<username>« müssen Sie lediglich Ihren Benutzernamen eintragen. Dieser wird Ihnen von "Terminal" angezeigt; in diesem Beispiel ist der Benutzername »erikesche«.*

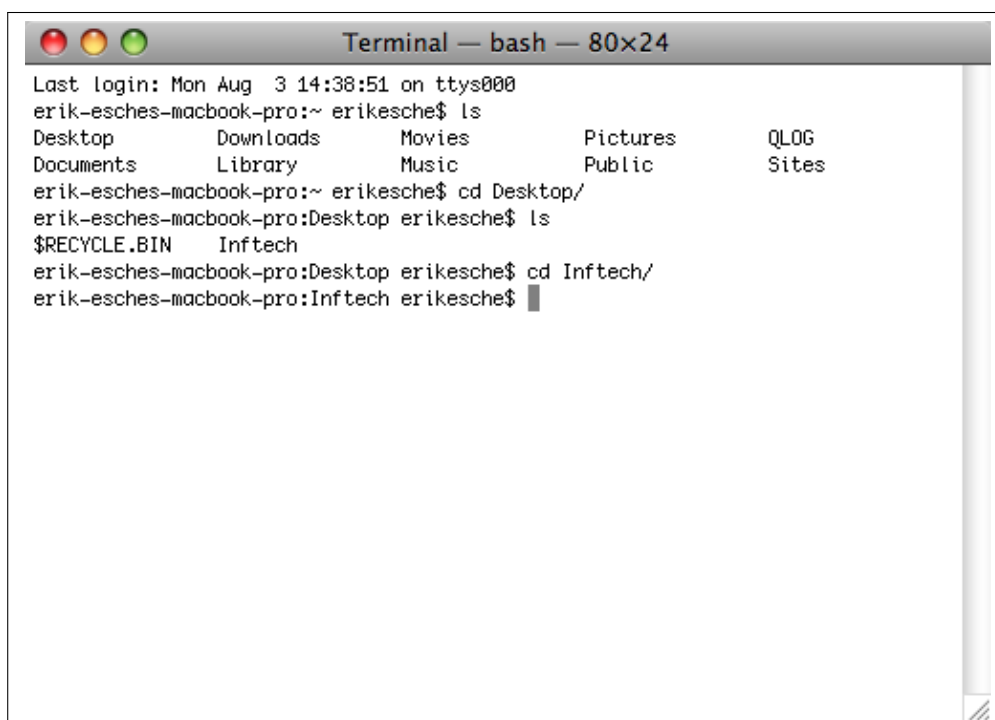
1.2.2 Kompilieren von Java-Programmen mit "Terminal"

Kompilieren ist das Übersetzen Ihres Programmcodes. Der Java-Compiler weist während des Übersetzens noch auf etwaige Fehler hin, die noch korrigiert werden müssen, bevor das Programm lauffähig ist. Wechseln Sie wie im vorherigen Abschnitt beschrieben in das Verzeichnis, in dem sich ihre Java-Dateien befinden. Mit dem Befehl »javac Test.java« lässt sich nun die Datei mit dem Namen »Test.java« kompilieren. Wollen Sie gleich mehrere Dateien auf einmal kompilieren, geben Sie deren Namen einfach nach dem Befehl »javac« durch Leertasten getrennt ein. Benutzen Sie den Befehl »javac *.java«, um sofort alle Java-Dateien des aktuellen Verzeichnisses zu kompilieren. Mit dem erfolgreichen Kompilieren entstehen class-Dateien, welche den übersetzten Programmcode enthalten; nach dem Kompilieren von "Test.java" entsteht eine Datei "Test.class".



```
Terminal — bash — 80x24
Last login: Mon Aug  3 13:39:43 on ttys000
erik-esches-macbook-pro:~ erikesche$ ls
Desktop      Downloads    Movies       Pictures     QLOG
Documents    Library      Music        Public       Sites
erik-esches-macbook-pro:~ erikesche$
```

Abbildung 1.3: Eingabe von »ls« direkt nach dem Starten von »Terminal«.



```
Terminal — bash — 80x24
Last login: Mon Aug  3 14:38:51 on ttys000
erik-esches-macbook-pro:~ erikesche$ ls
Desktop      Downloads    Movies       Pictures     QLOG
Documents    Library      Music        Public       Sites
erik-esches-macbook-pro:~ erikesche$ cd Desktop/
erik-esches-macbook-pro:Desktop erikesche$ ls
$RECYCLE.BIN  Inftech
erik-esches-macbook-pro:Desktop erikesche$ cd Inftech/
erik-esches-macbook-pro:Inftech erikesche$
```

Abbildung 1.4: Schritte zum Wechsel in das Beispiilverzeichnis »Inftech« auf dem Desktop.

1.2.3 Java-Programme mit "Terminal" starten

Nach dem Kompilieren können Sie Ihre Java-Programme starten. Dies geschieht in unserem Beispiel mit dem Befehl: `»java Test«` mit der Datei namens "Test.class". Die Dateierweiterung fällt also weg.

1.2.4 Schreiben von Java-Programmen

Zum Schreiben von Java-Programmen kann zum Beispiel das Mac OS-eigene Textprogramm "TextEdit.app" benutzt werden, welches sich direkt im Programmordner befindet. Beim Speichern Ihrer Datei müssen Sie aber darauf achten, dass Sie nicht im ".txt"- sondern im ".java"-Format speichern. Gehen Sie im Programmmenü auf "Ablage" → "Sichern". Es erscheint das übliche Fenster zum Speichern der Datei (siehe Abbildung 1.5). Wechseln Sie in das gewünschte Verzeichnis zum Speichern und geben Sie als Dateinamen zum Beispiel `»Test.java«` ein (siehe Abbildung 1.6). Klicken Sie nun auf `»Sichern«` und es erscheint ein weiteres Fenster (siehe Abbildung 1.7) mit der Frage, ob Sie tatsächlich die Dateierweiterung ".java" verwenden wollen. Bestätigen Sie dies mit `»".java" verwenden«`.

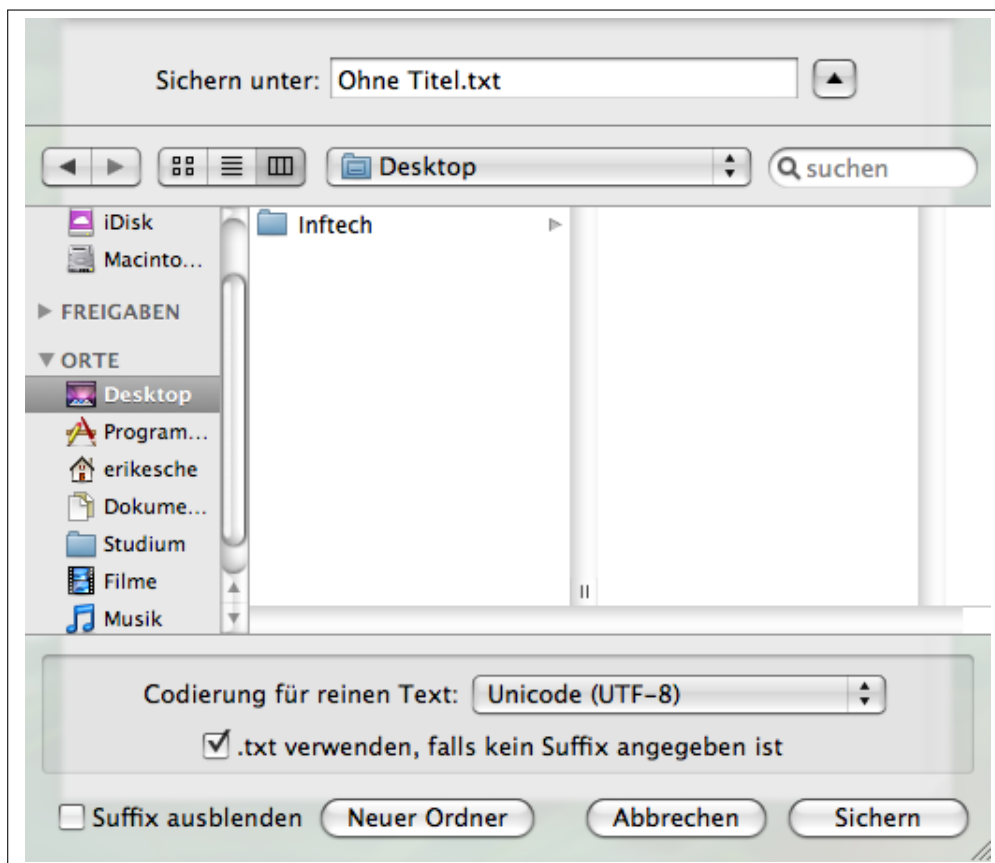


Abbildung 1.5: Gerade geöffnetes Speichermenü in "TextEdit.app".

1.3 Installation von Java unter Linux

Für die Installation unter linuxbasierten Betriebssystemen ist keine Anleitung verfügbar, da je nach Version andere Schritte notwendig sind. Bitte informieren Sie sich selbst wie das JDK auf ihrem System installiert wird und ob dies überhaupt notwendig ist.

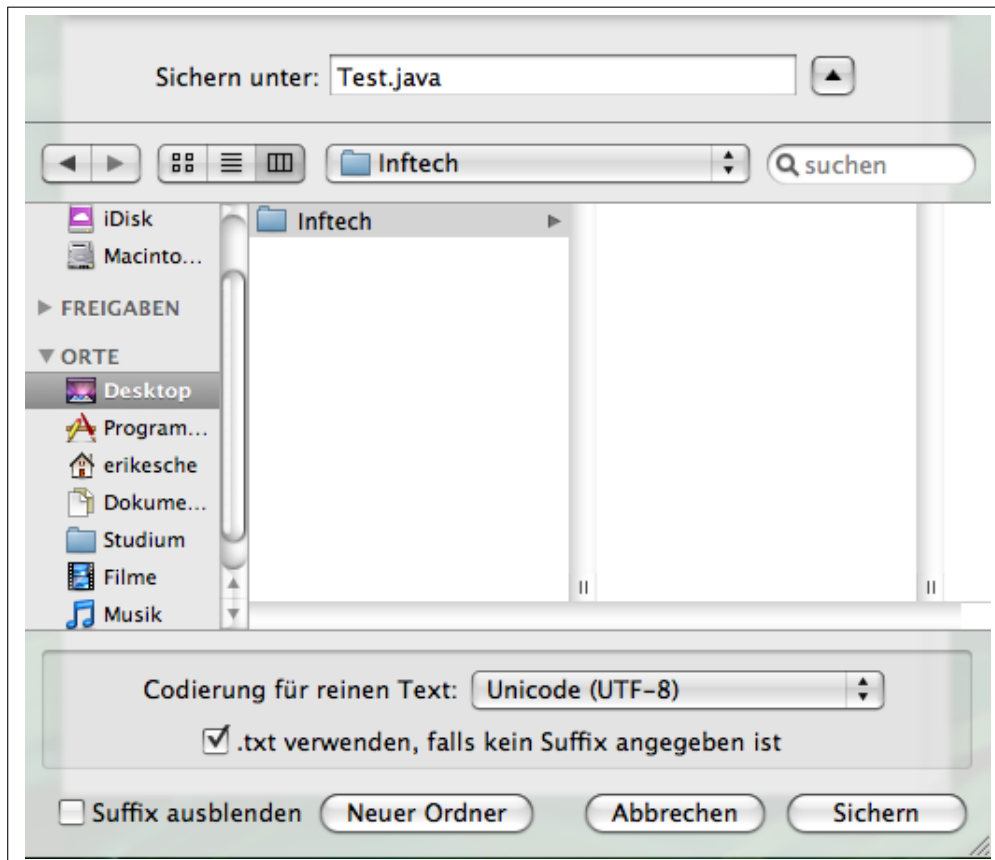


Abbildung 1.6: Test.java in "TextEdit.app" speichern.



Abbildung 1.7: Abfrage zum Speichern mit der Dateierdung ".java".

1.4 Installation von Java unter Windows XP

Anders als unter Mac OS X und manchen Linux-Distributionen ist das Java Development Kit (JDK) unter Windows nicht vorinstalliert. Je nach Rechner ist gar keine Java-Komponente oder nur das Java Runtime Environment (JRE) installiert, welches nicht ausreicht, um Java-Programme auszuführen. Zuerst muss überprüft werden, welche Java-Komponenten schon auf Ihrem Windows XP installiert sind. Dies und die Installation der fehlenden Komponenten wird im Folgenden gezeigt. Sollten Sie eine ältere Windows-Version (z.B. 2000, ME oder 98) nutzen, so können Sie zur Installation die Anleitung unter <http://www.torsten-horn.de/techdocs/java-install.htm> nutzen. Zu den neueren Windows-Versionen (Vista und 7) gibt es hier gesonderte Kapitel.

1.4.1 Vorabüberprüfung bereits installierter Java-Komponenten

Öffnen Sie die "Eingabeaufforderung", welche Sie unter "Start" → "Alle Programme" → "Zubehör" → "Eingabeaufforderung" finden. Geben Sie den Befehl »java -version« ein. Um mit dem Rechnernetz der TU kompatibel zu sein, benötigen Sie die Java-Version 1.5. Wenn Ihr Rechner nach der Eingabe eine Fehlermeldung ausgibt, dann ist keine Java-Komponente installiert. Andernfalls müsste eine Versionsnummer angegeben werden, die unter Umständen von 1.5 abweicht. Beim Verfassen dieser Anleitung war Java 1.6 die aktuellste Version; es kann aber auch eine noch ältere Java-Version installiert sein.

Sollten Sie die richtige Versionsnummer haben, muss nur noch überprüft werden, ob Java vollständig installiert ist. Geben Sie dazu den Befehl »javac« in die "Eingabeaufforderung" ein. Wenn das gesamte JDK installiert ist, sollte nun eine Auflistung von Befehlsoptionen erscheinen, andernfalls eine Fehlermeldung, dass der Befehl nicht bekannt ist.

Sollte Ihre Versionsnummer abweichen oder nicht das gesamte JDK installiert sein, fahren Sie mit den folgenden Schritten fort.

1.4.2 Installation des JDK 1.6

Das JDK 1.5, welches im TU-Rechnernetz installiert ist, ist leider nicht mehr von SUN erhältlich und der Support wird im Oktober 2009 eingestellt. Statt dessen können Sie das neue JDK 1.6 nutzen (auch JAVA SE JDK 6 genannt). Die aktuelle Version des JDK können Sie über <http://java.sun.com/javase/downloads/index.jsp> beziehen. Wählen Sie unter der Rubrik **Java SE Development Kit** das aktuellste Update, z.B. **JDK 6 Update 16**. Folgen Sie den Download- und Installationshinweisen der Webseite.

Nach der Installation muss sichergestellt werden, dass die Java-Installation in den System-Umgebungsvariablen eingetragen ist; nur dann können Sie diese auch nutzen. Die Umgebungsvariablen finden Sie in Windows XP unter „Start“ → Rechtsklick auf „Arbeitsplatz“ → „Eigenschaften“ → Registerkarte „Erweitert“ → „Umgebungsvariablen“: Es öffnet sich ein Fenster, in dem sich im unteren Kasten die Systemvariablen befinden. Bei der Bearbeitung der Einträge ist zu beachten, dass bereits eingetragene Werte nicht gelöscht werden dürfen:

Als Erstes wird die Variable **CLASSPATH** bearbeitet: Bei einer Standardinstallation sollte der folgende Eintrag enthalten sein:

„C:\Programme\Java\jre6\lib\ext\QTJava.zip“ (ohne die Anführungszeichen). Einzelne Einträge werden durch Semikola getrennt.

Als Nächstes sollte die Systemvariable **Path** überprüft werden: Hier sollte folgender Eintrag stehen:

„C:\Programme\Java\jdk1.6.0_16\bin“

Die letzte Variable, die eventuell bearbeitet werden muss, ist **QTJAVA**: Hier sollte folgender Eintrag stehen: „C:\Programme\Java\jre6\lib\ext\QTJava.zip“

Nun können Sie die vollständige JAVA-Installation nutzen.

1.5 Installation von Java unter Windows Vista

Die Installation und Einrichtung von Java unter Windows Vista und Windows 7 sind grundlegend gleich. Es gibt nur geringfügige Unterschiede. Nutzen Sie deshalb bitte die Installationsanleitung für Windows 7.

1.6 Installation von Java unter Windows 7

Die Installation von Java unter Windows 7 ist sehr ähnlich zu der unter den Vorgängerversionen des Betriebssystems. Auch hier werden die gleichen Schritte ausgeführt, die teilweise nur an anderer Stelle zu finden sind.

1.6.1 Überprüfung bereits installierter Komponenten

Öffnen Sie die „Eingabeaufforderung“ öffnen. Diese befindet sich unter: „Windows Symbol“ → „Alle Programme“ → „Zubehör“ → „Eingabeaufforderung“.

Abbildung 1.8 zeigt ein Beispiel der „Eingabeaufforderung“.

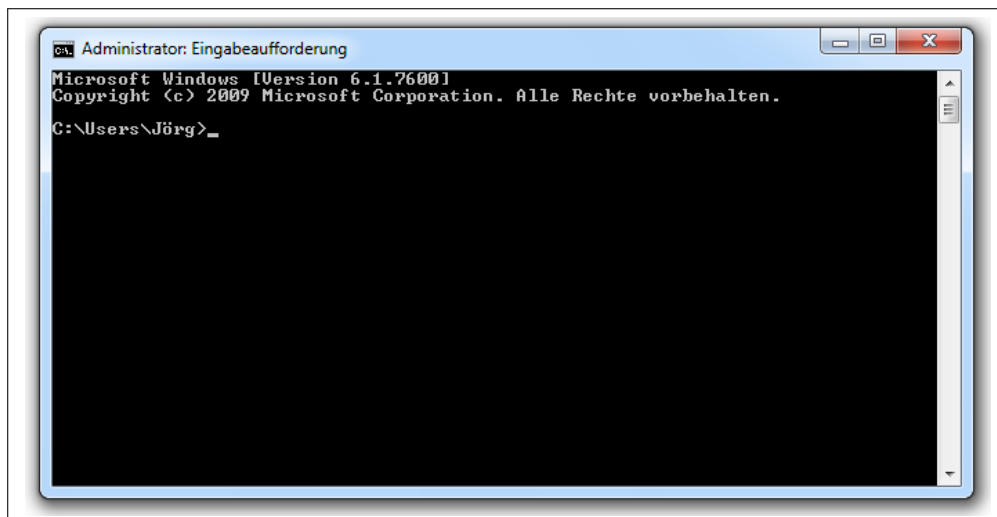


Abbildung 1.8: leere Eingabeaufforderung

Geben Sie die „Eingabeaufforderung“ den Befehl „javac“ ein. Falls Java noch nicht vollständig installiert ist, erscheint eine Ausgabe wie in Abbildung 1.9 auf der folgenden Seite dargestellt.

Erscheint eine andere Ausgabe, ist Java möglicherweise schon korrekt installiert.

1.6.2 Installation des JDK 1.6

Im Folgenden wird erklärt, wie Java installiert wird, falls dies noch nicht korrekt geschehen ist.

Schritt 1: Download

1. Öffnen Sie mit einem Browser die Seite <http://java.sun.com/javase/downloads/index.jsp>
2. Laden Sie dort das Java SE Development Kit (JDK) herunter. (Unter der Schaltfläche in Abbildung 1.10 auf der nächsten Seite.)

Schritt 2: Installation

Vor der Installation sollte der Ordner „C:\Programme\Java\“ wie in Abbildung 1.11 auf der folgenden Seite aussehen.



Abbildung 1.9: javac funktioniert noch nicht korrekt

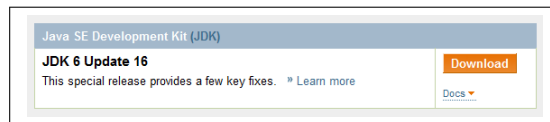


Abbildung 1.10: Downloadfläche für Java

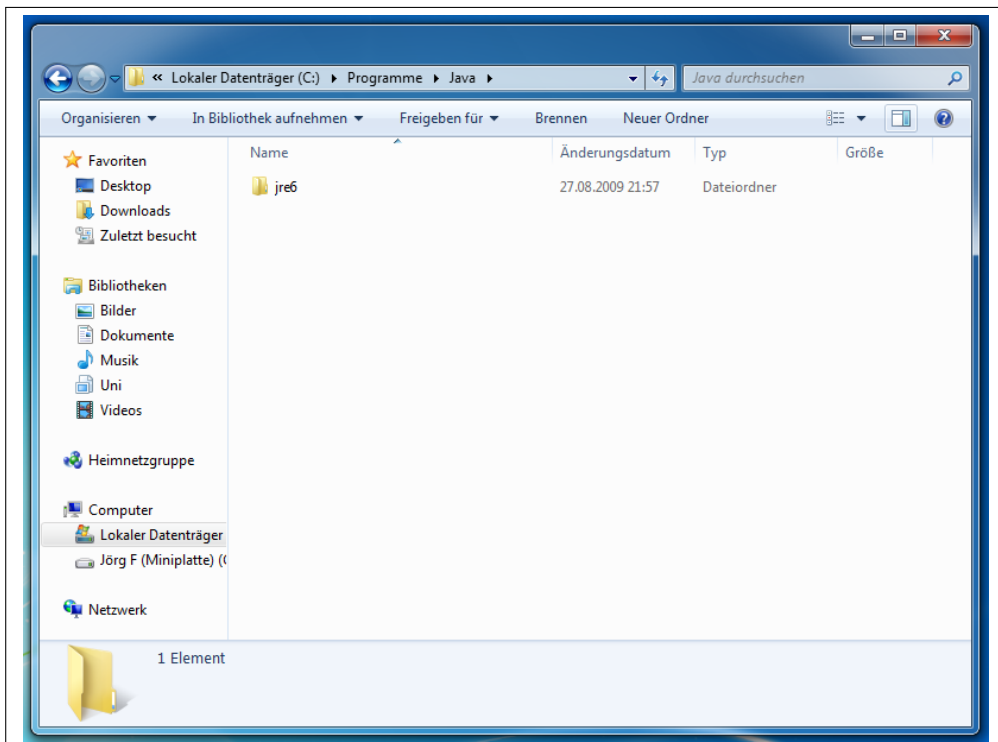


Abbildung 1.11: der Javaordner vor der Installation des JDK

Öffnen Sie nun die Datei, die in Schritt 1 heruntergeladen wurde und folgen Sie den Anweisungen der Installation. Lassen Sie den Pfad der Installation möglichst auf dem vordefinierten Wert. War die Installation erfolgreich, hat sich der Java-Ordner ähnlich wie in Abbildung 1.12 verändert.

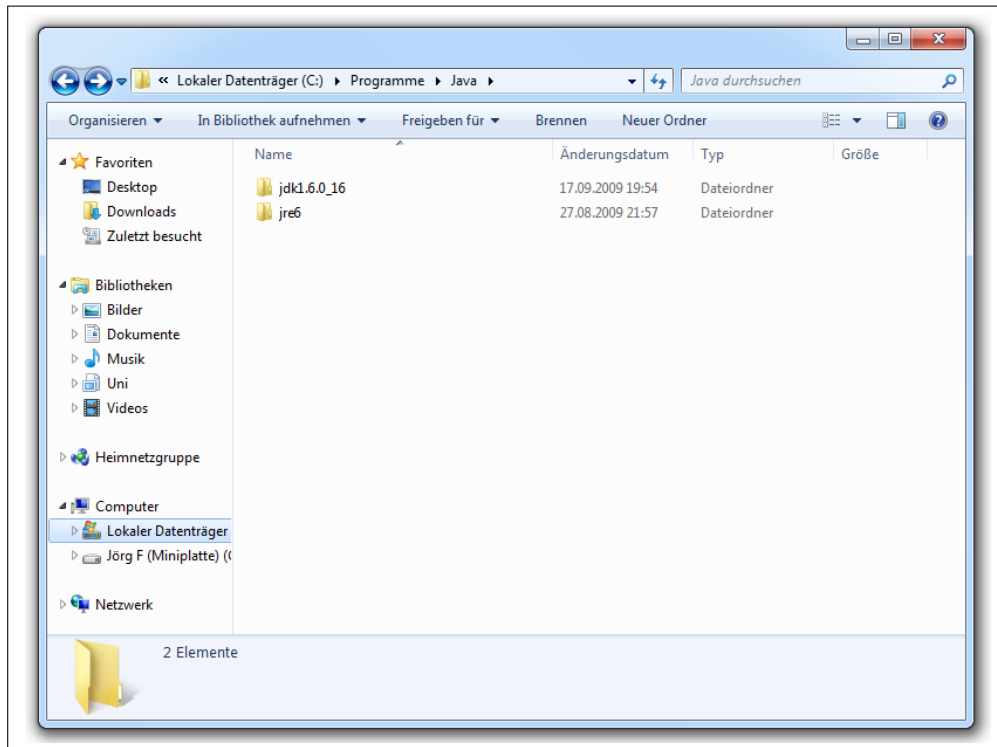


Abbildung 1.12: der Javaordner nach der Installation des JDK

Schritt 3: Einbinden des JDK in Windows

Nachdem Sie nun Java installiert haben, ist es noch nicht einsatzbereit. Es muss noch in Windows eingebunden werden.

1. Klicken Sie auf das Windows Symbol in der Taskleiste und machen Sie einen Rechtsklick auf „Computer“. (Abbildung 1.13)

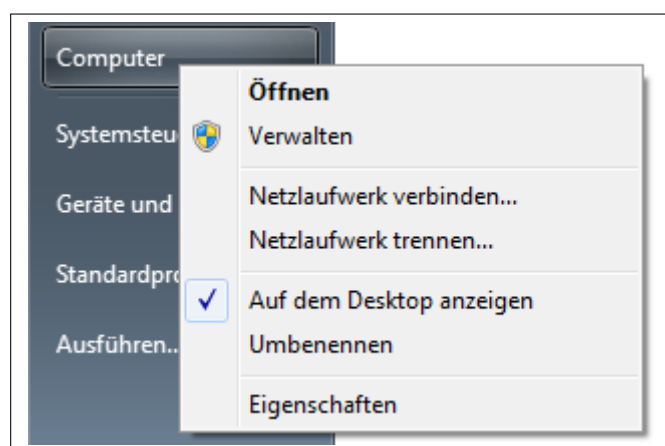


Abbildung 1.13: Rechtsklick auf „Computer“

2. Klicken Sie auf Eigenschaften.
3. Es befindet sich am linken Rand ein Feld „Erweiterte Systemeinstellungen“, auf das geklickt werden muss.

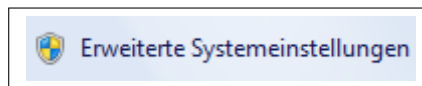


Abbildung 1.14: „Erweiterte Systemeinstellungen“

4. Klicken Sie hier auf die Schaltfläche im unteren Bereich mit der Aufschrift „Umgebungsvariablen“.

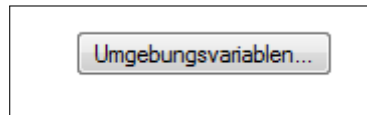


Abbildung 1.15: „Umgebungsvariablen“

5. Im unteren Bereich des aktuellen Fensters finden Sie die Systemvariablen. Suchen Sie dort das Feld „Path“ und klicken Sie es an.
6. Klicken Sie nun auf „Bearbeiten“.
7. Ergänzen Sie dort den in Abbildung 1.17 auf Seite 12 zu sehenden Text.
8. Schließen Sie alle offenen Fenster.

Schritt 4: Erneutes Testen

Testen Sie erneut in der „Eingabeaufforderung“, ob der Befehl „javac“ funktioniert. Ist dies der Fall sein, dann sollte sich die Eingabe wie in Abbildung 1.18 auf Seite 13 verändert haben.

1.7 Benutzung von Java unter Windows

1.7.1 Erstellen einer Java Datei

Eine *.java-Datei ist im Grunde das Gleiche, wie eine *.txt-Datei; nur die Endung ist eine Andere. Deshalb ist es möglich an einem Ort Ihrer Wahl einfach folgende Schritte ausführen:

1. Rechtsklick.
2. „Neu“.
3. „Textdokument“.

Geben Sie einen Namen für die Datei ein und bestätigen Sie entweder mit Enter auf der Tastatur oder indem Sie in einen Bereich jenseits der Datei klicken.

Wenn Sie bereits die Dateierweiterung .txt sehen, ändern Sie diese in .java und überspringen Sie den nächsten Punkt.

1.7.2 Sichtbarmachen der Endung

Falls Sie, wie im vorigen Punkt bereits erwähnt, die Dateierweiterung nicht sehen können, muss diese sichtbar machen.

Führen Sie folgende Schritte aus:

1. Öffnen Sie ein „Explorer“-Fenster.
2. Klicken Sie auf die Schaltfläche „Organisieren“ oder „Datei“ (je nach Windows Version).

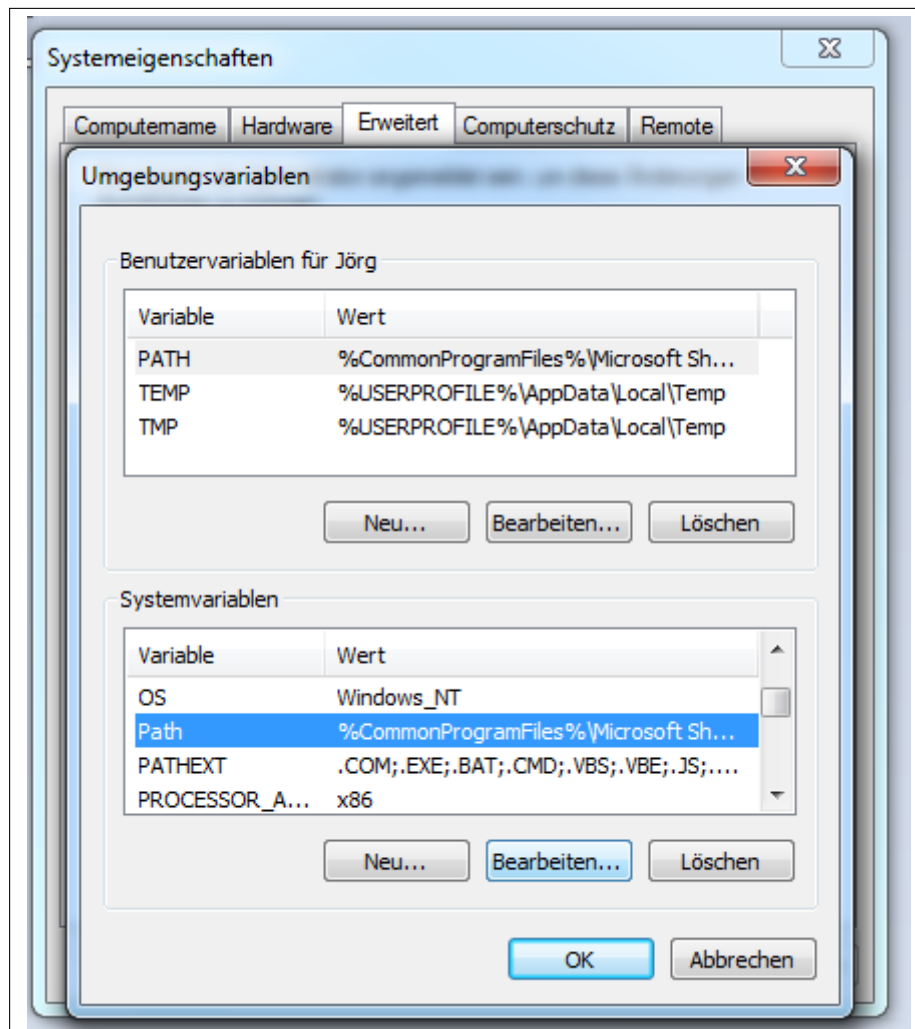


Abbildung 1.16: das Feld Systemvariablen

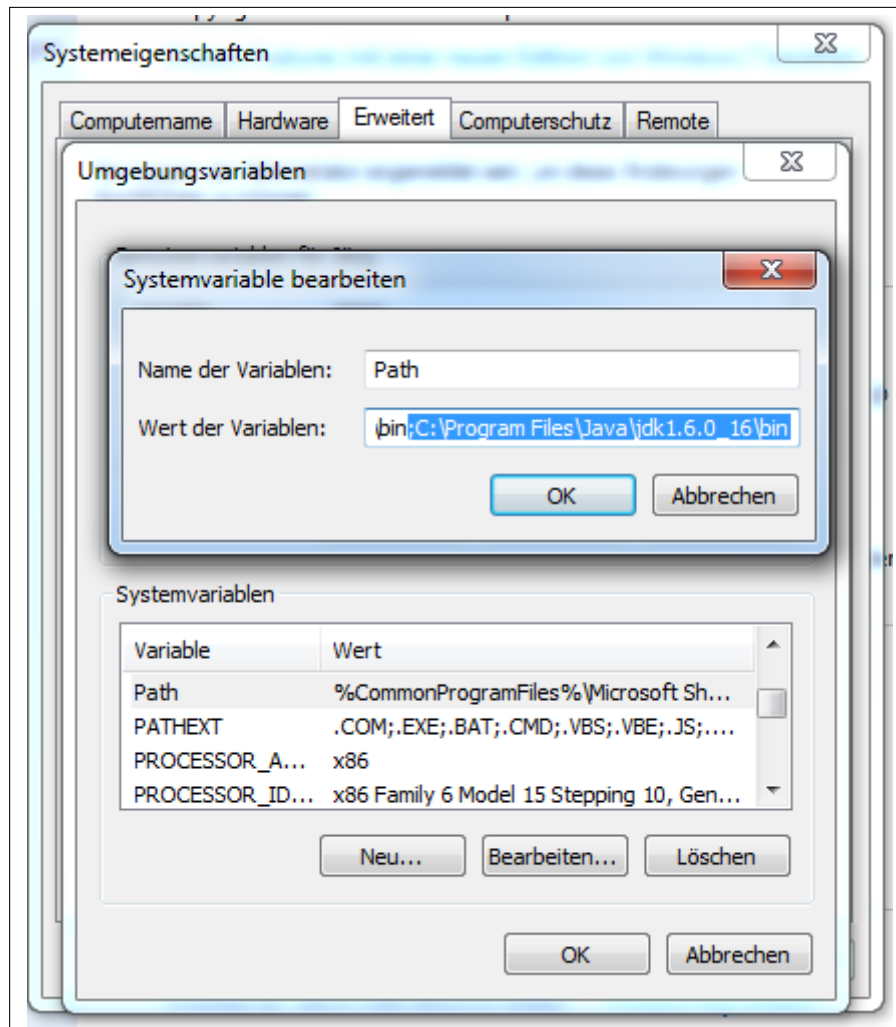
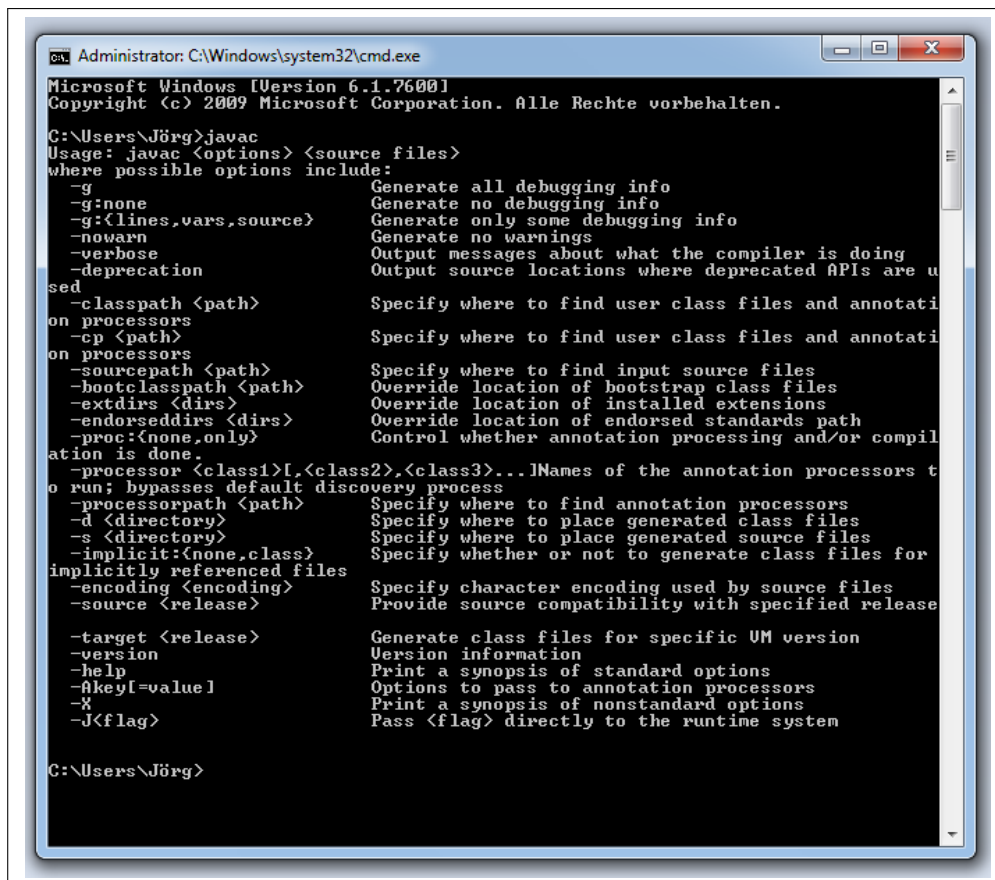


Abbildung 1.17: das Feld Path



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Jörg>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
  -cp <path>            Specify where to find user class files and annotations
  -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

C:\Users\Jörg>
```

Abbildung 1.18: Javac war erfolgreich

3. Wählen Sie „Ordneroptionen“ aus.
4. Öffnen Sie den Reiter „Ansicht“.
5. Entfernen Sie das Häkchen unter „Dateiendung bekannter Dateitypen ausblenden“.
6. Bestätigen Sie mit einem Klick auf OK.

Die Datei kann jetzt wie im vorherigen Abschnitt beschrieben umbenannt werden.

1.7.3 Eingabeaufforderung öffnen

Öffnen Sie den „Explorer“ im Ordner, in dem die java Datei liegt und wechseln Sie anschließend in den übergeordneten Ordner. Halten Sie die Taste „Shift“ gedrückt und klicken Sie dann mit der rechten Maustaste auf den Ordner, der Ihre Java Datei enthält.

Wählen Sie „Eingabeaufforderung hier öffnen“.

1.7.4 Kompilieren und Ausführen

Geben Sie in die „Eingabeaufforderung“ den Befehl „javac *.java“ ein um alle Javodateien in dem Ordner zu kompilieren. Ggf. werden Ihnen zu diesem Zeitpunkt Fehler in der Datei angezeigt. Wenn Sie mit dem „Explorer“ in den Ordner wechseln, sehen Sie, dass .class Dateien erzeugt wurden.

Um nun eine Datei auszuführen, geben Sie in der Eingabeaufforderung „java“ gefolgt von dem Dateinamen ohne Endung ein.

Beispiel: Um eine Datei „Beispiel.java“ zu kompilieren und auszuführen, müssen Sie nacheinander folgende Befehle eingeben:

```
javac *.java
java Beispiel
```

1.8 Das erste Java-Programm: Hello World!

In diesem Teil des Tutorial geht es um das Erstellen Ihres ersten Java-Programms, welches die Ausgabe »Hello World!« auf dem Bildschirm bzw. auf der Konsole erzeugen soll. Erstellen Sie hierzu die Datei »HelloWorld.java« und öffnen Sie diese.

Ein Java-Programm besteht immer aus einer oder mehreren Klassen, wobei jede Klasse in einer einzelnen Datei gespeichert wird. Da der Dateiname dem Klassennamen entsprechen muss, muss die Klasse in der soeben erstellten Datei »HelloWorld« heißen. Hierfür schreiben Sie den folgenden Java-Code in die Datei:

```
1
2 public class HelloWorld {
3
4 }
```

Dieser Code definiert den Inhalt der Datei mit »class« als Klasse, der Klassenname lautet »HelloWorld«. In den folgenden geschweiften Klammern folgt nun der Inhalt der Klasse bzw. des Programms. Eine Klasse, die von der Konsole aus gestartet werden soll, benötigt eine Einstiegsmethode, die automatisch aufgerufen wird und in der steht, was als Nächstes durchgeführt werden soll. Dies ist die »main«-Methode, der benötigte Code wurde eingefügt:

```
1
2 public class HelloWorld {
3
4     public static void main(String[] args) {
5
```

```
6     }  
7  
8 }
```

public, **static** und **void** sind Schlüsselworte, auf die erst in den kommenden Wochen eingegangen werden soll. Hinter dem Bezeichner der Methode (main) folgt in runden Klammern, welche Argumente an die Methode übergeben werden sollen (String[] args), dies kann dazu dienen weitere Argumente, die beim Starten des Programms in der Konsole übergeben worden sind, zu übertragen. In den nächsten Wochen wird auch hierauf noch näher eingegangen. Die Definition der »main«-Methode wird ebenfalls von geschweiften Klammern gefolgt, in denen geschrieben steht, was nach Aufruf der Methode ausgeführt wird.

Nun müssen noch die tatsächliche Ausgabe auf der Konsole realisiert werden, was in der »main«-Methode geschieht. Mit System.out.println(); kann eine neue Textzeile auf den Bildschirm ausgegeben werden, mit System.out.print(); wird an der Stelle, an der sich der Cursor in der Konsole derzeit befindet, weitergeschrieben. Der auszugebende Text wird in Anführungszeichen in die runden Klammern geschrieben.

Hier ist nun der gesamte Programmcode notiert:

```
1  
2 public class HelloWorld {  
3  
4     public static void main(String[] args) {  
5  
6         System.out.println("Hello World!");  
7  
8     }  
9  
10 }
```

Testen Sie nun ihr Programm, indem Sie mit der Konsole in das Speicherverzeichnis der Datei wechseln, es mit »javac HelloWorld.java« kompilieren und mit »java HelloWorld« ausführen. Wenn Sie alles richtig gemacht haben, sollte nun »Hello World!« auf der Konsole erscheinen.