

# Die Windows-Konsole

Dirk Louis

Peter Müller

# 1 Inhaltsverzeichnis

Die Windows-Konsole .....	1
1 Inhaltsverzeichnis .....	2
2 Die Windows-Konsole .....	3
2.1 Konsolenprogramme .....	3
2.1.1 Einsatzbereiche für Konsolenprogramme .....	3
2.1.2 Konsolenprogramme ausführen .....	4
2.2 Blick zurück in Wehmut – die Konsole früher und heute .....	4
2.2.1 Ein Konsolenfenster öffnen .....	5
2.3 Arbeiten mit der Konsole .....	6
2.3.1 Die wichtigsten Befehle .....	6
2.3.2 Die history .....	8
2.3.3 Programme starten .....	8
2.3.4 Batch-Dateien .....	10

## 2 Die Windows-Konsole

Viele Anwender – insbesondere jüngere Anwender, die mit Windows 98, NT oder XP groß geworden sind – erwarten von einem Programm, dass es über ein Fenster mit Menüleiste, Werkzeugleisten, Arbeitsfläche, Bildlaufleisten, Statusleiste und anderen Oberflächenelementen verfügt. Mit anderen Worten: sie denken bei Programmen an den Windows Explorer, den Netscape Navigator, an WinZip, RealPlayer, Paint, Word oder wie sie alle heißen.

So ist die Konsole, unter Windows meist **Eingabeaufforderung** genannt, weitgehend in Vergessenheit geraten. Sicherlich nicht ganz zu Unrecht, denn die meisten Aufgaben, für die man früher die Konsole benötigte, können heute bequemer und schneller im Windows Explorer erledigt werden.

Trotzdem hat die Konsole immer noch ihre Berechtigung und wird weiterhin zusammen mit den Windows-Betriebssystemen ausgeliefert. Nicht als Konkurrenz zum Windows Explorer – zum Kopieren, Verschieben oder Löschen von Dateien greift heutzutage wohl niemand mehr auf die Konsole zurück –, sondern zur Ausführung von Programmen, die über keine Fenster, keine grafische Benutzeroberfläche und keinen Sinn für Windowing verfügen. Solche Programme werden als **Konsolenprogramme** bezeichnet.

### 2.1 Konsolenprogramme

Heutzutage gibt es für Konsolenprogramme vor allem drei Einsatzbereiche.

#### 2.1.1 Einsatzbereiche für Konsolenprogramme

- Einfache Dienstprogramme, die nach dem Start automatisch – ohne auf weitere Eingaben seitens des Benutzer angewiesen zu sein – ablaufen. Typische Vertreter sind z.B. Compiler, Interpreter, Webserver, kleinere Hilfsprogramme. (Die Entwicklung solcher Dienstprogramme ist allerdings unter Unix/Linux traditionell wesentlich weiter verbreitet als unter Windows.)
- Kleinere Demoprogramme zum Erlernen einer Programmiersprache. Um GUI-Programme (Programme mit grafischer Benutzeroberfläche) schreiben zu können, bedarf es profunder Kenntnisse sowohl in der verwendeten Programmiersprache als auch in den Techniken der GUI-Programmierung (für die zumeist umfangreiche Bibliotheken eingesetzt werden). Für den Einstieg in die Programmierung ist es daher meist besser, Konsolenprogramme zu erstellen. Der Anfänger

kann sich dann ganz auf die Syntax der Sprache konzentrieren. Hat er sich mit der Programmiersprache vertraut gemacht, kann er sich in einem zweiten Schritt in die Programmierung von GUI-Anwendungen einarbeiten.

- Spieleprogramme. Diese wurden lange noch nach Einführung von Windows als DOS-Programme (in speziellem Grafikmodus) programmiert, da dies den Spieleentwicklern den Einsatz effizienterer Techniken und schnellerer, grafischer Animationen ermöglichte. Mittlerweile haben aber neue leistungsfähige Grafikkibliotheken für Win32 (Win95 und aufwärts) wie z.B. DirectX oder OpenGL sowie die immer schneller werdenden Rechner und Grafikausteile dazu beigetragen, dass sich die Spiele-Software-Industrie umgestellt hat.

### 2.1.2 Konsolenprogramme ausführen

Konsolenprogramme können unter Windows auf zweierlei Weise ausgeführt werden:

- Sie doppelklicken im Windows Explorer auf die EXE-Datei des Programms. Windows öffnet daraufhin ein Konsolenfenster und führt in dessen Umgebung das Konsolenprogramm aus. Im Konsolenfenster sieht der Anwender sowohl die Ausgaben des Programms als auch die Eingaben, die er über die Tastatur an das Programm schickt. Der große Nachteil dieses Verfahrens ist, dass das Konsolenfenster von Windows automatisch geschlossen wird, sowie das Programm beendet wurde. (Für kürzere Programme geht das Ganze meist so schnell, dass der Anwender das Konsolenfenster nur noch kurz aufflackern sieht.)
- Sie öffnen zuerst ein Konsolenfenster und rufen aus diesem dann das Programm aus. Auch hier dient das Konsolenfenster dem Programm sowohl zur Anzeige der Ausgabe als auch der Eingabe. Nach Beendigung des Programms bleibt das Konsolenfenster aber weiterhin geöffnet und der Anwender kann in Ruhe die letzten Ausgaben des Programms lesen, das Programm erneut starten, andere Befehle ausführen.

## 2.2 Blick zurück in Wehmut – die Konsole früher und heute

Früher war der Computer selbst die Konsole. Die Anwender saßen vor einem schwarzen Bildschirm (ganz früher noch mit grüner oder brauner Schrift) und sahen dort eine Eingabeaufforderung, ein einzelnes Zeichen (z.B. ein >), kurz **Prompt** genannt. Über die Tastatur konnte der Anwender Befehle eintippen, die nach dem Abschicken mit der (**Return**)-Taste vom Computer ausgeführt wurden. Neben allgemeinen Betriebssystembefehlen,

wie dem Wechseln des aktuellen Verzeichnisses oder dem Löschen einer Datei konnte man von der Konsole aus auch Programme aufrufen. Diese übernahmen dann die Konsole und nutzten sie, um Ergebnisse auszugeben oder Daten vom Anwender abzufragen.

Seitdem hat sich einiges geändert und die Anwender sind es gewohnt, auf den PCs eine grafische Windows-Oberfläche vorzufinden. Die Konsole wurde in das Start-Menü verbannt.

### 2.2.1 Ein Konsolenfenster öffnen

Konsolenfenster können Sie unter Windows über das Start-Menü aufrufen. Die genauen Aufrufpfade variieren von Windows-Version zu Windows-Version:

#### Windows 98

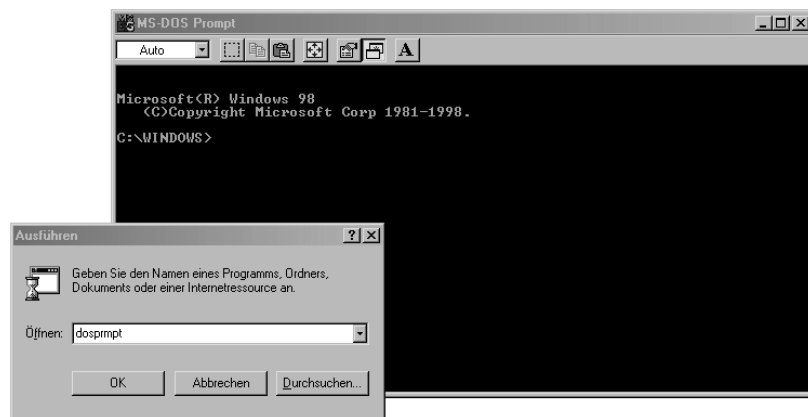


Abbildung 2.1: Konsole unter Win98 aufrufen (über Ausführen-Dialog)

Wählen Sie im *Start*-Menü den Befehl *Ausführen* aus. Geben Sie `dosprmt` ein und schicken Sie das Dialogfeld ab.

Alternativ können Sie die Konsole über *Start/Programme/MS-DOS-Eingabeaufforderung* aufrufen.

## Windows 2000 und Windows XP

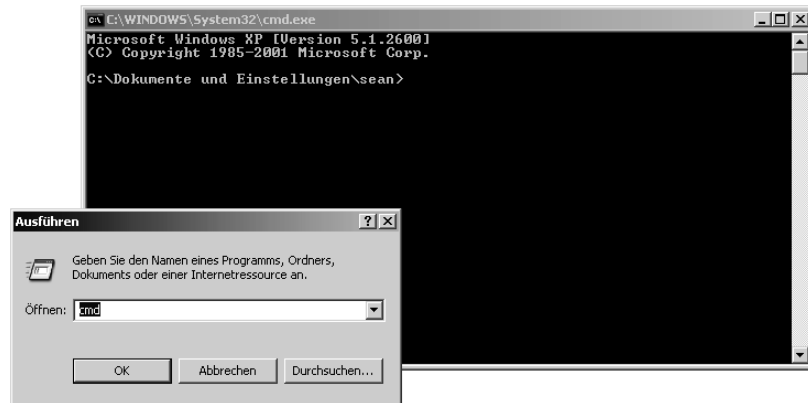


Abbildung 2.2: Konsole unter Windows XP aufrufen (über Ausführen-Dialog)

Wählen Sie im *Start*-Menü den Befehl *Ausführen* aus. Geben Sie *cmd* ein und schicken Sie das Dialogfeld ab.

Alternativ können Sie die Konsole über *Start/Alle Programme/Zubehör/Eingabeaufforderung* aufrufen.

## 2.3 Arbeiten mit der Konsole

Für den Nicht-Eingeweihten kann die Arbeit mit der Konsole schnell zur Qual werden. Wer jedoch die wichtigsten Befehle und Tricks zur Arbeitserleichterung kennt, wird sich mit ihr schnell anfreunden.

### 2.3.1 Die wichtigsten Befehle

Die beiden wichtigsten Aufgaben der Konsole sind die Ausführung von Programmen oder Systembefehlen und das Navigieren in der Verzeichnisstruktur.

#### Navigieren

Wenn Sie ein neues Konsolenfenster öffnen, beginnen Sie je nach Windows-Version entweder im Windows-Verzeichnis (Windows 98) oder in Ihrem persönlichen Homeverzeichnis (Windows XP). Das Verzeichnis, in dem Sie sich befinden, wird im Prompt angezeigt:

C:\Windows> für Windows 95

C:\Dokumente und Einstellungen\Benutzer> für Windows XP

Mit den folgenden Befehlen können Sie von hier aus in andere Verzeichnisse wechseln:

Befehl	Beschreibung
<code>cd unterverzeichnis</code>	Wechselt in das angegebene Unterverzeichnis (vorausgesetzt, dass aktuelle Verzeichnis enthält ein entsprechendes Unterverzeichnis)
<code>cd ..</code>	Wechselt in das übergeordnete Verzeichnis.
<code>D:</code>	Wechselt in ein anderes Laufwerk (hier D)
<code>cd c:\verz1\verz2</code>	Wechselt in das angegebene Verzeichnis

Tabelle 2.1: Navigationsbefehle

### Verzeichnisbefehle

Befehl	Beschreibung
<code>dir</code>	Listet alle Unterverzeichnisse und Dateien im aktuellen Verzeichnis auf.
<code>dir /p</code>	Wie oben, die Auflistung geschieht aber seitenweise. (Durch Drücken einer Taste können Sie die jeweils nächste Seite anzeigen lassen.)
<code>dir *.java</code>	Listet alle Dateien im aktuellen Verzeichnis auf, die die Extension <code>.java</code> haben.
<code>copy datei1.txt datei2.txt</code>	Erzeugt eine Kopie namens <code>datei2.txt</code> von <code>datei1.txt</code> .
<code>copy *.* unterverzeichnis</code>	Kopiert alle Dateien aus dem aktuellen Verzeichnis in das angegebene Unterverzeichnis
<code>del datei1.txt</code>	Löscht die <code>datei1.txt</code> .
<code>rename datei1.txt datei2.txt</code>	Ändert den Namen von <code>datei1.txt</code> in <code>datei2.txt</code> .
<code>mkdir name</code>	Erzeugt ein Unterverzeichnis <code>name</code>
<code>rmdir name</code>	Löscht ein Unterverzeichnis <code>name</code>

Tabelle 2.2: Verzeichnisbefehle

### Auswahl sonstiger Befehle

Befehl	Beschreibung
<code>(Strg)+C</code>	Bricht den aktuellen Befehl ab. (Geeignet beispielsweise zum Vorzeitigen Beenden von <code>dir /p</code> -Befehlen oder Programmen.)

(ESC)	Löscht die Kommandozeile
(Rücktaste)	Löscht das vorhergehende Zeichen.
(Einfg)	Schaltet den Einfüge-Modus ein und aus
cls	Löscht die Konsole und zeigt einen Prompt an.
doskey	Schaltet die history ein.
edit	Einfachen Konsoleneditor öffnen.
path	Zeigt den eingestellten Systempfad an.
set	Listet alle Umgebungsvariablen auf.
set name=wert	Richtet eine Umgebungsvariable ein.
set name=	Löscht eine Umgebungsvariable.
ver	Zeigt die Windows-Version an.

*Tabelle 2.3: Sonstige Befehle*

### 2.3.2 Die history

Unter Windows 2000 und Windows XP zeichnet die Konsole alle von Ihnen abgeschickten Befehle auf. Mit Hilfe der **(Pfeil oben)**- und **(Pfeil unten)**-Tasten können Sie die Liste der eingegebenen Befehle (die so genannte "history") durchgehen und so längere Befehle bequem auswählen und erneut abschicken.

Unter Windows 98 wird standardmäßig keine history aufgezeichnet. Sie können diese äußerst angenehme Option aber einschalten.

Um die history für die aktuelle Sitzung mit der Konsole einzuschalten, geben Sie einfach den Befehl `doskey` ein.

Um die history für alle zukünftigen Sitzungen einzuschalten, laden Sie die Datei `C:\autoexec.bat` in einen Editor und fügen Sie eine neue Zeile

```
doskey
```

ein.

### 2.3.3 Programme starten

Um ein Programm von der Konsole aus zu starten, gibt es drei Möglichkeiten:

- Sie wechseln in der Konsole zu dem Verzeichnis, in dem die Programmdatei steht, geben hinter dem Prompt den Programmnamen und etwaige Aufrufparameter ein und schicken ab.



```
C:\Windows> cd c:\Meine Programme
```

```
C:\Meine Programme> weckmich 7:15
```

- Wenn das Programm eine Datei verarbeitet, ist es vielleicht einfacher in das Verzeichnis der Datei zu wechseln und das Programm unter Angabe des vollständigen Befehlspfades aufzurufen:

```
C:\Windows> cd c:\Meine Dateien
```

```
C:\Meine Dateien> c:\Meine Programme\MeinEditor
demo.txt
```

- Wenn Sie ein Programm von jedem beliebigen Verzeichnis aus ohne Angabe des vollständigen Pfades aufrufen wollen, müssen Sie das Programm in den Systempfad (Umgebungsvariable PATH) eintragen.

```
C:\Windows> cd c:\Meine Dateien
```

```
C:\Meine Dateien> MeinEditor demo.txt
```

#### Verzeichnis in Systempfad eintragen

Um ein Programm aus jedem Verzeichnis der Konsole heraus nur über den Programmnamen aufrufen zu können, müssen Sie das Verzeichnis, in dem die Programmdatei steht, in den Systempfad (Umgebungsvariable PATH) eintragen. Für den Fall, dass das Programm in *C:\Programme* steht, gehen Sie dazu wie folgt vor:

- Windows 95/98: Legen Sie zur Sicherheit eine Kopie der Datei *c:\autoexec.bat* an und laden Sie die Datei danach in einen Editor (beispielsweise Notepad, Aufruf über **START/PROGRAMME/ZUBEHÖR/EDITOR** oder über **START/AUSFÜHREN, NOTEPAD** eingeben und abschicken). Suchen Sie nach einem **PATH**-Eintrag und fügen Sie das Verzeichnis *C:\Programme* hinzu. Zum Beispiel:

alter Eintrag: `SET PATH=.;c:\dos;`

neuer Eintrag: `SET PATH=.;c:\dos;c:\programme;`

Das Semikolon dient zur Trennung der einzelnen Verzeichnisangaben in PATH.

Wenn Sie gar keine PATH-Angabe finden, dann fügen Sie eine neue PATH-Anweisung hinzu, z.B.:

`SET PATH=c:\programme;`

- Windows ME/2000/XP: Die Umgebungsvariable **PATH** wird über den Dialog der Systemeigenschaften verwaltet. Der Weg dorthin ist lang und von Betriebssystem zu Betriebssystem verschieden. Unter Windows Me lautet er

START/PROGRAMME/ZUBEHÖR/SYSTEMTOOLS/SYSTEMINFORMATIONEN, Menü TOOLS/SYSTEM KONFIGURATION, Register UMGEBUNG.

Unter Windows 2000 und Windows XP rufen Sie über **START** oder **START/EINSTELLUNGEN** die Systemsteuerung auf und gelangen via **SYSTEM**, Register **ERWEITERT**, Schalter **UMGEBUNGSVARIABLEN** zum Ziel. Danach können Sie die Systemvariable auswählen, zum Bearbeiten laden und den Verzeichnispfad mit Semikolon anhängen.

Um den neuen Pfad zu testen, müssen Sie den Rechner neu starten.

### 2.3.4 Batch-Dateien

Alle Befehle, die Sie direkt über die Konsole ausführen, können Sie auch in Batch-Dateien speichern und durch Aufruf der Batch-Datei ausführen. Dies lohnt sich beispielsweise für häufig benötigte, komplexe Befehle sowie immer gleiche Abfolgen mehrerer Befehle.

Angenommen Sie nutzen die Konsole, um Java-Quelltexte zu kompilieren. Ihre Java-Quelltextdateien stehen alle in Unterverzeichnissen des Ordners *C:\Eigene Dateien\MeineProjekte\Java*.

Um schneller nach dem Öffnen eines Konsolenfenster in dieses Verzeichnis zu springen, gehen Sie wie folgt vor:

1. Öffnen Sie die Konsole. Sie befinden sich im Windows- oder Ihrem Homeverzeichnis.
2. Geben Sie `edit` ein. Der DOS-Editor wird gestartet.
3. Tippen Sie den `cd`-Befehl zum Wechseln in das gewünschte Verzeichnis ein. (Wenn Sie noch weitere Befehle ausführen lassen möchten, beginnen Sie für jeden Befehl eine eigene Zeile.)

**Achtung!** Unter Windows 98 dürfen die Verzeichnisangaben in den Batch-Dateien keine Leerzeichen enthalten. Verwenden Sie für diese Verzeichnisse, die 8-Zeichen-Abkürzung mit Tilde. (Sie können sich die Abkürzung anzeigen lassen, indem Sie in einem Konsolenfenster zu dem übergeordneten Verzeichnis gehen und dann `dir` oder `dir /p` eingeben.)

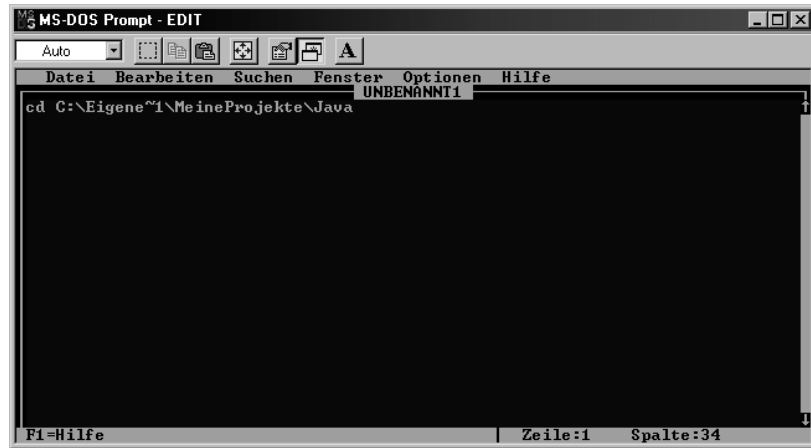


Abbildung 2.3: Anlegen einer Batch-Datei

4. Speichern Sie die Datei unter einem kurzen Namen, den Sie sich gut merken können, und mit der Extension `.bat` – beispielsweise `goJava.bat`.
5. Beenden Sie den Editor, um zurück zum Prompt zu gelangen.
6. Führen Sie die Batch-Datei aus, indem Sie `goJava` eintippen und abschicken.