

# Beschreibung des Programms

## **CODING1**

Programm zur Ver- / Entschlüsselung „on place“

## User Manual

Dipl.-Math. Jürgen Müller

Version 1.0 / Stand: 31.08.2005

## 1 Beschreibung der notwendigen Parameter des Programms CODING1

### 1.1 Anstarten

Sie können das Programm CODING1.EXE direkt aus MS-Windows heraus aufrufen (z.B. durch Doppel-Klick), worauf das Programm innerhalb einer DOS-Box gestartet wird.

### 1.2 Sprache

Zunächst werden Sie aufgefordert anzugeben, in welcher Sprache Sie den nachfolgenden Dialog wünschen:

```
Language / Sprache:
  english  --> 0
  deutsch  --> 1
```

Geben Sie die Ziffer ein, dessen zugehörige Sprache von Ihnen gewünscht wird.

### 1.3 Identifikation

Es folgt die Anzeige der Identifikationsseite von CODING1:

```
01111010101101101110000000001010001011011001000011110000110010101100101001000100
1..... * .. 0
0. . CODING1 * .. 1
1 .. CODING1 .. 1
1 . . CODING1 .. 0
0 .. CODING1 .. 0
0 .. CODING1 .. 1
1 .. CODING1 .. 0
0. . CODING1 .. 1
1 .. CODING1 .. 1
1 .. CODING1 .. 0
0 Programm zur direkten Ver-/Entschlüsselung von Dateien (1-Byte-Gruppen) .. 0
0 ... von Jürgen Müller .. 0
0 .. SYSJM Jürgen Müller .. 1
1 .. 6/2005 .. 1
1 .. Version 1.0 .. 1
0 .. nur für den privaten Gebrauch <--- .. 0
1 .. * Alle Rechte vorbehalten. Programm-Benutzung auf eigene Verantwortung. .. 1
1111001111011011010011011111010010110110010000011011011010110010011111100000010
```

### 1.4 Einschränkungen

Bitte beachten Sie, dass die vorliegende Version nicht zum kommerziellen Gebrauch berechtigt. Falls Sie dieses Programm (oder eines der Programme für höherer Sicherheitsanforderungen) für den kommerziellen Gebrauch verwenden wollen, wenden Sie sich bitte über eine der unten angegebenen Kontaktdaten an den Autor. Vielen Dank.

Bitte beachten Sie darüber hinaus, dass trotz höchster Qualitätssicherheit für das Programm keine absolute Garantie übernommen werden kann. Falls wider Erwarten Probleme auftreten sollten, die nachweislich nicht durch falsche Bedienung verursacht werden, wird sich der Autor umgehend um eine Problem-Behebung bemühen. Natürlich sind auch jedwede Anregungen Ihrerseits zu Programm, Dokumentation usw. jeder Zeit willkommen.

Für das hier verwendete Verfahren (siehe Kurzexposé) gibt es **keinerlei** Möglichkeit, einen verlorenen Schlüssel wieder zu beschaffen, auch für den Autor nicht (!). Bitte arbeiten Sie im Zweifelsfall vorsichtshalber mit Schlüsseldateien (siehe unten), um solche Probleme zu vermeiden. Vielen Dank für Ihr Verständnis.

Jede **einzelne Datei** muss **kleiner als 4 GB** sein, um ordnungsgemäß bearbeitet zu werden.

## 1.5 Allgemeines

Sie können das Programm nach dieser Eingangsmaske jeder Zeit während der Eingabe von Programm-Parametern entweder durch den Wert '9' bzw. ohne Eingabe weiterer Zeichen durch betätigen der RETURN(↵)-Taste ('Leereingabe') beenden, ohne dass eine Datei verschlüsselt oder entschlüsselt worden wäre.

Bei Überladeinformationen (siehe unten) können auch die Werte '0' und '-1' im Einzelfall dazu dienen.

Schließen Sie jede Eingabe wie gewohnt mit der RETURN(↵)-Taste ab.

## 1.6 Kontrollausgabedatei

Nach kurzer Anzeige der Identifikationsseite wird die Frage nach einer Datei für die Kontrollausgabe gestellt:

```
Geben Sie bitte den Namen der Kontrollausgabedatei an
(Leereingabe = Programmabbruch):
```

Geben Sie hier einen Dateinamen ggf. mit Pfad-Angabe für eine neu anzulegende oder zu überschreibende Datei an, in die alle relevanten Informationen (außer natürlich sicherheitsrelevante Informationen !) zu von Ihnen gewählten weiteren Parametern bzw. zu vom Programm erledigten Aufgaben protokolliert werden.

Das Programm arbeitet mit dem OEM-Zeichensatz (MS-DOS-Zeichensatz), nicht mit dem ANSI-Zeichensatz (MS-Windows), was beim späteren Lesen der Kontrollausgabe-Datei zu berücksichtigen ist.

## 1.7 Ausgabe der Kommandozeilen-Steuerungs-Parameter

Nun können Sie sich eine detaillierte Beschreibung der Steuerungs-Parameter für den Kommandozeilen-Aufruf des Programms in diese Kontrolldatei ausgeben lassen:

```
Soll eine Parameter-Beschreibung des Programms
in diese Kontrollausgabedatei ausgegeben werden ?
0   - Nein, das Programm soll fortgesetzt werden,
1   - Ja, danach soll das Programm fortgesetzt werden,
2   - Ja, danach soll das Programm beendet werden.
Geben Sie bitte die zugehörige Zahl an:
```

Dies geschieht mit den Werten '1' bzw. '2', der Wert '0' übergeht diese Ausgabe. Für die Werte '0' bzw. '1' wird das Programm danach fortgesetzt.

## 1.8 Angaben zu den zu behandelnden Dateien

Haben Sie das Programm fortgesetzt, werden Sie gefragt, in welcher Form Sie die Datei(en) angeben wollen, die verschlüsselt oder entschlüsselt werden soll(en):

```
Welche Dateien sollen bearbeitet werden?');
1  eine Datei bzw. bei "Wild-Card"-Angaben alle angesprochenen');
   Dateien, deren Name nachfolgend angegeben wird');
2  alle Dateien (ggf. mit "Wild-Card"-Angaben), deren Namen in der');
   nachfolgend angegebenen Datei stehen');
3  alle Dateien des nachfolgend angegebenen Verzeichnisses');
4  alle Dateien des nachfolgend angegebenen Verzeichnisses u n d');
   aller Unterverzeichnisse darunter liegender Stufen');
9  Programmabbruch.
Geben Sie bitte die zugehörige Zahl an:
```

Durch Eingabe des Wertes '1' können Sie die Datei oder die Dateien, die nachfolgend behandelt werden sollen, ggf. mit Pfad-Angabe direkt angeben, wobei Sie die von MS-DOS gewohnten Sonderzeichen '?' für ein beliebiges Zeichen und '\*' für eine Gruppe beliebiger Zeichen im Dateinamen angeben können („Wild-Card-Angabe“), um mehrere Dateien auszuwählen.

Mit Eingabe des Wertes '2' teilen Sie dem Programm mit, dass Sie nachfolgend eine Datei angeben möchten, die selbst jeweils pro Dateizeile eine Dateiangabe beinhaltet, die zu behandelnde Dateien identifiziert. D.h. Sie haben zuvor eine Datei mit Dateinamen bzw. Dateiangaben der Dateien vorbereitet, die im vorliegenden Programmablauf behandelt werden sollen.

In vielen Fällen wünschen Sie, alle Dateien eines Verzeichnisses oder gar eines Verzeichnisses **u n d** sämtlicher Unterverzeichnisse beliebiger Stufe zu bearbeiten. Dies ist durch Angabe der Werte '3' für die Dateien genau eines Verzeichnisses und '4' für alle Dateien innerhalb eines Verzeichnisbaums möglich. In diesen Fällen wird Sie das Programm nachfolgend nach dem Verzeichnisnamen fragen, dem Sie natürlich eine Pfadangabe voranstellen können, wie Sie das auch von MS-DOS her kennen.

Je nach Eingabewert werden Sie daher wie folgt aufgefordert:

'1':

Geben Sie bitte die Dateiidentifikation der zu bearbeitenden Datei(en) an (Leereingabe = Programmabbruch):

'2':

Geben Sie bitte den Dateinamen der Datei an, in der die Namen der zu bearbeitenden Dateien enthalten sind (Leereingabe = Programmabbruch):

'3', '4':

Geben Sie bitte den Namen des Verzeichnisses an, in dem die zu bearbeitenden Dateien aufgeführt sind (Leereingabe = Programmabbruch):

## 1.9 Bearbeitungsform

Damit steht zwar fest, welche Dateien potentiell in die Bearbeitung mit einbezogen werden, aber es bleibt noch die Frage, in welcher Form dies geschehen soll:

Wie soll(en) die Datei(en) bearbeitet werden?

- 1 verschlüsseln der Datei(en) einzeln
- 2 entschlüsseln der Datei(en) einzeln
- 3 verschlüsseln der Datei(en) als e i n e Datenmenge
- 4 entschlüsseln der Datei(en) als e i n e Datenmenge
- 9 Programmabbruch.

Geben Sie bitte die zugehörige Zahl an:

Die Begriffe „verschlüsseln“ und „entschlüsseln“ verstehen sich noch von selbst (obwohl diese Operationen auch vertauscht angewendet werden können (!)). Für die Sicherheit relevant verbleibt aber die Frage, was z.B. unter verschlüsseln „als eine Datenmenge“ zu verstehen ist.

Zunächst können einzeln verschlüsselte Dateien auch einzeln, d.h. unabhängig von anderen Dateien wieder entschlüsselt werden. Dies ist bei der Verschlüsselung der Dateien als einer Datenmenge nicht möglich (!). Hier können nur genau die Dateien, die ursprünglich als eine Datenmenge verschlüsselt wurden, wieder gemeinsam entschlüsselt werden.

Der Vorteil der Verschlüsselung als einer Datenmenge ist, dass die Programm-intern verwendeten Schlüssel-Parameter nicht für jede Datei wieder auf den Ausgangszustand zurückgesetzt werden. Vielmehr werden alle beteiligten Dateien als eine große Datei behandelt, womit Entschlüsselungsversuche unberechtigter Dritter nochmals erheblich erschwert werden.

Als Nachteil bleibt, dass so verschlüsselte Verzeichnisse **n i c h t** modifiziert werden dürfen bzw. die Veränderungen müssen vor der Entschlüsselung exakt wieder in den Ausgangszustand überführt werden. Dies kann z.B. auch dadurch geschehen, dass man sämtliche bei der Verschlüsselung beteiligten Dateien in der Reihenfolge ihrer Verschlüsselung (siehe hierzu z.B. die Verschlüsselungs-Protokolldatei) in eine Datei einträgt und diese als Datei der zu behandelnden Dateien (siehe oben) für die Entschlüsselung angibt. Das vorliegende Programm arbeitet also nur auf Dateiinhalten, modifiziert also keine Unterverzeichnis-Informationen oder ähnliches.

## 1.10 Schlüssel-Informationen

Die nun folgenden Informationen drehen sich sämtlich um alles, was mit dem Schlüssel zu tun hat:

### 1.10.1 Zeichensatz

Mit welchem Zeichensatz möchten Sie den Schlüssel eingeben?

- 1 ASCII-Zeichen (max. 256 Zeichen)
- 2 hexadezimale Zeichen  
(max. 512 Zeichen "0" bis "9" bzw. "A" bis "F")
- 9 Programmabbruch.

Geben Sie bitte die zugehörige Zahl an:

Hierbei geht es um den Zeichensatz, mit dem Schlüsselinformationen eingegeben bzw. eingelesen werden sollen. Um alle möglichen Zeichen an- und eingebbar zu machen, kann neben dem „normalen“ Zeichensatz (ASCII-Zeichen) für Zeichen, die z.B. durch die Tastatur eingegeben werden können, auch ein Zeichensatz gewählt werden (hexadezimale Zeichen), bei dem jeweils zwei Zeichen vom Programm zu einem Schlüsselzeichen zusammengesetzt werden. Damit lassen sich auch Schlüsselzeichen erzeugen, die mit der Tastatur (d.h. direkt) ggf. nicht oder nur sehr mühsam erzeugbar sind. Darin nicht bewanderte Benutzer sollten darauf ggf. zunächst verzichten.

### 1.10.2 Eingabe-Modus

In welchem Eingabe-Modus möchten Sie die Schlüssel-Zeichen eingeben?

- 0 zweimal über Tastatur o h n e Echo auf dem Bildschirm  
(das 2. Mal zur Kontrolle!)
- 1 einmal über die Tastatur m i t Bildschirm-Kontrolle
- 2 über eine Datei, die den Schlüssel enthält
- 9 Programmabbruch.

Geben Sie bitte die zugehörige Zahl an:

### 1.10.3 Tastatur-Eingabe

Falls Sie die Schlüsselzeichen über die Tastatur eingeben wollen, wird Ihnen mit (Modus='1'):

Geben Sie bitte Ihren Schlüssel in hexadezimalen /ASCII-Zeichen  
(Schlüssel-Ende: Leerzeile) ein:

bzw. mit (Modus='0'):

Geben Sie bitte Ihren Schlüssel in hexadezimalen /ASCII-Zeichen  
(Schlüssel-Ende: Leerzeile) das 1. Mal ein:

und mit:

Geben Sie bitte Ihren Schlüssel in hexadezimalen /ASCII-Zeichen  
(Schlüssel-Ende: Leerzeile) das 2. Mal ein (Kontroll-Eingabe):

die Gelegenheit gegeben.

### 1.10.4 Eingabe via Schlüssel-Datei

Falls Sie eine Schlüsseldatei verwenden wollen (Modus='2'), werden Sie mit:

Wie möchten Sie den Namen der Datei, die den Schlüssel enthält, eingeben?

- 0 zweimal über Tastatur o h n e Echo auf dem Bildschirm  
(das 2. Mal zur Kontrolle!)
- 1 einmal über Tastatur m i t Bildschirm-Kontrolle
- 9 Programmabbruch.

Geben Sie bitte die zugehörige Zahl an:

zunächst nach der Art der Eingabe des Dateinamens der Schlüsseldatei gefragt. Danach wird mit:

Geben Sie bitte den Namen der Datei an, die den Schlüssel enthält  
(Leereingabe = Programmabbruch):

bzw. mit:

Geben Sie bitte den Namen der Datei, die den Schlüssel enthält,  
das 1. Mal an (Leereingabe = Programmabbruch):

und mit:

Geben Sie bitte den Namen der Datei, die den Schlüssel enthält,  
das 2. Mal an (Kontroll-Eingabe):

der Dateiname erfragt.

### 1.10.5 Schlüsselzeichen-Bearbeitung

Verwenden Sie eine Schlüsseldatei, wenn Sie einen komplexen Schlüssel oder einen "Zufalls-Schlüssel" verwenden wollen oder müssen. Eine im aktuellen Programmlauf verwendete Schlüsseldatei wird in diesem Programmlauf **n i c h t** behandelt (also auch nicht zerstört), selbst wenn diese explizit oder implizit als zu behandelnde Datei angegeben wurde (!). Das gilt auch für die Überladedatei (siehe unten), die Kontrollausgabe-Datei, die Datei der Dateinamen sowie die Programmdatei selbst (siehe jeweils oben), falls diese Dateien tatsächlich verwendet werden.

Prinzipiell werden alle Sonderzeichen außer Zeilenabschlusszeichen (meist durch RETURN( $\zeta$ )-Taste erzeugt bzw. <carriage return><line feed> bzw. 0D<sub>hex</sub>- plus 0A<sub>hex</sub>- Zeichen) als Bestandteil eines Schlüssels aus ASCII-Zeichen aufgefasst. Zeilenabschlusszeichen dürfen auch bei der Schlüsseingabe durch hexadezimale Zeichen beliebig verwendet werden, um z.B. Schlüsselinformationen auf einzelne Zeilen aufzuteilen. Zeilenabschlusszeichen werden bei der Schlüsseingabe allgemein ignoriert bzw. bei Tastatur-Eingabe nur in Form einer zusätzlichen Leerzeile als Schlüsselende interpretiert.

Umfasst ein Schlüssel mehr als 256 Schlüsselzeichen, so wird der Rest der Zeichen ignoriert. Werden hingegen weniger als 256 Schlüsselzeichen angegeben, so werden die angegebenen Schlüsselzeichen so oft wiederholt und ggf. am Ende abgeschnitten, bis genau 256 Schlüsselzeichen für die weitere Verarbeitung vorliegen. Trotzdem sollten nicht zu wenige und nicht zu homogene Zeichen als Schlüsselzeichen angegeben werden, da dies im Extremfall (z.B. nur hexadezimale Null-Zeichen) zu keiner Verschlüsselung führen könnte (!).

Bis auf diese wenigen Extremfälle ist dem Verschlüsselungsergebnis auch bei kurzen Schlüsseln nicht in irgend einer Form anzusehen, ob der Ausgangsschlüssel ein kurzer oder langer Schlüssel war (!). Lediglich beim Suchen nach dem Schlüssel wird ein Dritter wahrscheinlich zunächst kürzere Schlüssel ausprobieren, bevor er längere Eingabeschlüssel in Betracht zieht - und dann wohl aufgibt.

### 1.11 Eigentliche Programm-Bearbeitung

Falls die Frage nach sogenannten „Überladeinformationen“ (siehe unten) mit dem Wert '0' beantwortet wird, so startet jetzt das Programm mit der eigentlichen Bearbeitung (z.B.):

```
Beginn der Bearbeitung: Do. 31.03.2005, 19:26:33.
```

Damit ist das Programm **n i c h t** mehr unterbrechbar (!). Da alle Dateien direkt bearbeitet werden (es werden keinerlei Kopien erstellt), würde eine Unterbrechung ggf. eine oder auch mehrere Dateien **unwiederbringlich zerstören (!)**, d.h. Sie sollten vorher **unbedingt Sicherungskopien** auf z.B. anderen Speichermedien erstellen bzw. anlegen (!). Diese können Sie bei erfolgreicher Verschlüsselung nochmals durch das Verschlüsselungsergebnis ersetzen, um so Schutz bei Störungen während der Entschlüsselung zu erhalten.

Bei der Bearbeitung werden alle behandelten Dateien und Verzeichnisse jeweils auf dem Bildschirm und in der oben angegebenen Protokolldatei mitprotokolliert, so dass jeder Schritt aktuell und auch später noch von Ihnen verfolgt werden kann.

Mit einer Kurz-Statistik, dem Ende der Bearbeitung und der Programm-Ende-Zeile (z.B.)

```
212430600 Bytes in      54 Datei(en) aus      9 Verzeichnissen bearbeitet.  
Ende der Bearbeitung: Do. 31.03.2005, 19:28:03.
```

```
Ver-/Entschlüsselungsprogramm C O D I N G 1 1.0 beendet.
```

verabschiedet sich das Programm.

## 2 Beschreibung der Überlade-Informationen des Programms CODING1

Die nachfolgenden Angaben beziehen sich auf sogenannte „Überladeinformationen“, die in den meisten Fällen nicht benötigt werden (Wert '0' unten).

Unter „Überladen“ wird die Möglichkeit verstanden, einen Schlüssel durch gezielte Veränderungen manuell oder auch durch eine zweite Datei zu modifizieren. Die Veränderungen können dabei auf bestimmte Stellen des Schlüssels und sie können auf bestimmte Änderungswerte innerhalb der Überladeinformationen beschränkt werden.

Damit ist es z.B. im Direktmodus möglich, eine zweite Datei, die Überladedatei, dazu zu verwenden, Werte an „zufälligen“ Stellen im bisherigen Schlüssel durch gewisse „zufällige“ Werte der Überladedatei zu überschreiben.

### 2.1 Überladeinformation

Allgemein besteht jede einzelne Überladeinformation aus zwei Angaben:

- der Stellenangabe und
- dem Überladewert.

Im „Direktmodus“ bestehen diese Angaben jeweils aus den Zeichenwerten zweier hintereinander liegender Zeichen innerhalb einer beliebigen Datei bzw. innerhalb einer beliebigen Eingabezeichenkette (Wert '7' unten). In den sonstigen Fällen müssen diese Angaben wie folgt aussehen:

```
Sollen Schlüsselangaben überladen werden?
0  nein, keine Schlüsselangaben-Überladung
1  ja, in der Form "<bytenr-dez>:<zeichen>,"
2  ja, in der Form "<bytenr-dez>:<hexwert>,"
3  ja, in der Form "<bytenr-dez>:<dezwert>,"
4  ja, in der Form "<bytenr-hex>:<zeichen>,"
5  ja, in der Form "<bytenr-hex>:<hexwert>,"
6  ja, in der Form "<bytenr-hex>:<dezwert>,"
7  ja, in Form einer fortlaufenden Zeichenkette
9  Programmabbruch.
```

Geben Sie bitte die zugehörige Zahl an:

Dabei steht

'bytenr-dez/-hex'	für die dezimale bzw. hexadezimale Nummer der angesprochenen Schlüsselstelle zwischen 1 und 256 bzw. zwischen 1 und 100 <sub>hex</sub> ,
'zeichen'	für ein beliebiges Zeichen,
'hexwert'	für hexadezimale Halbbytezeichen aus 0 bis 9 bzw. A bis F mit einem Wert zwischen 0 und FF,
'dezwert'	für einen Dezimalwert zwischen 0 und 255.

### 2.2 Stellen-Intervall

Geben Sie bitte das Intervall zu berücksichtigender Stellen im Schlüssel an, die überladen werden dürfen:

kleinste mögliche Stellenangabe (1 bis 256 (0=Abbruch)):

und

größte mögliche Stellenangabe (nnn bis 256 (0=Abbruch)):

Durch diese Angaben wird festgelegt, auf welche Stellen sich Überladeinformationen beziehen müssen, um für die Änderung des Schlüssels potentiell von Bedeutung zu sein. Erst wenn zusätzlich die nachfolgend beschriebene Werte-Intervall-Angabe ebenfalls erfüllt ist, führt eine Überladeinformation tatsächlich zum Überladen einer Schlüsselstelle mit dem in der Überladeinformation angegebenen Wert.

### 2.3 Werte-Intervall

Geben Sie bitte das Intervall zu berücksichtigender Werte an, mit denen die Stellen im Schlüssel überladen werden dürfen (in Dezimal-Darstellung):  
kleinste mögliche Wertangabe (0 bis 255 (-1=Abbruch)):

und

größte mögliche Wertangabe (nnn bis 255 (-1=Abbruch)):

Durch diese Angaben wird festgelegt, welche Wertangaben Überladeinformationen beinhalten müssen, um für die Änderung des Schlüssels potentiell von Bedeutung zu sein. Erst wenn zusätzlich die zuvor beschriebene Stellen-Intervall-Angabe ebenfalls erfüllt ist, führt eine Überladeinformation tatsächlich zum Überladen einer Schlüsselstelle mit dem in der Überladeinformation angegebenen Wert. Dabei kann eine Stelle im Schlüssel auch mehrfach überladen werden. Die letzte auf eine Stelle anzuwendende Überladeinformation entscheidet in diesem Fall, welcher Wert der endgültige Schlüsselwert an dieser Stelle sein wird.

Stellen- und Werte-Intervall-Angaben sind im Prinzip nur für Überladedateien hauptsächlich im Direktmodus von Interesse, um z.B. „Trivial“-Überladungen zu verhindern. Im Direktmodus kann nämlich **jede** Datei als Überladedatei angegeben werden, auch Programm-Dateien (\*.COM, \*.EXE) bzw. sonstige Binärdateien (!).

### 2.4 Eingabe-Modus

In welchem Eingabe-Modus möchten Sie die Überladeinformationen eingeben?

- 0 zweimal über Tastatur o h n e Echo auf dem Bildschirm  
(das 2. Mal zur Kontrolle!)
- 1 einmal über die Tastatur m i t Bildschirm-Kontrolle
- 2 über eine Datei, die die Überladeinformation enthält
- 9 Programmabbruch.

Geben Sie bitte die zugehörige Zahl an:

### 2.5 Tastatur-Eingabe

Falls Sie die Überladeinformationen über die Tastatur eingeben wollen, wird Ihnen je nach Informationsart mit:

'1':

Geben Sie bitte Ihre Überladeinformation in der Form "<bytenr-dez>:<zeichen>,"

'2':

Geben Sie bitte Ihre Überladeinformation in der Form "<bytenr-dez>:<hexwert>,"

'3':

Geben Sie bitte Ihre Überladeinformation in der Form "<bytenr-dez>:<dezwert>,"

'4':

Geben Sie bitte Ihre Überladeinformation in der Form "<bytenr-hex>:<zeichen>,"

'5':

Geben Sie bitte Ihre Überladeinformation in der Form "<bytenr-hex>:<hexwert>,"

'6':

Geben Sie bitte Ihre Überladeinformation in der Form "<bytenr-hex>:<dezwert>,"

'7':

Geben Sie bitte Ihre Überladeinformation in Form einer fortlaufenden  
Zeichenkette

jeweils mit

(Info-Ende: Leerzeile) ein:

bzw.

(Info-Ende: Leerzeile) das 1. Mal ein:

bzw.

(Info-Ende: Leerzeile) das 2. Mal ein (Kontroll-Eingabe):

die Gelegenheit gegeben.

## 2.6 Eingabe via Überlade-Datei

Falls Sie eine Überladedatei verwenden wollen, werden Sie mit:

Wie möchten Sie den Namen der Datei, die die Überladeinformation enthält, eingeben?

- 0 zweimal über Tastatur o h n e Echo auf dem Bildschirm (das 2. Mal zur Kontrolle!)
- 1 einmal über Tastatur m i t Bildschirm-Kontrolle
- 9 Programmabbruch.

Geben Sie bitte die zugehörige Zahl an:

zunächst nach der Art der Eingabe des Dateinamens der Überladedatei gefragt. Danach wird mit:

Geben Sie bitte den Namen der Datei an, die die Überladeinformation enthält (Leereingabe = Programmabbruch):

bzw. mit:

Geben Sie bitte den Namen der Datei, die die Überladeinformation enthält, das 1. Mal an (Leereingabe = Programmabbruch):

und mit:

Geben Sie bitte den Namen der Datei, die die Überladeinformation enthält, das 2. Mal an (Kontroll-Eingabe):

nach dem Dateinamen gefragt.

## 2.7 Überladeinformations-Bearbeitung

Falls mehrere Überladeinformationen angegeben werden sollen, empfiehlt sich fast immer eine entsprechende Überladedatei. Eine im aktuellen Programmablauf verwendete Überladedatei wird in diesem Programmablauf n i c h t behandelt (also auch nicht zerstört), selbst wenn diese explizit oder implizit als zu behandelnde Datei angegeben wurde (!).

Zeilenabschlusszeichen dürfen auch bei der Eingabe von Überladeinformationen beliebig verwendet werden, um z.B. Informationen auf einzelne Zeilen aufzuteilen. Zeilenabschlusszeichen werden bei der Eingabe von Überladeinformationen allgemein ignoriert bzw. nur bei der Tastatur-Eingabe in Form einer zusätzlichen Leerzeile als Ende der Überladeinformationen interpretiert.

Die Bearbeitung der Überladeinformationen wird mit dem Hinweis:

Ende der Überlade-Informations-Eingabe.

abgeschlossen.

## 3 Kontakt-Daten

**Anschrift:**

Systemberatung Jürgen Müller  
Mauerstr. 26  
D-64289 Darmstadt

**eMail-Adresse:**

[info@sysjm.de](mailto:info@sysjm.de)

**web-Adresse:**

[www.sysjm.de](http://www.sysjm.de)

**Telefon:**

+49 (0)6151 / 71 51 42

**Fax:**

+49 (0)6151 / 71 02 42