

**W**Eingabe

**Bedienungsanleitung**

---

**Dateneingabeprogramm**  
für Microsoft® Windows® XP/VISTA/7/8/10

von Rolf Rimmele



## Inhalt

Lesetips.....	5
WEingabe stellt sich vor.....	7
Programmeigenschaften.....	7
Bedienungsanleitung.....	9
Installation.....	9
Deinstallation.....	9
Programm WEINGABE.EXE starten.....	10
Eingabemaske bestimmen.....	10
Daten eingeben.....	10
Weitere Bedienungsmerkmale.....	11
Daten speichern.....	13
Programm WEingabe beenden.....	13
Datenexport.....	13
Wie man mit einer Initialisierungsdatei eine Eingabemaske erstellt.....	15
Format der Datei *.INI.....	15
Anlegen einer Initialisierungsdatei.....	15
Demonstrationsbeispiel.....	16
Syntax-Referenz.....	21
[CODES]-Abschnitt.....	21
[CODES#]-Abschnitt.....	24
[SUB]-Abschnitt.....	26
[ITEMS]-Abschnitt.....	28
[NAMES]-Abschnitt.....	29
[NAMES#]-Abschnitt.....	31
[RANGE]-Abschnitt.....	33
[LABEL]-Abschnitt.....	34
[LABEL#]-Abschnitt.....	35
[SKIP]-Abschnitt.....	36
[SKIPMESSAGE]-Abschnitt.....	38
[MISSING]-Abschnitt.....	39
[MISSING#]-Abschnitt.....	41
[HELP]-Abschnitt.....	42
[HELP#]-Abschnitt.....	43
[MESSAGE]-Abschnitt.....	45
[ALIAS]-Abschnitt.....	46
[AUTOFILL]-Abschnitt.....	47
[AUTOFILL#]-Abschnitt.....	48
[CASELABEL]-Abschnitt.....	49
[LEVEL]-Abschnitt.....	50
[SET]-Abschnitt.....	52
TITEL.....	52
DELIMITER.....	52
DEZIMAL.....	52
AUTO.....	53
NEXT.....	53
SKIPMESSAGE.....	54
MINUS.....	54
SPS.....	55
SAV.....	57
CASELABEL.....	57
LAUNCH.....	57
Index.....	59
Anhang.....	60
Fragebogen-Beispiel.....	60
Die Initialisierungsdatei BEISPIEL.INI.....	62

---

## **Lesetips**

Es ist nicht erforderlich, die vorliegende Bedienungsanleitung von Anfang bis Ende zu lesen, um mit WEingabe arbeiten zu können. Hier einige Hinweise, auf welchen Seiten Sie welche zweckdienlichen Informationen finden.

Wenn Sie daran interessiert sind, wie man mit WEingabe praktisch arbeitet, lesen Sie die Seiten 9 bis 14. Sie enthalten alle Informationen darüber, wie man das Programm installiert und startet, wie man Daten eingibt, abspeichert und exportiert.

Wenn Sie hingegen wissen wollen, wie man eine Eingabemaske erstellt, sollten Sie zunächst die Seiten 15 bis 20 in Verbindung mit dem im Anhang ab Seite 60 abgedruckten Fragebogenbeispiel lesen. Danach kennen Sie alle elementaren Regeln, und sollten in der Lage sein, eine einfache Eingabemaske zu erstellen. Sollten die Instruktionen dazu nicht ausreichen, können Sie in der ausführlichen Syntax-Referenz ab Seite 21 die benötigten Details nachschlagen.



## Was ist WEingabe ?

**WEingabe** ist ein Anwendungsprogramm für **Windows**<sup>®</sup> **XP/VISTA/7/8/10**. Es sorgt für die einfache, sichere und schnelle Eingabe und Kodierung von Daten jeder Art, z.B. aus Motivations-, Interessen- und Persönlichkeitsfragebögen, aus Einstellungs- oder Leistungstests. Die Struktur der Datensätze wird in einer Initialisierungsdatei definiert, die vom Benutzer nach einfachen Regeln mittels eines Texteditors angelegt werden kann. WEingabe ist besonders geeignet für Datentypen im Bereich sozialwissenschaftlicher empirischer Untersuchungen wie Fragebogenitems und Testaufgaben, insbesondere Itemreihen mit dichotomer oder mehrstufiger Antwortskala. WEingabe verwendet ein universelles Dateiformat (tabellarische ASCII-Datei) mit vielfältigen Exportmöglichkeiten. Es ist einfach installierbar und hat einen relativ geringen Speicherbedarf, zugleich können beliebig viele Datensätze erfaßt werden.

**WEingabe** besitzt eine Benutzeroberfläche, die hohen Bedienungskomfort bietet und eine optimale Benutzerführung ermöglicht. Sein Funktionsumfang wurde von Version zu Version erweitert, viele Details wurden verbessert, und Verbesserungsvorschläge von Benutzern wurden eingearbeitet.

Die WEingabe kennzeichnenden Merkmale sind:

- Einfache und problemlose Erstellung der Eingabemaske.
- Volle Portierbarkeit der Datenstruktur nach **SPSS** für **WINDOWS** einschließlich aller Variablendefinitionen (Name, Typ, Format, Label, Missing-Values).
- ergonomische und übersichtlich aufgebaute Bedienungs Oberfläche.
- leicht verständliche, selbstinstruktive Funktionsweise.
- Schnelle, bequeme und fehlerminimierende Eingabeprozedur.
- On-Line-Hilfesystem.

## Programmeigenschaften

WEingabe ist gezielt auf die Anforderungen abgestimmt, die sich bei die Eingabe und Kodierung der Daten in empirischen Forschungsprojekten üblicherweise stellen. Dazu gehören:

- Es müssen Kontrollmechanismen eingebaut sein, durch die sich inkorrekte oder illegale Benutzereingaben ( z. B. durch Tippfehler oder mangelnde Sorgfalt ) sicher abfangen und blockieren lassen.
- Es müssen präzise Formatbeschreibungen für die als Eingabe zulässigen Zeichen und Werte verfügbar sein.
- Eine passende Eingabemaske zu entwickeln, muß leicht und schnell zu bewerkstelligen sein. Das gleiche gilt, wenn es darum geht, die mit der Dateneingabe befaßten Personen einzuweisen oder einzuarbeiten.
- Programmdesign und Programmfunktionen sollen keinen Optionen- Wirrwarr ("Featuritis") bieten, sondern rationell und evident gestaltet sein.
- Die Bedienungs Oberfläche muß so ergonomisch sein, daß sie ein durchgehend flüssiges und schnelles Arbeitstempo garantiert .
- Die eingegebenen und abgespeicherten Daten müssen ohne Informationsverlust in datenverarbeitende andere Anwendungen, etwa in Statistikprogramme wie SPSS, übertragen werden können. Dazu gehört unter anderem, daß Variablentyp, Variablenname, Label oder Missing-Values exportierbar sind. Müßte der Datensatz im Zielprogramm erst diesbezüglich nachbereitet werden, wäre dies bei großen Datenmengen eine zeitraubende Angelegenheit.

Für empirische Vorhaben, in denen diese Kriterien bei der Dateneingabe an erster Stelle stehen, ist **WEingabe** eine optimale Lösung, denn:

- Es ist einfach und intuitiv zu bedienen, setzt nur minimales Computerwissen voraus und erfordert praktisch keine Einarbeitungsphase. Die wenigen Kenntnisse, die man braucht, kann man sich mit Hilfe der ausführlichen **kontextsensitiven Online-Hilfe** erwerben. Mit ein

paar Mausklicks läßt sich das ganze Programmfenster erkunden, d.h. man wandert einfach per Maus von Element zu Element und läßt sich per Mausklick dessen jeweilige Beschreibung anzeigen.

- Der Aufbau des Programmfensters und der technische Ablauf der Eingabeprozedur ist logisch klar und übersichtlich. Eine Fehlbedienung ist so gut wie ausgeschlossen. Alle wichtigen Befehle können über Schaltflächen angewählt werden. Durch frei positionierbare Werkzeugleisten (toolbars) kann die Arbeitsfläche den eigenen Vorlieben angepaßt werden. Die Dateneingabe mit WEingabe ist folglich schnell und dadurch letztlich auch kostensparend.
- **WEingabe** stellt leistungsstarke Mittel bereit, um die Dateneingabe zu unterstützen und zu kontrollieren. Ein eigenes Hilfefenster im Programmfenster erlaubt es, synchron zur Dateneingabe den Benutzer ständig mit aktuellen Erläuterungen und Anweisungen, z.B. Kodierungsvorschriften, zu versorgen. Er kann also flüssiger und schneller arbeiten. Besonders wichtige Hinweise oder Warnungen können auch in Message-Boxen präsentiert werden. Man kann strenge und exakte Zulässigkeits- und Bereichsregeln für die einzugebenden Zeichen und Werte festlegen. So können Fehler bereits beim Eintippen zuverlässig abgefangen werden. Zur Erhöhung der Flexibilität stehen 'Skip'-Regeln zur Verfügung.
- **WEingabe** speichert die Daten in einem Format, das von fast allen tabellenverarbeitenden Programmen wie Excel, dBase und insbesondere von SPSS gelesen (importiert) werden kann. Seine besondere Fähigkeit liegt jedoch in seinem hohen Anpassungsgrad an die Eigenschaften des Statistikprogrammes **SPSS** für WINDOWS. Mit WEingabe können nicht nur Datenwerte an sich, sondern auch SPSS-spezifische Formatbeschreibungen der Daten portiert werden. Das Programm erlaubt SPSS-kompatible Definitionen von **Variablennamen**, **Variablentypen**, **Variablen-Labels** und **Missing-Values**, und kann aus diesen Formatbeschreibungen je nach Wunsch entweder eine innerhalb SPSS ablauffähige SYNTAX-Datei (\*.SPS) erzeugen oder (ab Version 4) eine SPSS-Datenbank (\*.SAV) erstellen, in der Daten und Formatangaben gleichermaßen enthalten sind.
- **WEingabe** bietet einen praktischen Weg, um eine Eingabemaske zu erstellen. Man muß nicht umständlich mit Tastenkombinationen hantieren, sondern man schreibt eine Initialisierungsdatei, aus der das Programm während der Laufzeit die benötigten Informationen liest. Diese Initialisierungsdatei ist eine reine ASCII-Textdatei, die mit einem ganz normalen Texteditor angelegt wird. Die Konstruktionsregeln sind einfach und flexibel. Eine Standardmaske für einen mittleren Fragebogen zu definieren, läßt sich mit wenigen Zeilen erledigen und dauert vielleicht 15-30 Minuten. Eine einmal erstellte Initialisierungsdatei kann jederzeit gelesen, bearbeitet, verändert oder als Muster für weitere Initialisierungsdateien benutzt werden.
- **WEingabe**'s beansprucht wenig Festplattenplatz und läßt sich schnell kopieren und an Eingabe weiterreichen, die es dann auf dem eigenen Computer installieren können. Das mitgelieferte Setup-Programm macht die Installation auch für Anfänger einfach. WEingabe deinstalliert sich außerdem auf Kommando wieder selbsttätig.

# Bedienungsanleitung

## Installation

- Sie benötigen einen Rechner mit einer CPU 80386 oder höher, Microsoft® Windows 98/ME/2000 oder Microsoft® Windows XP, und eine Maus. Die Bildschirmauflösung sollte nicht unter 800\*600 Bildpunkten liegen, da sonst das Programmfenster nicht in seiner vollen Größe sichtbar ist.
- Zur Installation brauchen Sie den Windows-Installer, der ab den späteren Versionen von Windows 98 und bei 2000/XP im Lieferumfang enthalten ist, ansonsten auf der Website von Microsoft besorgt werden muss. Legen Sie die Installationsdiskette ins Laufwerk A: und wählen Sie "**Ausführen...**" im Startmenü der Taskleiste. Geben Sie "<Laufwerksname>:\setup" oder "<Laufwerksname>:\weingabe.msi" ein.
- Der Installationsvorgang läuft selbsttätig ab bzw. ist selbsterklärend. Sie können ein Zielverzeichnis bestimmen oder das vom Setup-Programm vorgeschlagene Verzeichnis "**C:\PROGRAMME\WEINGABE**" akzeptieren. Nach vollendeter Installation befinden sich folgende Dateien im Zielverzeichnis Ihrer Festplatte:

**WEINGABE.EXE** - das Hauptprogramm

**BEISPIEL.TXT** - Beispiel einer Initialisierungsdatei mit der Definition einer Eingabemaske.

Sie können diese Datei verwenden, um zu Demonstrationszwecken Daten einzugeben, beispielsweise damit Sie WEingabe ausprobieren und seine Funktionsweise kennenlernen können. Da WEingabe erwartet, daß der Name der Initialisierungsdatei mit der Extension .INI endet, nennen Sie zu diesem Zweck die Datei BEISPIEL.TXT in BEISPIEL.INI um. Den Inhalt der Datei BEISPIEL.TXT finden Sie abgedruckt im Anhang ab S. 62.

**Hinweis:** Bei der Installation von WEingabe zum Zwecke der realen Dateneingabe muß die Initialisierungsdatei natürlich vorher von Ihnen selbst erstellt und benannt werden. Wie Sie eine Initialisierungsdatei im einzelnen erstellen und was Sie mit ihr machen müssen, wird ab S.15 ausführlich beschrieben.

**INIEDIT.EXE** - ein Texteditor zum Schreiben der Initialisierungsdatei (Eingabemaske) mit nützlichen Assistenzfunktionen

**INIEDIT.CHM** – Online-Hilfe für INIEDIT

Außerdem:

WEINGABE.HLP - Hilfedatei

WEINGABE.CNT - Hilfedatei

TIPS.TXT - Tips beim Programmstart

SPSSIO.DLL - wichtige Systemdatei

- Durch die Installation wird dem Menü Programme, das Windows anzeigt, wenn Sie auf die Schaltfläche Start der Taskleiste klicken und dann auf Programme zeigen, ein neuer Eintrag namens WEingabe hinzugefügt.

## Deinstallation

WEingabe läßt sich jederzeit automatisch deinstallieren und vollständig von der Festplatte löschen. Und so geht's:

- Öffnen Sie die Windows-Systemsteuerung und darin den Ordner *Software*. Klicken Sie auf den Eintrag WEingabe und wählen Sie *Entfernen*.

Gelöscht werden sowohl das durch die Installation geschaffene Verzeichnis als auch alle bei der Installation erzeugten, oben namentlich aufgeführten Dateien. Nicht gelöscht werden Dateien, die im Verlauf der Dateneingabe angelegt wurden, so daß kein unbeabsichtigtes Löschen von Datenbeständen befürchtet werden muß.

## Programm WEINGABE.EXE starten

- Klicken Sie auf die Schaltfläche Start, und zeigen Sie dann auf Programme.
- Klicken Sie auf den Eintrag WEingabe.

## Eingabemaske bestimmen

Nach dem Start muß zunächst die Initialisierungsdatei gewählt werden, die das Format der einzugebenden Daten beschreibt (Eingabemaske). Außerdem informiert die Initialisierungsdatei WEingabe über den Dateinamen, unter dem die eingegebenen Daten gespeichert werden (siehe S. 13). Es erscheint eine Dialogbox mit der Überschrift " \*.INI-Datei bestimmen ". Bestimmen Sie das Verzeichnis, markieren Sie dann den Namen der gewünschten Initialisierungsdatei und klicken Sie abschließend auf die Schaltfläche Öffnen. Die Dateneingabe kann nun unmittelbar beginnen.

Ist im gleichen Verzeichnis bereits eine Datendatei vorhanden, die namensgleich mit dem Titel der INI-Datei ist (der Name der Datendatei setzt sich aus dem Titel der INI-Datei und der Endung **.SYS** zusammen, siehe S.13), wird diese automatisch geöffnet und die Dateneingabe kann an der Stelle fortgesetzt werden, wo sie beim letzten Mal unterbrochen wurde. Findet WEingabe keine solche Datei, wird eine neue Datei angelegt und die Dateneingabe beginnt von vorne.

Beim Einlesen einer \*.SYS-Datei checkt WEingabe außerdem das Variablenformat und vergleicht es mit den Formatdefinitionen der Initialisierungsdatei. Unstimmigkeiten führen zu einer Fehlermeldung und es können keine Daten eingegeben werden.

## Daten eingeben

Die Dateneingabe wird über Eingabefelder abgewickelt, deren Format in der Initialisierungsdatei beschrieben werden muß. Ein Eingabefeld kann entweder 1 ein- bis mehrstellige numerische Variable, oder 1 String-Variable, oder aber bis zu 99 einstellige numerische Variablen aufnehmen. Die für ein Eingabefeld geltenden Kodierungsvorschriften im Bezug auf Art und Anzahl der als Eingabe erlaubten Zeichen werden in den Abschnitten [CODES], [CODES#] und [SUB] der Initialisierungsdatei festgelegt: Es können nur solche Zeichen eingegeben werden, die in den Abschnitten [CODES] und [CODES#] als zulässig deklariert worden sind, während die Anzahl der Zeichen, die eingegeben werden müssen oder dürfen, im Abschnitt [SUB] geregelt wird. Im Abschnitt [RANGE] können außerdem Wertebereiche definiert werden, innerhalb derer die eingegebenen Werte liegen müssen. Eine genaue Beschreibung der Formatierungsanweisungen finden Sie ab Seite 21.

Tastaturfunktionen und Mausbedienung (Bewegung des Cursors, Markieren, Einfügen oder Löschen von Zeichen) sind konform zum Windows-Standard. Zur Unterstützung der Dateneingabe können außerdem effiziente Orientierungshilfen eingesetzt werden, siehe Abschnitt [MESSAGE], Abschnitt [HELP] und Abschnitt [HELP#].

Man kann sowohl innerhalb eines Datensatzes als auch zwischen verschiedenen Datensätzen beliebig hin- und her blättern, um bereits getätigte Eingaben zu überprüfen und gegebenenfalls zu ändern. Zum Blättern können entweder Tastatur oder die Schaltflächen der Werkzeugleiste benutzt werden.

Durch Drücken der RETURN-Taste oder durch Anklicken der entsprechenden Schaltfläche gelangt man zum nächsten Eingabefeld. WEingabe akzeptiert jedoch keine "leeren" Eingabefelder: Man kann das nächste Eingabefeld nicht aktivieren, bevor nicht mindestens 1 Zeichen eingegeben wurde. Wenn die erforderlichen Daten für ein Eingabefeld nicht vorhanden sind (sogenannte "user-missings") muß dies durch Eingabe eines entsprechenden Missing-Value ausgeglichen werden. Durch diese Vorkehrung soll erstens der Versuchung entgegengewirkt werden, bei der Dateneingabe zu "mogeln". Zweitens wird die Kompatibilität zu SPSS für WINDOWS gewahrt, das leere Datenfelder nicht als "user-missing", sondern als "system-missing" behandelt. Ein Überspringen von Eingabefeldern ist nur möglich, wenn dafür im Abschnitt [SKIP] entsprechende Skip-Regeln vorhanden sind.

## Weitere Bedienungsmerkmale

Für alle wichtigen Programmbefehle gibt es Symbolleisten mit Schaltflächen, die mit der Maus angeklickt werden können. Befehle, deren Schaltfläche eine graue Beschriftung aufweist, sind vorübergehend nicht wählbar. Die Bedeutung einer Schaltfläche läßt sich am einfachsten herausfinden, indem man mit dem Mauszeiger darüber kurz verharrt und das "Quickinfo" liest, das daraufhin eingeblendet wird. Überhaupt lassen sich alle Elemente des Programmfensters per Maus erkunden. Zu diesem Zweck gibt es eine spezielle Schaltfläche, die an einem aufgemalten Pfeil mit Fragezeichen zu erkennen ist. Wenn Sie diese mit der Maus anklicken, verändert sich der Mauszeiger und wird selbst zu einem Pfeil mit Fragezeichen. Danach brauchen Sie nur noch auf den Bereich oder das Element im Programmfenster klicken, worüber Sie mehr wissen wollen. Auf gleiche Weise lassen sich auch die Menüs erkunden. Symbolleisten lassen sich beliebig im Programmfenster verschieben, indem Sie mit der auf eine leere Stelle der Leiste zeigen, und mit gedrückter linker Maustaste die Leiste an die gewünschte Position ziehen.

## Beschreibung der wichtigsten Menü-Befehle

- **Menü Datei**

⇒ Der Befehl **Neu**

Verwenden Sie diesen Befehl, um eine Eingabemaske zu laden und ein Fenster für die Dateneingabe zu öffnen. WEingabe zeigt ein Dialogfeld an, in dem Sie die Datei finden und öffnen können, die die Eingabemaske enthält.

Alternativ kann auch der Befehl Öffnen verwendet werden.

Shortcut: **STRG+N**

⇒ Der Befehl **Öffnen**

Verwenden Sie diesen Befehl, um eine Eingabemaske zu laden und ein Fenster für die Dateneingabe zu öffnen. WEingabe zeigt ein Dialogfeld an, in dem Sie die Datei finden und öffnen können, die die Eingabemaske enthält.

Alternativ kann auch der Befehl Neu verwendet werden.

Es lassen sich auch mehrere Eingabemasken gleichzeitig öffnen. Mit Hilfe des Menüs Fenster schalten Sie zwischen mehreren Eingabefenstern um.

Symbolleiste:

Shortcut **STRG+O**

⇒ Der Befehl **Schließen**

Schließt das aktive Fenster einschließlich der damit verbundenen Eingabemaske. Bevor das Fenster geschlossen wird, nimmt WEingabe automatisch eine Datensicherung vor. Um das Fenster erneut zu öffnen, verwenden Sie die Befehle Öffnen oder Neu.

Der Befehl Schließen beendet nicht das Programm, benutzen Sie dazu den Befehl Beenden.

⇒ Der Befehl **Drucken**

Verwenden Sie diesen Befehl, um den Inhalt des Datenansichtfensters auszudrucken. Bei Auswahl des Befehls erscheint das Dialogfeld Drucken, in dem Sie Optionen zur Druckereinstellung angeben können.

⇒ Der Befehl **Beenden**

Verwenden Sie diesen Befehl zum Beenden Ihrer Sitzung mit WEingabe. Alternativ dazu können Sie aus dem Systemmenü den Befehl Schließen wählen. Das Systemmenü wird geöffnet, wenn Sie die linke obere Fensterecke anklicken.

Shortcut: **ALT+F4**

- **Menü Bearbeiten**

⇒ Der Befehl **Eingabefeld vorher**

Geht zum vorangehenden Eingabefeld

Shortcut: **BILD UNTEN**

⇒ Der Befehl **Datensatz vorher**

Geht zum gleichen Datenfeld des vorangehenden Datensatzes

Shortcut: **STRG+Pfeil links**

⇒ Der Befehl **Datensatzanfang**

Geht zum ersten Datenfeld des Datensatzes

Shortcut: **STRG+POS 1**

⇒ Der Befehl **nächstes Eingabefeld**

Geht zum nächsten Eingabefeld

Shortcut: **BILD OBEN**

⇒ Der Befehl **nächster Datensatz**

Geht zum vorangehenden Eingabefeld

Shortcut: **STRG+Pfeil rechts**

⇒ Der Befehl **Datensatzende**

Geht zum letzten Datenfeld des Datensatzes

Shortcut: **STRG+ENDE**

⇒ Der Befehl **Anfang erster Datensatz**

Geht zum ersten Datenfeld des ersten Datensatzes

Shortcut: **UMSCHALT+POS 1**

⇒ Der Befehl **Eingabefeld aktuell**

Geht zum aktuellen (letzten) Eingabefeld

Shortcut: **UMSCHALT+ENDE**

- **Menü Extra**

⇒ Der Befehl **Datensatz löschen**

Verwenden Sie diesen Befehl, um einen bereits teilweise eingegebenen Datensatz wieder zu löschen. Vollständig eingegebene Datensätze können nicht mehr gelöscht werden. In diesem Falle erscheint der Befehl in grauer Schrift und ist nicht wählbar. Vor dem endgültigen Löschen wird eine Sicherheitsabfrage ausgegeben.

⇒ Der Befehl **Gehezu**

Öffnet ein Dialogfeld, in das sie die Nummer eines Datensatzes sowie die Nummer eines darin zu findenden Datenfeldes eingeben können. Das gesuchte Feld erscheint danach im Dateneingabefenster und kann dort editiert werden.

- **Menü Hilfe**

⇒ Der Befehl **Hilfethemen**

Verwenden Sie diesen Befehl, um den Inhaltsbildschirm der Hilfe darstellen zu lassen. Von diesem Bildschirm aus können Sie alle Bedienungsanweisungen für WEingabe Schritt für Schritt durchgehen. Sobald Sie die Hilfe geöffnet haben, ist es immer möglich, die Schaltfläche Hilfethemen anzuklicken, um zum Inhaltsbildschirm zurückzukehren.

Shortcut: **F1**

## Daten speichern

Die Speicherung der eingegebenen Daten geschieht selbsttätig. WEingabe schreibt fertig eingegebene Datensätze sofort auf Festplatte stellt auch bei Beendigung des Programmes oder beim Schließen des Dateneingabefensters sicher, daß vorher alle noch nicht gespeicherten Daten und vorgenommenen Änderungen gesichert werden.

WEingabe speichert die eingegebenen Daten in zwei Dateien. Die Namen der beiden Dateien setzen sich zusammen aus dem Namen der Initialisierungsdatei und den Namenserverweiterungen \*.SYS bzw. \*.DAT. Die Namen werden von WEingabe selbsttätig gebildet. Wenn also z.B. die Initialisierungsdatei FRAGEBOGEN.INI heißt, stehen nach Beendigung des Programmes die eingegebenen Daten automatisch im aktuellen Arbeitsverzeichnis in den Dateien FRAGEBOGEN.SYS und FRAGEBOGEN.DAT. Arbeitsverzeichnis ist immer das Verzeichnis, in dem sich die Initialisierungsdatei befindet. Die Datei \*.SYS ist die interne Arbeitsdatei für WEingabe. Die Datei \*.DAT ist die **Exportdatei**, eine tabellarisch aufgebaute ASCII-Textdatei im Format "Tab-Delimited", die es erlaubt, die Daten nach SPSS für WINDOWS oder in andere tabellenverarbeitende Programme zu transferieren, auch SPSS solche Dateien importieren.

Da viele Programm, unter anderem auch SPSS, beim Einlesen von Tab-Delimited-Dateien die Spaltenbreite bzw. die Stellenzahl einer Variablen (SPSS) durch die Breite des ersten Datenwertes in der Spalte bestimmt, werden beim Schreiben der Exportdatei Werte, deren Stellenzahl die in der Eingabemaske definierte Feldbreite unterschreiten, durch nachfolgende Leerzeichen aufgefüllt, so daß alle Datenfelder eine konstante Feldbreite aufweisen.

## Programm WEingabe beenden

Die Dateneingabe kann jederzeit unterbrochen und das Programm beendet werden. Benutzen Sie dazu den Befehl Beenden oder Schließen im Menü Datei, alternativ können Sie auch den Befehl Schließen im Systemmenü des Programmfenster verwenden. Es können keine Daten verloren gehen, da WEingabe, wie schon erwähnt, selbsttätig für die Speicherung sorgt und Sie in der nächsten Arbeitssitzung genau an die Stelle zurückführt, an der sie aufgehört haben.

## Datenexport

Nach jeder Arbeitssitzung wird eine **Exportdatei** mit der Dateierweiterung \*.DAT angelegt, in die alle Daten geschrieben werden. Die Exportdatei ist eine tabellarisch aufgebaute ASCII-Textdatei im Format "Tab-delimited", in dem Datenfelder durch Tabulatorzeichen und Datensätze durch Zeilenendezeichen getrennt sind. Dieses Format macht es möglich, die Daten in andere tabellenverarbeitende Programme (Excel, dBase usw.), aber auch nach SPSS zu exportieren. Das Tabulatorzeichen als Trennzeichen zwischen Datenfeldern kann je nach Maßgabe des Zielprogrammes auch durch ein beliebiges anderes Zeichen ersetzt werden (siehe dazu S. 52, Abschnitt DELIMITER).

Der Datentransfer nach SPSS für WINDOWS ist einfach und wird von WEingabe auf zweifache Art unterstützt.

### 1.

Sie können die Exportdatei \*.DAT benutzen, um die Daten nach **SPSS** einzulesen und dort weiter zu verarbeiten:

- Wählen Sie im Menü DATEI (FILE) von SPSS die Option ÖFFNEN (OPEN) → DATEN (DATA). Dadurch öffnet sich die Dialogbox 'Datei öffnen' ('Open File'). Sie enthält eine ausklappbares Listenfenster mit Dateitypen. Klicken Sie hier auf den Eintrag *Daten(\*.dat)*. Suchen und wählen Sie danach Ihre Exportdatei. Die weitere Schritte dürften selbsterklärend sein, notfalls befragen Sie das SPSS-Handbuch. Achten Sie aber darauf, bei der Auswahlfrage, ob die erste Zeile der Datei Variablennamen enthält oder nicht, das richtige Kästchen anzukreuzen.
- Überprüfen Sie die Variablendefinitionen (siehe nächsten Abschnitt).
- Speichern Sie die Daten im SPSS-eigenen Format unter einem anderen Dateinamen neu ab.

Die Exportdatei \*.DAT enthält allerdings nur die Daten und Variablennamen, aber keine sonstigen Formatspezifikationen, nämlich:

- "Labels" zur Erläuterung der Variablennamen
- Code-Definitionen für fehlende Werte zu definieren (Missing-Values).
- Angaben über das genaue Zahlenformat numerischer Variablen oder die Länge von String-Variablen.

Diese Information können jedoch auf andere Weise übertragen werden. WEingabe kann nämlich die in der Initialisierungsdatei eingetragenen Definitionen von **Variablennamen**, **Variablentypen**, **Variablen-Labels** und **Missing-Values** in die Kommandosprache von **SPSS** übersetzen und daraus eine innerhalb **SPSS** lauffähige sogenannte **SYNTAX**-Datei (\*.SPS) zu erstellen. Wie Sie diese Syntax-Datei unter **SPSS** zu verwenden haben, entnehmen Sie bitte den einschlägigen Handbüchern. Wie Sie WEingabe dazu veranlassen, eine Syntax-Datei zu erzeugen, und welche Regeln beim Schreiben der Initialisierungsdatei diesbezüglich zu beachten sind, entnehmen Sie dem Abschnitt **SPS=** auf Seite **55**.

## 2.

Seit Version 4 von WEingabe kann man sich den Umweg über eine \*.DAT-Datei ersparen und ein Datenfile im originären SPSS-Format mit der Dateierdung \*.SAV erstellen. In diesem File sind zugleich alle Formatspezifikationen enthalten, sodaß es nicht nötig ist, eine Syntax-Datei \*.SPS anzulegen. Eine \*.SAV- Datei wird wie eine \*.DAT-Datei nach jeder Arbeitssitzung angelegt, aber nur wenn im Abschnitt [SET] der Eintrag SAV=1 vorhanden ist (siehe Seite 57).

Erfolgt der Datentransfer nach SPSS über eine \*.DAT-Datei, ist es wichtig zu wissen, wie SPSS Tab-delimited Dateien liest, daher hier eine Regeln und Warnhinweise, die bei der Dateneingabe bzw. bei der Definition der Eingabemaske beachtet werden sollten.

Werte können entweder numerisch oder im String-Format sein. Jeder Wert, der nicht numerische Zeichen enthält, wird als String-Wert interpretiert. Der Datentyp (numerisch oder String) wird durch den Typ des **ersten** eingelesenen Datenwertes in der Spalte bestimmt. Nachfolgende Werte anderen Datentyps werden in System-Missing-Werte umgewandelt. Diese Umwandlungen können auch durch nachträgliche Neudefinition der Variablentyps nicht mehr rückgängig gemacht werden. Man sollte deshalb beim Schreiben der Eingabemaske im Abschnitt [CODES] (siehe Seite 21) bei numerischen Variablen die Art der zulässigen Zeichen so spezifizieren, daß keinerlei nicht-numerische Zeichen eingegeben werden können. Die zweite Warnung betrifft String-Variablen. SPSS wandelt eine String-Variable immer dann in eine numerische Variable um, wenn der erste eingelesene nicht-leere Datenwert in der Spalte eine Zeichenfolge ist, die nur aus Ziffern besteht. Um ganz sicher zu gehen sollte man deshalb vermeiden, daß Werte für String-Variablen vorkommen, die ausschließlich aus numerischen Zeichen (0 - 9) bestehen. Falls möglich ist es am besten, man unterbindet bei String-Variablen die Eingabe numerischer Zeichen völlig.

Eine andere 'Angewohnheit' von SPSS beim Einlesen von Dateien im Format Tab-delimited besteht darin, daß Stringvariablen, deren Wert für **alle** eingelesenen Fälle (Cases) 'leer' ist, d.h. ausschließlich aus Leerzeichen besteht, beim Import komplett gelöscht ('dropped') werden. Da ein solcher Fall insbesondere bei Fragebogendaten nie auszuschließen ist, sollte man als Missing-Value bei Stringvariablen nie das Leerzeichen wählen, sondern beispielsweise den Punkt oder irgendein anderes Symbol. Unter Umständen kann es sogar empfehlenswert sein, den Gebrauch des Leerzeichens durch eine entsprechende Kodierungsvorschrift im Abschnitt [CODES] (siehe Seite 21) ganz zu unterbinden.

## Wie man mit einer Initialisierungsdatei eine Eingabemaske erstellt.

### Format der Datei \*.INI

Die \*.INI-Datei besteht aus Abschnitten und aus zeilenweise angeordneten Anweisungen innerhalb dieser Abschnitte. Funktion und Syntax dieser Abschnitte werden auf den folgenden Seiten ausführlich dargestellt.

Die Abschnitte haben das folgende Format:

**[ABSCHNITTNAME]**  
**Schlüssel=Wert**

In diesem Beispiel ist [ABSCHNITTNAME] der Name eines Abschnitts. Die eckigen Klammern ([ ]) um den Namen sind unbedingt erforderlich, und die linke eckige Klammer muß am äußersten linken Rand einer Zeile stehen. Der Abschnittsname muß in **Großbuchstaben** geschrieben sein. Leerzeichen sind nicht zugelassen. Wichtig: Eckige Klammern dürfen **nur** verwendet werden, um Abschnittnamen zu kennzeichnen, sonst funktioniert das Programm nicht korrekt.

Der Ausdruck **Schlüssel=Wert** steht für eine Wertzuweisung. Der Schlüssel benennt den Bezug der Wertzuweisung. Das Ziel kann ein Eingabefeld, eine Variable oder eine Programmfunktion sein. Unmittelbar dahinter muß ein Gleichheitszeichen stehen (=). Der Wert kann je nach Bedeutung eine Ganzzahl oder eine Zeichenfolge (ohne Einschluß in Anführungszeichen) sein. Alle Anweisungen müssen am Zeilenanfang beginnen und mit einem Zeilenendezeichen (**RETURN**-Taste) enden. Eine Anweisung darf, sofern nicht ausdrücklich zugelassen, keine Leerzeichen enthalten. Sie können Kommentare in einen Abschnitt einfügen. Jede Kommentarzeile muß mit einem Semikolon (;) beginnen. Ein Abschnitt endet, wenn der nächste Abschnitt beginnt. Die letzte Zeile der INI-Datei sollte immer aus einem alleinstehenden Zeilenendezeichen (Return-Taste) bestehen.

Reservierte Abschnittnamen, deren Gebrauch Sie ab Seite 21 ausführlich erläutert finden, sind **[SET]** (S.52), **[ALIAS]** (S.46), **[ITEMS]** (S.28), **[CODES]** (S.21), **[CODES#]** (S.24), **[SUB]** (S.26), **[NAMES]** (S.29), **[NAMES#]** (S.31), **[LABEL]** (S.34), **[LABEL#]** (S.35), **[RANGE]** (S.33), **[SKIP]** (S.36), **[MISSING]** (S.39), **[MISSING#]** (S.41), **[MESSAGE]** (S.45), **[SKIPMESSAGE]** (S.38), **[HELP]** (S.42), **[HELP#]** (S.43), **[AUTOFILL]** (S.47), **[AUTOFILL#]** (S.48), **[CASELABEL]** (S.49) und **[LEVEL]** (S.50). In ihnen dürfen nur syntaktisch zulässige Ausdrücke oder Kommentare stehen, sonst ist das Verhalten des Programms undefiniert. Die Reihenfolge der Abschnitte ist beliebig.

### Anlegen einer Initialisierungsdatei

Zum Schreiben einer Initialisierungsdatei oder zum Ändern von Einträgen verwenden Sie einen Texteditor, am besten das Programm IniEdit, das zum Lieferumfang von WEingabe gehört. Es ist speziell für das Schreiben von Initialisierungsdateien entwickelt worden. Das Programm besitzt neben den üblichen Standardfunktionen nützliche Extras, die dem Benutzer bestimmte Aufgaben, die bei der Erstellung einer Eingabemaske routinemäßig anfallen, abnehmen oder sie wesentlich erleichtern können. Unter anderem lassen sich mit ihrer Hilfe Zeilen automatisch numerieren oder umnumerieren, und es stehen Steuerbefehle zur Verfügung, mit denen man sich schnell und zielgenau zwischen verschiedenen Stellen im Text hin- und herbewegen kann. Für die meisten Funktionen stehen Tastaturkürzel zur Verfügung, die sich reibungslos in den Arbeitsablauf integrieren lassen. Bei Verwendung eines anderen Texteditors ist es wichtig, daß die Initialisierungsdatei im reinen **Textformat** erstellt und gespeichert wird, d.h. keinesfalls irgendwelche Formatierungen enthalten darf.

Achtung: Seien Sie sehr sorgfältig, wenn Sie die Initialisierungs-Datei schreiben oder bearbeiten. Fehlerhafte Einträge oder Änderungen können zu unvorhersehbaren Ergebnissen bei der Ausführung von WEingabe führen.

Unterziehen Sie die Dateneingabeprozedur einem Probelauf, um sicher zu gehen, daß die fertige Initialisierungsdatei wie beabsichtigt funktioniert und vor allem keine schwerwiegenden Fehler enthält, durch welche die Lauffähigkeit von WEingabe gestört werden könnte.

Wenn Sie die zum Ausprobieren notwendige Tipparbeit sich sparen oder verringern wollen, können Sie durch den Eintrag AUTO=1 im Abschnitt [SET] der Initialisierungsdatei und mit den Tasten F11 und F12 zufällige Datenwerte erzeugen. Genauere Hinweise dazu finden Sie auf Seite 53.

Wählen Sie für die fertige Initialisierungsdatei einen passenden Namen, z.B. TESTHEFT. Für die Identifizierung der Datei durch WEingabe ist es wichtig, daß Sie dem Namen die Endung .INI hinzufügen, so daß der vollständige Dateiname im angeführten Beispiel TESTHEFT.INI lautet. Den Namen der Initialisierungsdatei verwendet WEingabe später bei der Dateneingabe zur Bildung des Dateinamens für die Datenfiles (siehe S.13), so daß zusammengehörende Dateien immer am gleichlautenden Namen zu erkennen sind. Die Initialisierungsdatei kann in jedem beliebigen Verzeichnis stehen. Wichtig zu wissen ist nur, daß dieses Verzeichnis zugleich immer das **Arbeitsverzeichnis** von WEingabe ist, in dem sich auch die Dateien mit den eingegebenen Daten befinden, die das Programm anlegt.

Um beim Rücklauf den Überblick zu bewahren, empfiehlt es sich INI-Dateien, die für verschiedenen Eingaber/innen bestimmt sind, auch verschiedene Namen zu geben.

## Demonstrationsbeispiel

Im folgenden werden an einem kleinen Demonstrationsbeispiel die Grundzüge besprochen, nach denen beim Schreiben einer Initialisierungsdatei zu verfahren ist.

Erstellt werden soll eine Eingabemaske für einen Fragebogen, der aus den 9 Fragen besteht, die im Anhang ab Seite 60 abgedruckt sind. Die Fragen sind nach Gutdünken zusammengestellt und stehen in keinem inhaltlichen Zusammenhang.

Aus den 9 Fragen lassen sich insgesamt 23 Variablen bilden. Die Fragen 3,4,5 schließen jeweils 2 Variablen ein, Frage 7 umfaßt 3 Variablen und Frage 8 enthält 10 Variablen. Die Variablen in Frage 8 lassen sich (bei einstelliger numerischer Kodierung der Antworten, siehe Seite 21, Punkt 2b) in ein einziges Eingabefeld packen, für alle anderen Variablen wird je 1 Eingabefeld benötigt. Somit umfaßt die zu erstellende Eingabemaske 14 Eingabefelder.

Die Anzahl der Eingabefelder muß im Abschnitt [ITEMS] deklariert werden. Wir beginnen also unsere Initialisierungsdatei mit folgendem Eintrag:

**[ITEMS]**  
**N=14**

Als nächstes definieren wir für jedes Eingabefeld den Variablentyp (numerisch oder String) und die zur Kodierung verwendbaren Zeichen. Dies geschieht im Abschnitt [CODES].

### [CODES]

Die Variable in Eingabefeld 1 ist numerisch und soll mit 0 oder 1 codiert werden. Fehlt die Ankreuzung, soll eine 9 eingegeben werden.

**1=019**

Diese Anweisung definiert die Variable in Eingabefeld 1 als einstellig-numerisch und begrenzt den Bereich der zulässigen Codes auf die Ziffern 1,0,9. Andere Zeichen können nicht eingegeben werden.

Die Variablen in den Eingabefeldern 2 und 3 und 4 sind ebenfalls numerisch, als Wert kommen jedoch auch zweistellige Zahlen in Betracht. Für Variablen diese Typs muß folgender Eintrag gemacht werden:

**2=@**

**3=@**

**4=@**

Das Symbol @ kennzeichnet eine Variable, bei der als Werteingabe eine beliebige (1-oder mehrstellige) **Ganzzahl** erlaubt ist. Es können keine nicht-numerischen Zeichen eingegeben werden.

Da alle drei Variablen vom gleichen Typ sind, können die drei Definitionen zusammengefaßt und in Form einer Aufzählung ( Einzelheiten siehe S.21) innerhalb einer einzigen Zeile niedergeschrieben werden, so daß also folgender Eintrag richtig ist:

**2,3,4=@**

Die Variablen in den Eingabefeldern 5 und 6 können wie die Variable in Eingabefeld 1 einstellig numerisch kodiert werden. Da es 5 Antwortalternativen gibt, sollen zur Kodierung die Zahlen 1 bis 5 und als Missing-Value der Wert 9 verwendet werden. Wir schreiben also:

**5=123459**

**6=123459**

oder wir vereinfachen, da die Kodierungsvorschriften der beiden Variablen identisch sind, zu:

**5,6=123459**

Die Eingabefelder 7, 8, 10, 11 und 12 beinhalten beide eine String-Variable. Die einfachste Kodierungsregel für String-Variablen hat die folgende Form:

**7=**

**8=**

Die beiden Einträge bestehen aus einer leeren Zuweisung (hinter dem Gleichheitszeichen darf sich kein Leerzeichen befinden). Eine leere Zuweisung läßt als Eingabe alle "druckbaren" Zeichen zu, d.h. alle Zeichen mit einem ASCII-Code von 32-127 sowie alle deutschen Umlaute. Im Fragebogenbeispiel jedoch soll WEingabe zwecks Fehlervermeidung nur Großbuchstaben und das Zeichen \_ als gültig anerkennen und den Versuch, ein anderes Zeichen einzutippen, mit einem mahnenden Piepton abwehren. Die Kodierungsregel muß dann lauten:

**7= ÄÖÜABCDEFGHIJKLMNOPQRSTUVWXYZ\_.**

**8= ÄÖÜABCDEFGHIJKLMNOPQRSTUVWXYZ\_.**

.....

Weil die Variablen der fünf Eingabefelder 7,8,10,11,12 alle zum Typ String gehören und die Kodierungsvorschriften identisch sind, kann man der Einfachheit halber schreiben:

**7,8,10THRU12= ÄÖÜABCDEFGHIJKLMNOPQRSTUVWXYZ\_.**

Mit dem Bereichsoperator THRU (siehe Seite 21) kann man die Aufzählung unmittelbar aufeinanderfolgender Eingabefelder durch eine Intervallangabe vereinfachen.

Man beachte, daß in der Zeichenmenge rechts vom Gleichheitszeichen das Leerzeichen fehlt, was in diesem Fall bedeutet, daß es auch nicht eingegeben werden kann. Dies kann, wenn nur einzelne Wörter einzutippen sind, eine nützliche Maßnahme sein. Falls das Leerzeichen erlaubt sein soll, muß es also explizit in die Zeichenfolge eingetragen werden.

Obwohl nur Großbuchstaben als zulässige Zeichen deklariert, braucht bei der Eingabe nicht die SHIFT-Taste gedrückt werden, da WEingabe eingegebene Kleinbuchstaben automatisch in Großbuchstaben konvertiert.

Sollen auch Kleinbuchstaben zulässig sein, müssen sie angegeben werden, also z.B.:

**7=äöüabcdefghijklmnopqrstuvwxyzÄÖÜABCDEFGHIJKLMNOPQRSTUVWXYZ\_.**

Die Deklaration für Eingabefeld 9 soll lauten:

**9=129**

Eingabefeld 13 soll die 10 einstellig-numerisch zu kodierenden Variablen von Frage 8 aufnehmen.

Da alle Variablen auf einer 5-stufigen Antwortskala bewertet werden, reicht es aus, wenn wir schreiben:

**13=123459**

Eingabefeld 14 liefert ein schon "fortgeschritteneres" Beispiel. Die einzugebende Zeichenfolge ist vom Typ String, sie unterliegt jedoch bestimmten Einschränkungen. An den Positionen 1,2,5,6 dürfen nur Buchstaben vorkommen, an den Positionen 3 und 4 nur Zahlen oder im Falle von „missing data“ der Buchstabe X. Da es bei Kennwörtern auf unbedingte Korrektheit ankommt, entschließen wir uns zu einer Sicherheitsmaßnahme.

Zunächst definieren wir den Variablentyp in dem Eingabefeld, nämlich:

**14=**

Um zu verhindern, daß aus Versehen statt Zahlen Buchstaben eingegeben werden und umgekehrt, definieren wird separat für jede Position im Eingabefeld die Menge der zulässigen Zeichen. Dazu erstellen wir einfach einen **eigenen** Abschnitt mit dem Namen **[CODES14]**, und schreiben dort analog zum bisherigen Vorgehen unsere Anweisungen hinein, also:

**[CODES14]**

**1=ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ**

**2=ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ**

**3=0123456789X**

**4=0123456789X**

**5=ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ**

**6=ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ**

Somit ist WEingabe nun in der Lage, die Zeicheneingabe zu überwachen und die gewünschte Aussiebung zu besorgen. Was das Programm aber noch nicht kennt, ist die Breite der Eingabefelder, aus der hervorgeht, wieviele Zeichen eingegeben werden dürfen oder aber mindestens eingegeben werden müssen. Dazu dient der Abschnitt **SUB**. Die Einträge diesem Abschnitt folgen der gleichen Logik wie die Einträge im Abschnitt CODES, man gibt Eingabefeld für Eingabefeld die gewünschte Feldbreite an, nämlich:

**[SUB]**

**0=1**

Mit dem Eintrag 0=1 kann man einen Vorgabewert festlegen. Dieser gilt dann für alle Eingabefelder, die durch keinen eigenen Eintrag im Abschnitt [SUB] vertreten sind. Besitzen in einem Fragebogen alle Eingabefelder die gleiche Feldbreite, kann man so mit einem einzigen Eintrag alle Eingabefelder definieren. In unserem Fragebogenbeispiel bezieht sich der Ausdruck 0=1 auf die Felder 1,5, 6 und 9 für die, wie zu sehen ist, kein expliziter Eintrag existiert. Ihre Feldbreite wird auf 1 festgelegt. Da die Felder 1, 5, 6 und 9 im Abschnitt [CODES] als numerisch ausgewiesen sind, bedeutet dies, daß eine einstellige Zahl eingegeben werden muß.

**2,3,4=2**

**7,8,10THRU12=40**

**13=10**

**14=6**

Für das Eingabefeld 2 gilt, daß die einzugebende Zahl maximal 2 Stellen haben darf. Gleiches trifft zu für die Eingabefelder 3 und 4. Wegen der identischen Feldbreite dürfen die drei Felder in einer einzigen Zeile aufgezählt werden ( Einzelheiten dazu siehe S. 26).

Der Text in den Eingabefeldern 7, 8 und 10 bis 12 darf aus maximal 40 Zeichen bestehen. Mit dem Bereichsoperator THRU ( siehe S.26) kann man längere Aufzählungen durch Angabe von Intervallen ersetzen.

Das Eingabefeld 13 ist im Abschnitt [CODES] als numerisch deklariert. In ihm sollen die Daten des Itemkomplexes unter Frage 8 auf einer 5-stufigen Skala kodiert werden. Der Eintrag 13=10 legt fest, daß 10 Antworten zu kodieren sind bzw. 10 einzelne Variablenwerte einzugeben sind. Da zur Kodierung nur einstellige Zahlenwerte verwendet werden können, regelt der Eintrag zugleich, wieviele Zeichen insgesamt eingetippt werden müssen.

Die Abschnitte **[ITEMS]**, **[CODES]** und **[SUB]** bilden das Grundgerüst einer Initialisierungsdatei und liefern den minimalen Konfigurationsrahmen, der für eine sinnvoll strukturierte Dateneingabe von WEingabe benötigt wird. Das Optimum der vorhandenen Möglichkeiten wird durch diese Konfiguration allerdings lange nicht ausgeschöpft. Wie der folgende Überblick zeigt, bietet WEingabe eine ganze Reihe anderer nützlicher Optionen. Eine detaillierte Beschreibung aller Optionen finden Sie im Referenzteil ab Seite 21.

- Im Abschnitt [NAMES] können den Variablen Namen zugewiesen werden. Die Namen werden mit in die Datendatei geschrieben und können darin ohne Umweg nach SPSS exportiert werden. Soll beispielsweise die Variable in Eingabefeld 2 den Namen 'GEBOREN' erhalten, muß in der Initialisierungsdatei ein Abschnitt [NAMES] mit folgendem Eintrag vorhanden sein:

**[NAMES]**  
**2=GEBOREN**

Schließt ein Eingabefeld mehrere Variablen ein, wie in unserem Beispiel das Feld 13, wird für jede Variable ein "numerierter" Name gebildet, welcher die relative Position der Variablen im Eingabefeld wiedergibt. Lautet z.B. der Eintrag im Abschnitt [NAMES]

**13=SELBST**

bewirkt diese Deklaration, daß folgende Variablennamen gebildet werden:

SELBST01, SELBST02, SELBST03,.....,SELBST10.

Sie können für die Variablen auch andere Namen wählen, indem Sie folgenden eigenen Abschnitt hinzufügen:

**[NAMES13]**  
**3=NEUNAME**

Dadurch erhält die Variable an Position 3 des Eingabefeldes 13 nicht den Namen SELBST03, sondern wird in NEUNAME umbenannt.

- Im [RANGE] - Abschnitt lassen sich für numerische Variablen vom Typ @, d.h. Variablen mit ganzzahligem Wertebereich, die zulässigen Werte spezifizieren. Beispiel:

**[RANGE]2=73THRU80,99**

Diesem Eintrag zufolge sind alle befragten Personen nach 1972 und vor 1981 geboren, deswegen können auch nur Werte eingegeben werden, die entweder im Bereich 73-80 liegen oder den Missing-Value (99) darstellen.

- Variablen-Labels für den Export nach SPSS können im Abschnitt [LABEL] deklariert werden. Bei "gelabelten" Eingabefeldern wird der Label-Text darüber hinaus im Programmfenster angezeigt. Dadurch können Labels auch als Orientierungsstütze bei der Dateneingabe eingesetzt werden.

Labels werden nach den gleichen Regeln deklariert wie Variablennamen auch. Man kann sowohl eine Eingabefeld als Ganzes "labeln", als auch einzelne Positionen innerhalb eines Feldes mit einem Label versehen (siehe [LABEL#]-Abschnitt S.35, das Zeichen # ist Platzhalter für die Nummer eines Eingabefeldes).

- Die Abschnitte [MISSING] (S.39) und [MISSING#] (S.41) dienen der Definition von Missing-Values für den Export nach SPSS für WINDOWS sowie der Festlegung eines Vorgebewertes bei Skip-Aktionen.

- Mittels der Abschnitte [HELP] (S.42) und [HELP#] (S.43) läßt sich eine äußerst praktische On-Line-Hilfe realisieren. Sie können zu jedem Eingabefeld eine spezifische Benutzerinformation schreiben, die automatisch synchron zur Dateneingabe im Hilfefenster ausgegeben wird. Sogar für die einzelnen Eingabepositionen innerhalb eines Feldes läßt sich die Ausgabe differentieller Hilfetexte realisieren. Diese On-Line-Hilfe eignet sich ideal zur Instruktion über die bei der Dateneingabe zu befolgenden Kodierungsregeln. Auch erspart man sich auf diese Weise den Zeitaufwand, der sonst benötigt würde, um die mit der Dateneingabe befaßten Personen anzuleiten.

- Sie können Skip-Regeln formulieren. Skip-Regeln sind bedingte Sprunganweisungen, die angeben, welche Eingabefelder übersprungen werden können, falls für die aktuelle Variable ein bestimmter Wert eingegeben wird. Dazu dienen die Abschnitte [SKIP] (S.36) und [SKIPMESSAGE] (S.38).

- Sie können Eingabefelder durch Meldungsfenster ankündigen, in denen Sie dem Benutzer Hinweise geben und seine Aufmerksamkeit fokussieren können. Lesen Sie dazu den Abschnitt [MESSAGE] auf Seite 45.

- In den Abschnitten [AUTOFILL] und [AUTOFILL#] (S.47 und S.48) kann Eingabefeldern ein fester Wert zugewiesen werden, der automatisch eingefügt wird, wenn der Benutzer die **Taste F11** drückt.
- Im Abschnitt [CASELABEL] (S.49) kann im Hilfenfenster zu jedem Datensatz ein spezifischer Info-Text präsentiert werden
- Im Abschnitt [LEVEL] (S.50) können hierarchisch strukturierte Daten in Ebenen unterteilt werden.

Ein ausführliches Anwendungsbeispiel einer vollständigen Initialisierungsdatei mit allen oben besprochenen Konfigurationsmöglichkeiten bietet die im Anhang abgedruckte Initialisierungsdatei BEISPIEL.INI.

# Syntax-Referenz

## [CODES]-Abschnitt

Im [CODES]-Abschnitt können jedem Eingabefeld die bei der Dateneingabe zulässigen Codes zugewiesen werden. Zeichen, die der angegebenen Regel widersprechen, können nicht eingegeben werden. Die Einträge im [CODES]-Abschnitt definieren außerdem implizit den **Variablentyp** in dem jeweiligen Eingabefeld.

Es gibt 3 Typen von Eingabefeldern:

- 1) Typ "**String**",
- 2a) Typ "**einfach-numerisch**"
- 2b) Typ "**multipl-numerisch**".

1. Ein Eingabefeld ist vom Typ "**String**", wenn sich unter den zulässigen Codes mindestens ein Zeichen befindet, das keine Ziffer darstellt.

Eingabefelder vom Typ "**String**" sind dazu da, um einer **String**variablen (z.B. Religion, Berufsbezeichnung) "Werte" zuzuweisen. Die Länge der Stringvariablen kann im Abschnitt [SUB] (S.26) spezifiziert werden. Das Programm WEingabe erlaubt Längen, die von **1** bis **255** rangieren können. Ist bei der Dateneingabe die spezifizierte Anzahl an Zeichen erreicht, können keine weiteren Zeichen mehr eingegeben werden.

2. Ein Eingabefeld ist vom Typ "**numerisch**", wenn sich unter den zulässigen Codes ausschließlich Zeichen befinden, die Ziffern darstellen.

a) Ein "**einfach-numerisches**" Eingabefeld dient der Werteingabe für eine Variable, die als Kodierung eine beliebige, ein- oder mehrstellige Zahl erlaubt. Die Anzahl der Stellen (falls zulässig, einschließlich eines potentiellen Dezimaltrennzeichens, siehe dazu unten) kann im Abschnitt [SUB] spezifiziert werden. Bei negativen Zahlenwerten muß außerdem eine Stelle für das Minuszeichen eingerechnet werden (siehe dazu auch Abschnitt [SET], S. 54). WEingabe erlaubt Breiten, die von 1 bis 99 rangieren können. Ist bei der Dateneingabe die spezifizierte Anzahl an Stellen erreicht, können keine weiteren Ziffern mehr eingegeben werden.

Ein Eingabefeld ist vom Typ "einfach-numerisch", wenn auf der rechten Seite der Zuweisung keine Ziffernfolge, sondern das Zeichen @ (als Substitut für die Ziffernfolge 1234567890) steht.

b) Ein "**multipl - numerisches**" Eingabefeld ermöglicht es, die Werteingabe für **mehrere** Variablen in einem einzigen Eingabefeld zu lokalisieren. Es muß sich dabei um Variablen handeln, die **einstellig-numerisch** codiert werden müssen. Ein treffendes Beispiel sind Variablen, die Subskalen bzw. Subitems eines übergeordneten Items darstellen. In einem "multipl numerischen" Eingabefeld entspricht jede eingegebene Ziffer einem Variablenwert. Die Anzahl der Variablen eines Eingabefeldes kann im Abschnitt [SUB] spezifiziert werden. Die vom Programm WEingabe erlaubte Variablenzahl kann von **1** bis **99** rangieren.

Ein Eingabefeld ist immer dann vom Typ "**multipl numerisch**", wenn die rechte Seite der Zuweisung aus einer Zeichenfolge besteht, die nur Ziffern enthält.

Die Option, ein Eingabefeld als **multipl-numerisch** zu deklarieren, ist für die Dateneingabe sehr hilfreich, weil ein Editierfeld zur Verfügung gestellt wird, dessen Breite der Anzahl der einzugebenden Variablenwerte entspricht. Es können und müssen genauso viele Werte eingegeben werden, wie es Variablen im Eingabefeld gibt. Dies bedeutet, eine Eingabebestätigung durch Drücken von <RETURN> ist erst dann sowohl notwendig als auch wirksam, wenn die Werte aller Variablen im Eingabefeld vollständig eingegeben sind. Dies beschleunigt die Dateneingabe stark und verringert gleichzeitig die Fehlerwahrscheinlichkeit.

Ein weiterer Vorteil: Man erspart sich die Namengebung für die einzelnen Variablen, denn es genügt völlig (siehe Abschnitt [NAMES], S.29), dem ganzen Eingabefeld einen "Stammmamen" zuzuweisen. WEingabe ergänzt diesen Stammmamen bei Erstellen der Exportdatei durch ein numerierendes Suffix und generiert auf diese Weise selbsttätig die Namen der Unterskalen.

Der [CODES]-Abschnitt muß Eintragungen gemäß der folgenden Regel enthalten:

---

**<Nummer des Eingabefeldes>=<Bezeichnung zulässiger Codes>**

Die folgenden Beispiele zeigen die Variationsmöglichkeiten dieser Regel:

[CODES]

0=12349

Durch den Eintrag 0= wird allen Eingabefeldern, die keinen eigenen Eintrag besitzen, ein Vorgabewert zugewiesen, der festlegt, welche Zeichen zur Kodierung verwendet werden dürfen. Fehlt der Eintrag für den Vorgabewert, sind programmseitig die Zeichen **0 1 9** als zulässige Kodierung voreingestellt.

Der obige Eintrag legt fest, daß alle nicht eingetragenen Eingabefelder vom Typ "multipel-numerisch" sind, wobei die Werte 1 bis 4 oder der Wert 9 als Eingabe zulässig sind.

1=123459

Dieser Eintrag enthält die Zuweisung für Eingabefeld 1. Er legt fest, daß das Eingabefeld vom Typ "multipel - numerisch" ist, wobei die Werte 1 bis 5 oder der Wert 9 als Eingabe zulässig sind.

2=@

In Eingabefeld 2 sind alle Eingaben zulässig, deren Ergebnis eine **ganze** Zahl ist. Ein korrespondierender Eintrag im Abschnitt [SUB] (S. 26) legt die Anzahl der Stellen fest, die die Zahl maximal haben darf. Die eingegebene Zahl muß **positiv** sein, wenn nicht im Abschnitt [SET] durch den Eintrag MINUS=1 (siehe S. 54) die Eingabe negativer Vorzeichen ausdrücklich zugelassen ist.

3=@,

In Eingabefeld 2 sind alle Eingaben zulässig, deren Ergebnis eine **reelle** Zahl ergibt. Durch das **Komma** hinter dem Zeichen @ wird bewirkt, daß ein Dezimaltrennzeichen eingegeben werden kann. Ein korrespondierender Eintrag im Abschnitt [SUB] (S. 26) legt die Anzahl der Stellen (**einschließlich** der potentiellen Positionen für Dezimaltrennzeichen und Nachkommastellen) fest, die die Zahl maximal haben darf. Das Dezimaltrennzeichen kann mit der Komma-Taste auf der alphanumerischen Haupttastatur oder mit der Dezimaltaste auf dem numerischen Ziffernblock eingegeben werden, sofern nicht im Abschnitt [SET] durch den Eintrag DEZIMAL= (siehe S. 52) ein anderes Zeichen vereinbart wurde. Enthalten die einzugebenden Daten Zahlenwerte < 0, muß außerdem eine Stelle für das negative Vorzeichen eingerechnet werden. Bei Eingabefeldern, die mit =@, deklariert sind, ist im Unterschied zu ganzzahligen Eingabefeldern (siehe oben) die Eingabe negativer Vorzeichen **immer** möglich.

4=

Eingabefeld 4 ist vom Typ "String", da der Eintrag eine leere Zuweisung enthält. Die Regel lautet nämlich: Immer wenn ein leere Zuweisung auftritt, können alle "druckbaren" Zeichen eingegeben werden, d.h. Zeichen mit einem ASCII-Code von **32-127** sowie alle deutschen Umlaute. Die Länge des Strings kann durch einen korrespondierenden Eintrag im Abschnitt [SUB] (S. 26) festgelegt werden. Fehlt dieser Eintrag, gilt der programmseitig eingestellte Vorgabewert (Länge = 1).

**Vorsicht bei leeren Zuweisungen: Hinter dem Gleichheitszeichen darf sich kein Leerzeichen befinden.**

5=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ\_\*

Eingabefeld 5 ist ebenfalls vom Typ "String". Jedoch können nur Zeichen eingegeben werden, die in der angegebenen Zeichenmenge vorkommen. Es handelt sich in dem Beispiel um alle Klein- und Großbuchstaben sowie um die Zeichen \_ und \*. Man beachte, daß im obigen Beispiel rechts vom Gleichheitszeichen kein Leerzeichen vorkommt, was in diesem Fall bedeutet, daß es auch nicht eingegeben werden kann. Das gleiche gilt für alle Umlaute.

6=ABCDEFGHIJKLMNOPQRSTUVWXYZ

Die Zuweisung für Eingabefeld 6 schreibt als zulässige Eingabe ausschließlich Großbuchstaben vor. Normalerweise werden Großbuchstaben erzeugt, indem gleichzeitig mit der entsprechenden Buchstabentaste die SHIFT-Taste gedrückt wird. WEingabe besitzt jedoch eine automatische **Umwandlungsfunktion**, die die Betätigung der SHIFT-Taste überflüssig macht. Es **konvertiert** jeden eingegebenen Kleinbuchstaben selbsttätig in einen Großbuchstaben, wenn Kleinschreibung verboten, aber Großschreibung erlaubt ist.

7=123456789

Bei Eingabefeld 7 (Typ "multipel-numerisch") muß jede Werteingabe eine einstellige Zahl sein, die von 1 bis 9 variieren kann. Wieviele Werteingaben erlaubt sind, d.h. wieviele Variablen oder Subskalen das Eingabefeld umfaßt, wird durch den korrespondierenden Eintrag im Abschnitt [SUB] (S. 26) geregelt. Für Eingabefelder ohne eigenen [SUB]- Eintrag gilt der dort eingetragene Vorgabewert, oder wenn auch dieser fehlt, der programmseitig voreingestellte Wert (=1).

8=019

Eingabefeld 8 ist ebenfalls vom Typ "multipel-numerisch". Es umschließt 18 Unterskalen (siehe Eintrag 8=18 in den Beispielen zum Abschnitt [SUB], S.26), wobei die Skalen dichotomisiert sind (0,1). Der Wert 9 stellt den Missing-Value dar.

**Weitere Regeln:**

1. Gelten für mehrere Eingabefelder identische Codierungsvorschriften, kann die Angabe der entsprechenden Feldnummern auch als Aufzählung innerhalb einer einzigen Zeile durchgeführt werden, also:  
**2,5,7,8,10=123459**
2. Die Liste der Eingabefelder kann auch Bereichsangaben enthalten. Eine Bereichsangabe hat 3 Komponenten: niedrigster Wert - Bereichsoperator - höchster Wert. Bereichsoperator ist das Wort THRU (Großschreibung erforderlich !). Also:  
**2,5THRU8,10=123459**
3. Die Liste der Eingabefelder darf (einschließlich Kommata und THRU-Operatoren) nicht mehr als 2046 Zeichen umfassen.
4. Zur Beachtung:  
**Rechts vom Gleichheitszeichen (Zuweisungsoperator) befindliche Leerzeichen gelten immer als nicht-numerisches Zeichen und werden somit als zur Codierungsvorschrift gehörend aufgefaßt.** In der Deklaration **numerischer Eingabefelder** dürfen also keine Leerzeichen vorkommen.

Die folgende Tabelle zeigt die Beziehungen zwischen Variablentyp, Eigenschaften des Eingabefeldes und der funktionalen Bedeutung der in den Abschnitten [CODES], [CODES#], [SUB] und [RANGE] vorkommenden Deklarationen.

Variablentyp	Anzahl der Variablen pro Eingabefeld	zulässige Anzahl Zeichen pro Variablenwert	zulässiger Wertebereich	zulässige Breite des Eingabefeldes
mehrstellig-numerisch	1	[SUB]	[RANGE]	1-99
String	1	[SUB]	[CODES] [CODES#]	1-255
einstellig numerisch	[SUB]	1	[CODES] [CODES#]	1-99

## [CODES#]-Abschnitt

Die im [CODES]- Abschnitt (S. 52) stehenden Kodierungsregeln gelten immer für alle Elemente eines Eingabefeldes .

Der [CODES#]-Abschnitt dagegen erlaubt es, für jedes Element eines Eingabefeldes eine eigene Eingaberegeln bezüglich der zulässigen Codes zu spezifizieren. Das Zeichen # ist ein Platzhalter und ist durch die Nummer des betreffenden Eingabefeldes zu ersetzen.

Der [CODES#]-Abschnitt muß Eintragungen gemäß der folgenden Regel enthalten:

---

**<laufende Nummer des Eingabefeldes>=<Bezeichnung zulässiger Codes>**

Der Abschnitt [CODES#] ist z.B. dann erforderlich, wenn es sich bei den Elementen des Eingabefeldes um mehrere einstellig-numerische Variablen handelt, und wenn nicht für alle diese Variablen die gleichen Kodierungsregeln gelten.

Beispiel: Das Eingabefeld Nr. 12 umfasse 6 einstellig-numerische Variablen, von denen die Variablen 1, 3, 4, 5 auf einer fünfstufigen Skala erfaßt werden, während die Variablen 2 und 6 dichotomisiert sind und nur die Werte 0 und 1 annehmen können. In diesem Falle müssen die Kodierungsvorschriften für das Eingabefeld in einem eigenen Abschnitt [CODES12] folgendermaßen spezifiziert werden:

[CODES12]

1=12345

2=01

3=12345

4=12345

5=12345

6=01

Auch bei Stringvariablen sind Anwendungsfälle denkbar, beispielsweise dann, wenn an bestimmten Positionen nur ganz bestimmte Zeichen auftauchen dürfen, wie bei einem Kennwort, das aus der Zeichenfolge Großbuchstabe - Großbuchstabe - Ziffer - Ziffer - Großbuchstabe - Großbuchstabe bestehen soll. Angenommen Eingabefeld 12 enthielte eine solche Variable, dann wäre folgender Abschnitt notwendig;

[CODES12]

1=ABCDEFGHIJKLMNOPQRSTUVWXYZ

2=ABCDEFGHIJKLMNOPQRSTUVWXYZ

3=0123456789

4=0123456789

5=ABCDEFGHIJKLMNOPQRSTUVWXYZ

6=ABCDEFGHIJKLMNOPQRSTUVWXYZ

Es gelten folgende Regeln:

1. **Wichtig:** Im Abschnitt [CODES#] können zwar Zeichen spezifiziert werden, doch ergibt sich daraus **keine** Deklaration des Variablentyps (String oder numerisch). Der Variablentyp **muß** durch einen zusätzlichen Eintrag im Abschnitt [CODES] definiert werden. Für das Eingabefeld 12 im obigen Beispiel muß also im Abschnitt [CODES] ein expliziter oder impliziter Eintrag vorhanden sein, der das Feld als String-Typ kennzeichnet, also entweder

12=

oder als impliziter Eintrag

0=

Wie auf Seite 22 nachzulesen, wird durch den Eintrag 0= der Vorgabewert festgelegt. Fehlt dieser Eintrag, besteht der Vorgabewert aus den Zeichen 0 1 9.

2. Die Deklaration zulässiger Codes im Abschnitt [CODES] ist gültig für alle Elemente eines Eingabefeldes, für die im Abschnitt [CODES#] keine eigenen Codierungsvorschriften angegeben sind.
3. Folglich ist es sinnvoll, Variablen bzw. Feldelemente mit identischen Kodierungsvorschriften durch einen gemeinsamen Eintrag im Abschnitt [CODES] abzudecken, während im Abschnitt [CODES#] nur solche Variablen eingetragen werden, für die abweichende Kodierungsvorschriften gelten sollen

Dazu ein Beispiel:

```
[CODES]  
20=019
```

```
[CODES20]  
5=12345  
7=1234
```

4. Die Nummer des Eingabefeldes im Abschnittsbezeichner [CODES#] darf keine führenden Nullen enthalten.
5. Die Deklarationen der Feldelemente im Abschnitt [CODES#] dürfen - im Gegensatz zum Abschnitt [CODES] - **nicht** in Form einer Aufzählung geschehen, sondern müssen in getrennten Zeilen stehen. In jeder Zeile darf nur eine Nummer angegeben sein.

## [SUB]-Abschnitt

Im Abschnitt [SUB] wird die Breite der Eingabefelder festgelegt. Die Einträge geben an, wieviele Zeichen entweder eingegeben werden können oder aber eingegeben werden müssen, abhängig vom Typ des Eingabefeldes (siehe Abschnitt [CODES], Seite 21).

Ist das bezeichnete Eingabefeld vom Typ "einfach-numerisch", besagt der Eintrag, wieviele Stellen (falls zulässig, einschließlich eines potentiellen Dezimaltrennzeichens, siehe dazu Seite 22) die eingegebene Zahl maximal haben darf. Enthalten die Daten Zahlenwerte  $< 0$ , muß außerdem Platz für das negative Vorzeichen eingerechnet werden.

Ist das bezeichnete Eingabefeld vom Typ "String", ergibt der Eintrag die maximale Stringlänge.

Ist das bezeichnete Eingabefeld vom Typ "multipel-numerisch", beziffert der Eintrag die Anzahl aller Variablen oder Subskalen im Eingabefeld. Da es sich nur um einstellig-numerische Variablen handeln kann, bestimmt der Eintrag gleichzeitig, wieviele Zeichen insgesamt eingegeben werden müssen. Diese Grenze kann bei der Dateneingabe weder unter - noch überschritten werden. **Ist das Limit erreicht, werden weitere Eingaben blockiert. Ist das Limit noch nicht erreicht, kann das Eingabefeld nicht verlassen werden.**

Der höchste zulässige Zuweisungswert im Abschnitt [SUB] ist **99** für **numerische** Variablen und **255** für **String**-Variablen.

Die Breite des Eingabefensters wird der Anzahl der einzugebenden Zeichen oder Ziffern zwar soweit wie möglich angepaßt, hat aber eine natürliche Obergrenze. Diese liegt bei 45 bis 50 Zeichen. **Numerische** Eingabefelder lassen keine mehrzeilige Darstellung zu. Passen nicht alle Zeichen in eine Zeile, wird das Fenster zeilenweise "gescrollt", es ist also immer nur 1 Zeile sichtbar. Wer diesen Effekt vermeiden möchte, sollte die Dimensionalität numerischer Eingabefelder entsprechend knapper bemessen. Eingabefelder vom Typ "**String**" dagegen passen sich in Breite *und* Höhe an und wechseln bei Bedarf zu einem mehrzeiligen Darstellungsmodus über.

Der [SUB]-Abschnitt muß Eintragungen gemäß der folgenden Regel enthalten:

---

**<Nummer des Eingabefeldes>=<Anzahl der Elemente im Eingabefeld>**

Es folgen einige Beispiele.

[SUB]

0=1

Der Vorgabewert für alle nicht eingetragenen Eingabefelder ist 1. Dieser Wert ist auch programmintern voreingestellt für den Fall, daß der Eintrag **0=** fehlt.

2=3

Die eingegebene Wert für die Variable im Eingabefeld 2, die im Abschnitt [CODES] (Seite 21) als mehrstellig-numerisch ausgewiesen ist, kann bis zu 3 Stellen haben. Es kann sich somit um eine Zahl zwischen 0 und 999 handeln (weitergehende Restriktionen bezüglich des zulässigen Wertebereiches werden im Abschnitt [RANGE] (S. 33) geregelt).

4=30

Im Eingabefeld 4 können bis zu 30 Zeichen eingegeben werden. Im Abschnitt [CODES] (Seite 21) ist festgelegt, daß es sich um ein Eingabefeld vom Typ "String" handeln soll. Somit handelt es sich bei der Zuweisung um das Längenformat einer Stringvariablen.

8=18

Das Eingabefeld mit der Nummer 8 ist im Abschnitt [CODES] als "multipel-numerisch" deklariert. Der Wert 18 bedeutet, daß es 18 Variablen umfaßt und daß deswegen 18 aufeinanderfolgende einstellige Werte eingegeben werden müssen.

Zusätzlich gelten folgende Regeln:

1. Besitzen mehrere Eingabefelder die gleiche Breite, kann die Angabe der Feldnummern auch als Aufzählung innerhalb einer einzigen Zeile durchgeführt werden, also:  
2,5,7,8,10=20
2. Die Liste der Eingabefelder kann auch Bereichsangaben enthalten. Eine Bereichsangabe hat 3 Komponenten: niedrigster Wert - Bereichsoperator - höchster Wert. Bereichsoperator ist das Wort THRU (Großschreibung erforderlich !). Also:  
2,5THRU8,10=20
3. Die Liste der Eingabefelder darf (einschließlich Kommata und THRU-Operatoren) nicht mehr als 2046 Zeichen umfassen.
4. Wird einem numerischen Eingabefeld ein Wert zugewiesen, der größer ist als **99**, wird der Wert auf **1** zurückgesetzt.
5. Wird einem String-Eingabefeld ein Wert zugewiesen, der größer ist als **255**, wird der Wert auf **255** zurückgesetzt.

## **[ITEMS]-Abschnitt**

Dieser Abschnitt muß unbedingt vorhanden sein. Hier wird die Gesamtzahl aller Eingabefelder eingetragen. Der Maximalwert beträgt 5000.

Die Syntax für den Eintrag im [ITEMS]-Abschnitt lautet:

---

**N=<Anzahl der Eingabefelder>**

Beispiel:

[ITEMS]  
N=177

## [NAMES]-Abschnitt

Im Abschnitt [NAMES] können Variablennamen erzeugt werden, die zusammen mit den Daten in die Export-Datei (\*.DAT oder \*.SAV) einbezogen werden.

Der [NAMES]-Abschnitt muß Eintragungen gemäß der folgenden Regel enthalten:

---

**<Nummer des Eingabefeldes>=<Variablenname>**

Beispiel:

```
[NAMES]
1=STATUSM
2=BERUFM
3=STATUSV
4=BESCHV
```

Adressat der Namenszuweisung ist diejenige Variable, welche Referent des bezeichneten Eingabefeldes ist. Schließt das Eingabefeld mehrere Variablen ein, weil es vom Typ "multipel-numerisch" ist, **wird für jede Variable ein indizierter "Untername" erzeugt**, welcher die relative Position der Variablen im Eingabefeld wiedergibt.

Beispiel:

```
[NAMES]
12=ARBZD
```

```
[CODES]
12=01
```

```
[SUB]
12=5
```

Die obigen Deklarationen bewirken, daß folgende Variablennamen gebildet werden:  
**ARBZD01, ARBZD02, ARBZD03, ARBZD04, ARBZD05**

Die alternative Möglichkeit einer benutzerdefinierten Namenszuweisung an einige oder alle Variablen eines multipel-numerischen Eingabefeldes bietet der [NAMES#]-Abschnitt (siehe S. 31).

Zusätzlich gelten folgende Regeln:

1. Namen dürfen aus höchstens 8 Zeichen bestehen. Umfaßt ein Namenseintrag mehr als 8 Zeichen, so wird der Variablenname aus den ersten 8 Zeichen gebildet. Werden Unternamen gebildet (siehe oben), gilt zusätzlich die folgende Einschränkung:
2. Bei "multipel-numerischen" Eingabefeldern mit mehreren Variablen sollte der im Abschnitt [NAMES] eingetragene Name aus höchstens 6 Zeichen bestehen, da ansonsten die restlichen 2 Zeichen bei der Bildung von Unternamen (siehe oben) überschrieben werden.

3. Bei Variablen, für die es im Abschnitt [NAMES] keinen Namenseintrag gibt, wird folgendermaßen verfahren:
  - a) Zur Anzeige der Variablen auf dem Bildschirm während der Dateneingabe wird eine numerische Signatur verwendet, die aus der laufenden Nummer des Eingabefeldes, und - bei mehreren Variablen innerhalb eines Eingabefeldes - der Position der Variablen im Eingabefeld besteht.
  - b) Für den Datenexport nach SPSS oder anderen Programmen mittels der Exportdatei \*.DAT (siehe Seite 13) wird ein Standardname gebildet, nämlich "VAR####", z.B. VAR0005, VAR0032, VAR0155, wobei sich die Numerierung aus der Eingabereihenfolge ergibt.
4. Eine automatische Vergabe von Standardnamen an alle Variablen erreicht man folglich dadurch, daß man zwar den Abschnitt [NAMES] in die Konfigurationsdatei einfügt, aber keine Einträge macht, oder aber den Abschnitt [NAMES] ganz wegläßt.

## [NAMES#]- Abschnitt

Die im [NAMES]- Abschnitt (S. 29) stehenden Anweisungen für die Bildung von Variablennamen haben bei "multipel-numerischen" Eingabefeldern, die mehrere Variablen einschließen, den Nachteil, daß für alle Variablen im Eingabefeld ein standardisierter Unternamenname gebildet wird. Eine rein **benutzerdefinierte** Namenszuweisung an einige oder alle Variablen im Eingabefeld ist nicht möglich.

Diese Möglichkeit bietet der [NAMES#] - Abschnitt. Er erlaubt es, für jede Variable eines "multipel-numerischen" Eingabefeldes einen frei wählbaren Namen zu spezifizieren. Das Zeichen # ist ein Platzhalter und ist durch die Nummer des betreffenden Eingabefeldes zu ersetzen.

Der [NAMES#]-Abschnitt muß Eintragungen gemäß der folgenden Regel enthalten:

---

**<Laufende Variablennummer im Eingabefeld>=<Variablenname>**

Man vergleiche das folgende Beispiel mit dem oben angeführten Beispiel [NAMES] -Abschnitt:

```
[NAMES12]
1=ARBZD_A
2=ARBZD_B
3=ARBZD_C
4=ARBZD_D
5=ARBZD_E
```

```
[CODES]
12=01
```

```
[SUB]
12=5
```

Die obigen Deklarationen bewirken, daß folgende Variablennamen gebildet werden:  
ARBZD\_A, ARBZD\_B, ARBZD\_C, ARBZD\_D, ARBZD\_E

Auch eine Mischung beider Vorgehensweisen ist möglich  
Beispiel:

```
[NAMES]
12=ARBZD
```

```
[NAMES12]
1=ARBZD_A
3=ARBZD_C
5=ARBZD_E
```

```
[CODES]
12=01
```

```
[SUB]
12=5
```

Die obigen Deklarationen bewirken, daß folgende Variablennamen gebildet werden:  
ARBZD\_A, ARBZD02, ARBZD\_C, ARBZD04, ARBZD\_E

Wie das letzte Beispiel zeigt, gilt folgende Regel:

1. Namenszuweisungen im [NAMES#]-Abschnitt haben Vorrang vor Namenszuweisungen im [NAMES] - Abschnitt.
2. Für Variablen, die nicht im [NAMES#] - Abschnitt eingetragen sind, gelten die Deklarationen im [NAMES] - Abschnitt

Sonstige Regeln:

3. Namen dürfen aus höchstens 8 Zeichen bestehen. Namenseinträge, die länger als 8 Zeichen sind, sind ungültig.
4. Variablen, für die es sowohl im Abschnitt [NAMES] als auch im Abschnitt [NAMES#] keinen oder nur einen ungültigen Namenseintrag gibt, erhalten den Standardnamen "VAR####", z.B. VAR0014, VAR0032, VAR0155, wobei sich die Numerierung nach der Eingabereihenfolge richtet.
5. Die Nummer des Eingabefeldes im Abschnittsbezeichner [NAMES#] darf keine führenden Nullen enthalten

## [RANGE]-Abschnitt

Im [RANGE] - Abschnitt lassen sich für Eingabefelder vom Typ "einfach-numerisch" Regeln für zulässige Variablenwerte formulieren. Eingegebene Werte, die zu diesen Regeln in Widerspruch stehen, werden von WEingabe mit einer Fehlermeldung beantwortet und zurückgewiesen.

Der [RANGE]-Abschnitt muß Eintragungen nach der folgenden Regel enthalten:

---

**<Nummer des Eingabefeldes>=<Wert oder Wertebereich{,Wert oder Wertebereich}>**

Es gibt zwei Möglichkeiten, die zulässigen Werte anzugeben. Beide Möglichkeiten können miteinander kombiniert werden.

1. Man kann eine Folge von Einzelwerten auflisten. Die Werte müssen durch Kommata getrennt sein und die Liste darf keine Leerzeichen enthalten.

Beispiel:

[RANGE]

7=10,15,20,25,30

2. Man kann eine Folge von Wertebereichen auflisten. Die Bereichsangaben müssen ebenfalls durch Kommata getrennt werden. Es dürfen keine Leerzeichen vorkommen.

Eine Bereichsangabe hat drei Komponenten: niedrigster Wert - Bereichsoperator - höchster Wert.

Bereichsoperator ist das Wort THRU (**Großschreibung** erforderlich !). Die Bereichsangabe darf keine Leerzeichen enthalten.

Beispiel:

[RANGE]

7=1THRU5, 25THRU30

Das nächste Beispiel demonstriert eine Kombination der beiden Möglichkeiten:

[RANGE]

7=12, 1THRU5, 9,11, 25THRU30

Der Eintrag besagt, daß bei Eingabefeld 7 einer der folgenden Werte eingegeben werden muß: 1,2,3,4,5,9,11,12,25,26,27,28,29,30.

Sonstige Regeln:

1. Alle eingetragenen Werte müssen ganzzahlig sein.
2. Wie bereits mehrfach betont, dürfen auf der rechten Seite des Zuweisungsoperators = keine Leerzeichen stehen.
3. Nach dem letzten Wert in einer Zeile muß unmittelbar das <RETURN>-Zeichen folgen.
4. Besitzen mehrere Eingabefelder den gleichen Range, kann die Angabe der Feldnummern auch als Aufzählung innerhalb einer einzigen Zeile durchgeführt werden (die Aufzählung darf keine Leerzeichen enthalten), also:  
2,5,6,7,8,10=12, 1THRU5, 9,11
5. Die Aufzählung der Eingabefelder kann auch Bereichsangaben enthalten. Eine Bereichsangabe hat 3 Komponenten: niedrigster Wert - Bereichsoperator - höchster Wert. Bereichsoperator ist ebenfalls das Wort THRU (Großschreibung erforderlich !). Also:  
2,5THRU8,10=12, 1THRU5, 9,11
6. Alle Werte im [RANGE]-Abschnitt müssen ganzzahlig sein, Zahlen mit Nachkomma-Stellen werden ignoriert.
7. In welcher Reihenfolge die Werte angegeben sind, ist unwesentlich.

## [LABEL]-Abschnitt

Im [LABEL]-Abschnitt kann für jedes Eingabefeld ein "Label" formuliert werden. "Labels" sind inhaltliche Beschreibungen der einzugebenden Daten in knapper und verständlicher Form, die einen besseren Überblick verschaffen sollen.

Die Eintragungen im [LABEL] - Abschnitt erfüllen zwei Aufgaben:

1. "Gelabelte" Eingabefelder erscheinen im Programmfenster zusammen mit einer darüber eingeblendeten Titelzeile, die den Label-Text enthält. Auf diese Art und Weise fungieren Labels als wichtige Orientierungsstützen bei der Dateneingabe und helfen Irrtümer zu vermeiden.
2. Die "Labels" können als "**variable labels**" nach SPSS exportiert werden, entweder mittels einer Syntaxdatei, die WEingabe auf Wunsch erstellt, oder im Rahmen der Exportdatei \*.SAV. In SPSS dienen die Labels dazu, die Bedeutung der Variablen klarer zu machen und die Verständlichkeit von Tabellen und Diagrammen zu erhöhen. Die Prozedur für diesen Transfer ist an anderer Stelle beschrieben (siehe Abschnitt [SET], SPS=1, Seite 55 und SAV=1, Seite 57).

Der [LABEL]-Abschnitt muß Eintragungen entsprechend der folgenden Regel enthalten:

---

**<Nummer des Eingabefeldes>=<Label>**

Beispiel:

[LABEL]

1=Beruf Mutter

2=Beruf Vater

3=Alter der Mutter

4=Alter des Vaters

Der [LABEL]-Abschnitt eignet sich hauptsächlich für Eingabefelder, die nicht mehr als 1 Variable umfassen, was in der Regel nur auf Eingabefelder vom Typ "String" oder vom Typ "einfach- numerisch" zutreffen dürfte. Bei multipel-numerischen Eingabefeldern, die mehrere Variablen einschließen, kann der [LABEL]-Abschnitt zwar zur Beschriftung des Eingabefeldes benutzt werden, z.B. um die thematische Zusammengehörigkeit der einzelnen Variablen kenntlich zu machen, jedoch können keine variablenspezifischen "Labels" vergeben werden. Zu diesem Zweck muß zusätzlich der [LABEL#]-Abschnitt (Seite 35) verwendet werden.

Außerdem gelten folgende Regeln:

1. Labels sollten idealerweise nicht länger als 128 Zeichen (einschließlich Leerzeichen) sein, da dies ungefähr der Größe des Ausgabebereiches im Fenster entspricht. Längere Labels können meistens nicht in ganzer Länge wiedergegeben werden.
2. Zu einem Label gehören alle Zeichen zwischen dem Zuweisungsoperator = und dem <RETURN>-Zeichen.
3. In jeder Zeile darf nur 1 Feldnummer angegeben sein, d.h. Label-Deklarationen für verschiedene Eingabefelder müssen immer in getrennten Zeilen stehen.

## [LABEL#]-Abschnitt

Im [LABEL#] - Abschnitt kann für jede Variable in einem multipel-numerischen Eingabefeld ein eigenes "Label" definiert werden. Das Zeichen # ist dabei durch die Nummer des betreffenden Eingabefeldes zu ersetzen. "Labels" sind knappe inhaltliche Charakterisierungen der einzugehenden Daten.

Die Eintragungen im [LABEL#] - Abschnitt erfüllen zwei Funktionen:

1. Sie dienen der Orientierung bei der Dateneingabe, die Variablen eines Eingabefeldes können mit Hilfe von Labels besser identifiziert werden, da diese im Programmfenster angezeigt werden. So ist immer ersichtlich, mit welcher Variablen die Dateneingabe fortgesetzt werden muß.
2. Die "Labels" können als "**variable labels**" nach SPSS übertragen werden, entweder mittels einer Syntaxdatei, die WEingabe auf Wunsch erstellt, oder im Rahmen der Exportdatei \*.SAV. In SPSS dienen die Labels dazu, die Bedeutung der Variablen klarer zu machen und die Verständlichkeit von Tabellen und Diagrammen zu erhöhen. Die Prozedur für diesen Transfer ist an anderer Stelle beschrieben (siehe Abschnitt [SET], SPS=1, Seite 55 und SAV=1, Seite 57).

Der [LABEL#]-Abschnitt muß Eintragungen gemäß der folgenden Regel enthalten:

---

**<laufende Variablennummer im Eingabefeld>=<Label>**

Beispiel:

```
[LABEL12]
1=mit meiner Mutter
2=mit meinem Vater
3=mit meiner Schwester
4=mit meinem Bruder
5=mit meinem Lehrer
```

Besitzen die Variablen einen thematischen Zusammenhang, kann zusätzlich auch der Abschnitt [LABEL] (S. 34) benutzt werden, um das Eingabefeld global zu charakterisieren.

Beispiel:

```
12=Wie gerne redest Du mit folgenden Personen
```

Es gelten außerdem folgende Regeln:

1. Eintragungen im [LABEL#] - Abschnitt sind wirkungslos bei Eingabefeldern vom Typ "String", vom Typ "einfach-numerisch" sowie bei Eingabefeldern vom Typ "multipel-numerisch", sofern diese nur eine einzige Variable enthalten. In solchen Fällen muß das Label immer im Abschnitt [LABEL] deklariert werden.
2. Ist ein Eingabefeld mit mehreren Variablen belegt, müssen die Variablen dieses Feldes, deren label als "VARIABLE LABEL" nach SPSS übertragen werden soll, immer im [LABEL#]-Abschnitt eingetragen sein.
3. Labels im [LABEL#]-Abschnitt sollten idealerweise nicht länger als 80 Zeichen (einschließlich Leerzeichen) sein, da dies ungefähr der Größe des Ausgabebereiches im Fenster entspricht. Längere Labels können meistens nicht in ganzer Länge wiedergegeben werden.
4. Zu einem Label gehören alle Zeichen zwischen dem Zuweisungsoperator = und dem <RETURN>-Zeichen.
5. In jeder Zeile darf nur 1 Feldnummer angegeben sein, d.h. Label-Deklarationen für verschiedene Eingabefelder müssen immer in getrennten Zeilen stehen.

## [SKIP]-Abschnitt

Ein Fragebogen kann Fragen enthalten, deren Beantwortbarkeit davon abhängt, wie eine vorausgegangene Frage beantwortet wurde. Wenn beispielsweise die Frage lautet: "Wieviele Kinder haben Sie ?", können nachfolgende Fragen nach dem Alter der einzelnen Kinder nur von Personen beantwortet werden, die 1 oder mehrere Kinder haben.

Der [SKIP]-Abschnitt enthält bedingte Sprunganweisungen, die angeben, mit welcher Variablen, d.h. an welcher Stelle des Fragebogens die Eingabeprozedur fortgesetzt werden soll, falls der eingegebene Wert für die aktuelle Variable eine bestimmte Größe besitzt

Eine Anweisung im [SKIP]-Abschnitt muß folgenden Aufbau besitzen:

---

**<Nummer des Ausgangsfeldes>=<Nummer des Sprungziel><Bedingung>**

Es folgen einige elementare Demonstrationsbeispiele:

1=9(0)

Falls der eingegebene Wert für die Variable im Eingabefeld 1 gleich 0 ist, wird die Eingabe bei der Variablen fortgesetzt, die erstes Element von Eingabefeld 9 ist.

2=5(OHNE,KEIN)

Bei der Variablen des Eingabefeldes 2 handelt es sich um eine Stringvariable. Die Sprunganweisung zu Eingabefeld 5 wird ausgeführt, wenn entweder 'OHNE' oder 'KEIN' als Wert eingegeben wurde.

3=11(0,1),12(9, 2THRU5)

Falls der eingegebene Wert für die Variable im Eingabefeld 3 gleich 0 oder 1 ist, erfolgt ein Sprung zu Eingabefeld 11, falls der eingegebene Wert gleich 9 oder eine Zahl von 2 bis 5 ist, wird das Eingabefeld 12 angesprungen.

Es können also einem Ausgangsfeld auch mehrere Sprungziele zugeordnet werden, wobei die einzelnen Zuordnungen durch ein Komma voneinander zu trennen sind.

Bei der Angabe der Bedingungen gelten überwiegend die gleichen Syntaxregeln wie bei der Deklaration von Missing-Values im Abschnitt [MISSING] (S. 39), nämlich

1. Erlaubt sind beliebige Kombinationen einzelner Variablenwerte und (nur bei numerischen Variablen) Bereichsangaben mittels der Operators THRU (Großschreibung erforderlich !).
2. Die Bereichsangabe muß folgende Syntax aufweisen: <erster Wert>THRU<letzter Wert>. Die Bereichsangabe darf keine Leerzeichen enthalten.
3. Alle Werte und Bereichsangaben, aus denen sich der Bedingungs Ausdruck zusammensetzt, sind durch Kommata zu trennen. Der Bedingungs Ausdruck muß von runden Klammern umschlossen sein und darf, wie überhaupt die ganze Zeile, keine Leerzeichen enthalten.

In den obigen Ausdrücken ist das SKIP-Kriterium immer der Wert des Eingabefeldes, von dem aus der Sprung erfolgt. Man kann aber auch<sup>1</sup> den Wert anderer Eingabefelder als Sprungkriterium festlegen. In diesem Fall muss die Anweisung so aussehen:

5=10(2=99)

5= 10(2=99), 11(88)

oder:

5=10(2=99), 10(3=22) ;*falsch* ist 5=10(2=99,3=22)

oder

5=10(2=99,88), 11(3=11THRU15)

Im in runde Klammern eingeschlossenen Bedingungsteil steht links vom Gleichheitszeichen die Nummer eines Eingabefeldes und rechts davon der Wert oder die Werte, die vorliegen müssen, damit der Sprung erfolgt. Beide Anweisungstypen können auch gemischt auftreten (Beispiel 2), es können mehrere Felder in Form einer Sowohl-als auch-Bedingung aufgeführt sein (Beispiel 3) und die Werte können wie bei einfachen Sprunganweisungen auch als Liste oder Bereich angegeben sein (Beispiel 4).

---

<sup>1</sup> nur Premium-Version

Es gelten außerdem folgende Regeln:

1. Basis für Sprunganweisungen kann jedes beliebige Eingabefeld sein, mit **Ausnahme** des letzten Eingabefeldes.
2. Sollen Eingabefelder, die mehr als 1 Variable enthalten (Typ multipel-numerisch), als Basis für Sprunganweisungen benutzt werden, muß jede einzelne SKIP-Bedingung als **Folge** fester Variablenwerte formuliert sein, also z.B. **12=24(11225),1(99999,00000)** bei einem Eingabefeld mit 5 Variablen. Die Zahlenfolge muß so viele Werte enthalten wie das Eingabefeld Variablen enthält. Die Sprunganweisung wird nur dann ausgeführt, wenn **jede** Variable im Eingabefeld den für sie vorgegebenen Wert aufweist.
3. Die Angaben für Sprungziele und Bedingungen dürfen pro Zeile insgesamt nicht länger als 2048 Zeichen sein.
4. Zwischen dem Sprungziel und der Bedingung darf auf keinen Fall ein Leerzeichen stehen.
5. In jeder Zeile darf nur 1 Ausgangsfeld angegeben sein.
6. Sprungziel einer Skip-Anweisung kann auch das erste Eingabefeld des nächsten Datensatzes sein. Beispiel: 77=1(9). Ansonsten sind als Sprungziel nur Felder des gleichen Datensatzes erlaubt.
7. Die Nummer des Sprungzieles muß größer sein als die Nummer des Ausgangsfeldes. Eine Ausnahme bildet der Sprung zum **ersten** Eingabefeld des **nächsten** Datensatzes. Lautet das Sprungziel "1", wird immer an den Beginn des nächsten Datensatzes gesprungen.
8. Den Variablen, die übersprungen werden, wird immer der Wert zugewiesen, der entweder im Abschnitt [MISSING] (S. 39) oder im Abschnitt [MISSING#] (S. 41) als Missing-Value definiert ist.
9. Für alle Variablen, die übersprungen werden sollen, muß deswegen entweder im Abschnitt [MISSING] oder im Abschnitt [MISSING#] ein Missing-Value definiert sein. Ist diese Voraussetzung nicht erfüllt, wird die Sprunganweisung nicht ausgeführt. Falls für eine Variable mehrere Missing-Values definiert sind, wird immer der zuerst angegebene Wert verwendet.
10. WEingabe verwendet die in den Abschnitten [MISSING] und [SKIP] angegebenen Werte ohne zu prüfen, ob diese den in den Abschnitten [CODES] und [RANGE] vorgeschriebenen Bedingungen genügen. Es wird nur geprüft, ob die im Abschnitt [SUB] stehenden Angaben erfüllt sind. Überschreitet die Zeichenanzahl eines Missing-Value den dort eingetragenen Wert, werden überzählige Zeichen ignoriert.

## [SKIPMESSAGE]-Abschnitt

Es ist empfehlenswert wenn nicht sogar erforderlich, dem Benutzer einen Hinweis darauf zu geben, mit welcher Variablen oder an welcher Stelle des Fragebogens er die Eingabe fortsetzen soll, wenn die normale Eingabereihenfolge aufgrund einer Sprunganweisung (siehe Abschnitt [SKIP], S. 36) unterbrochen wird. Diesem Zweck dient der Abschnitt [SKIPMESSAGE].

Die Syntax lautet:

---

**<Nummer des Ausgangsfeldes>=<Text>**

Beispiel:

10=Folgende Fragen überspringen ?//Wenn ja, weiter mit Frage 22 auf Seite 5 oben.

Dieser Eintrag bewirkt, daß immer dann ein Meldungsfenster erscheint, wenn in das Eingabefeld 10 ein Wert eingetippt wurde, für den im Abschnitt [SKIP] eine Sprunganweisung existiert. Inhalt des Fensters ist die rechts vom Gleichheitszeichen stehende Instruktion. Außerdem sind zwei Schaltflächen "**JA**" und "**NEIN**" vorhanden. Wählt der Benutzer den **JA**-Button, springt das Programm unmittelbar zu dem Eingabefeld, das im Abschnitt [SKIP] als Sprungziel deklariert wurde, im anderen Falle unterbleibt der Sprung und die Dateneingabe wird mit Feld 11 fortgesetzt.

Zusätzlich gelten folgende Regeln:

1. Der Text im Meldungsfenster kann bis zu 2046 Zeichen umfassen.
2. Soll der Text mit Zeilenumbrüchen dargestellt werden, sind die gewünschten Stellen durch zwei unmittelbar aufeinanderfolgende Schrägstriche zu kennzeichnen (siehe obiges Beispiel).
3. Ist mehreren Eingabefeldern das gleiche Sprungziel zugeordnet, können die Nummern der Eingabefelder auch in einer einzigen Zeile, getrennt durch einzelne Kommata, zusammengefaßt werden, z.B.  
2,5,6,7,8=Weiter mit Frage 22  
Die Aufzählung kann auch Bereichsangaben enthalten, z.B.  
2,5THRU8=Weiter mit Frage 22

Eine andere und vor allem bequemere Methode, darauf aufmerksam zu machen, daß Fragen zu überspringen sind, bietet der Eintrag SKIPMESSAGE=1 im Abschnitt [SET]. Er bewirkt universell, daß vor jeder Ausführung einer SKIP-Anweisung eine standardisierte Instruktion ausgegeben wird, mit welchem Datenfeld die Eingabe fortzusetzen ist. Beispiel: "Weiter mit Feld 1 im nächsten Datensatz". Einzelheiten dazu finden Sie auf Seite 54.

Beide Methoden können auch vermischt werden, da sich der Eintrag SKIPMESSAGE=1 nur bei Eingabefeldern auswirkt, für die im Abschnitt [SKIPMESSAGE] kein eigener Eintrag vorliegt.

## [MISSING]-Abschnitt

Der [Missing]-Abschnitt dient der Definition von "Missing-Values". Hier können jene Variablen eingetragen werden, für die beim Erstellen der Variablendefinitionen in der Syntax-Datei (siehe [SET]-Abschnitt) ein MISSING VALUES - Kommando geschrieben werden soll. Der Abschnitt ist außerdem erforderlich für die Anwendung von SKIP-Regeln (siehe S. 36) sowie bei Verwendung des [LEVEL]-Abschnittes (siehe Seite 50) .

Der [MISSING]-Abschnitt muß Eintragungen gemäß der folgenden Regel enthalten (die in geschweiften Klammern stehenden Bezeichner sind optional):

---

**<Nummer des Eingabefeldes{, Nummer des Eingabefeldes}>=<Missing-Values>**

Die folgenden Beispiele zeigen die Variationsmöglichkeiten dieser Regel:

[MISSING]

1=9

Missing-Value für die Variable(n) im Eingabefeld 1 ist der Wert 9

2,5,6,8=1

Missing-Value für die Variable(n) in den Eingabefeldern 2,5,6 und 8 ist der Wert 1

3,4=7,8,9

Missing-Values für die Variable(n) in den Eingabefeldern 3 und 4 sind die Werte 7, 8 und 9

9=0,9 THRU 12,15

Missing-Values für die Variable(n) im Eingabefeld 9 sind die Werte 0, 9 bis 12 und 15

Bei der Angabe von **Wertebereichen** mittels des Operators THRU muß - als Ausnahme von der sonstigen Regel- vor und nach dem THRU ein Leerzeichen stehen.

10=OHNE

Die Variable im Eingabefeld 10 ist eine Stringvariable. Missing-Value ist das Wort 'OHNE'.

1THRU5,7,10THRU20=9

Missing-Value für die Variable(n) in den Eingabefeldern 1 bis 5, 7, 10 bis 20 ist der Wert 9

1. Besitzen mehrere Eingabefelder den gleichen Missing-Value, kann die Deklaration der Eingabefelder auch in Form einer Aufzählung erfolgen (siehe Beispiel 2).
2. Die Aufzählung der Eingabefelder kann auch Bereichsangaben enthalten (siehe letztes Beispiel oben) Eine Bereichsangabe hat 3 Komponenten: niedrigster Wert - Bereichsoperator - höchster Wert. Bereichsoperator ist das Wort THRU (Großschreibung erforderlich!).
3. Stringvariablen, für die ein MISSING VALUES-Kommando erstellt werden soll, müssen vom Typ "short string" sein, d.h. ihr 'Wert' darf aus nicht mehr als 8 Zeichen bestehen.
4. Rechts vom Gleichheitszeichen dürfen (einschließlich Kommata) jeweils nicht mehr als 256 Zeichen stehen.
5. Während numerischen Variablen mehrere Missing-Values\* zugeordnet werden können, dürfen Stringvariablen nur einen einzigen Missing-Value besitzen.
6. Missing-values für numerische Variablen müssen immer ganzzahlig sein.
7. Es erfolgt keine Gültigkeitsprüfung des Formates der deklarierten Missing-Values. WEingabe verwendet die angegebenen Werte ohne zu prüfen, ob diese den in den Abschnitten [CODES] und [RANGE] vorgeschriebenen Bedingungen genügen. Es wird nur geprüft, ob die im Abschnitt [SUB] stehenden Angaben erfüllt sind. Überschreitet die Zeichenanzahl eines Missing-Value den dort eingetragenen Wert, werden überzählige Zeichen beim Einlesen abgeschnitten.

---

\* SPSS für Windows akzeptiert entweder bis zu 3 einzelne Werte oder 1 Bereichsangabe und und 1 Einzelwert.

Der [MISSING]-Abschnitt dient aber nicht nur der Definition von missing-values. Bei SKIP-Operationen kann es, besonders bei hierarchisch strukturierten Datensätzen, erforderlich sein, daß die übersprungenen Eingabefelder nicht mit dem vorgegebenen missing value ausgefüllt werden, sondern der im vorherigen Datensatz eingegebene Wert wiederholt wird. Daher kann man im [MISSING]-Abschnitt nicht nur Werte definieren, sondern *stattdessen* ein **Eingabefeld** bezeichnen<sup>2</sup>, dessen Wert in das bei einer Skip-Operation übersprungene Feld übertragen werden soll.

Zu diesen Zweck muss im [MISSING]-Abschnitt die Nummer von Eingabefeldern zusammen mit einem Verweis auf ein Feld, das den einzugebenden Wert liefert, eingetragen werden. Der Verweis kann entweder ein *anderes* (vorhergehendes) Feld im *gleichen* Datensatz oder das *gleiche* Feld in einem *früheren* Datensatz betreffen. Er besteht aus dem Zeichen @, unmittelbar gefolgt von der Nummer eines Eingabefeldes oder einer Positionsangabe für einen Datensatz.

---

**<Nummer des Eingabefeldes{, Nummer des Eingabefeldes}>=@<Feldverweis>**

Die folgenden Beispiele zeigen Anwendungsmöglichkeiten dieser Regel:

**[SKIP]**

34=37(99)

**[MISSING]**

.....

35=@ 15

36=@ 15

oder auch:

35,36@15

.....

Beim Überspringen von Feld 35 erhält es automatisch den Wert von Feld 15 im gleichen Datensatz zugewiesen.

**[SKIP]**

10=20(5)

**[MISSING]**

11,12,13,14,15,16,17,18,19=@-1

oder auch:

11THRU19=@-1

Eine **negative** Zahl hinter dem @ bezeichnet keine Feldnummer, sondern gibt die relative Position des **Datensatzes** an, auf den WEingabe zugreift, um den Wert zu holen. In diesem Fall füllt WEingabe beim Überspringen der Felder 11-19 diese mit dem korrespondierenden Wert im N - 1ten, also unmittelbar vorausgehenden Datensatz aus. Auch Angaben wie @-2 (Datensatz N - 2), @-3 (Datensatz N - 3) usw. sind gültig.

Wirksam sind außerdem folgende Regeln:

1. Werte, die aus Verweisen resultieren, gelten nicht als missing value und werden daher in der Variablendefinition für SPSS auch nicht als solche gekennzeichnet bzw. exportiert.
2. In einer Zeile kann entweder ein Wert oder ein Verweis stehen. Alle Einträge, in denen unmittelbar hinter = ein @ folgt, gelten als Verweis, aller anderen als Wert. Für multipelnumerische Eingabefelder (mehr als 1 Variable) sind nur Wertzuweisungen zulässig.
3. Logischerweise kann nur auf vorangehende Felder oder vorangehende Datensätze verwiesen werden.
4. Bei fehlerhaften Zuweisungen (falsche Feldnummern) ist dafür gesorgt, dass keine nicht-numerischen Zeichen in numerische Eingabefelder geschrieben werden und die vorgegebene Feldbreite nicht überschritten wird. Weitere Typen-Checks erfolgen nicht.

---

<sup>2</sup> nur Premium-Version

## [MISSING#]-Abschnitt

Der [Missing#]-Abschnitt dient der Spezifikation unterschiedlicher Missing-Values innerhalb eines einzelnen Eingabefeldes. Das Zeichen # ist ein Platzhalter für die Nummer des betreffenden Eingabefeldes. Dieser Abschnitt ist erforderlich, wenn es sich bei den Elementen des Eingabefeldes um mehrere einstellig-numerische Variablen handelt, und wenn nicht alle diese Variablen den gleichen Missing-Value aufweisen.

Die Syntax lautet:

---

**<laufende Variablennummer im Eingabefeld>=<Missing-Values>**

Beispiel: Das Eingabefeld Nr. 5 umfasse 8 einstellig numerische Variablen mit den Missing-Values 9,9,0,0,0,9,9,1. In diesem Falle gibt es zwei Alternativen, um die Missing-Values zu deklarieren:

Alternative A:

```
[MISSING5]
1=9
2=9
3=0
4=0
5=0
6=9
7=9
8=1
```

Alternative B:

```
[MISSING]
5=9
[MISSING5]
3,4,5=0    oder    3THRU5=0
8=1
```

1. Besitzen mehrere Variablen eines Eingabefeldes den gleichen Missing-Value, kann die Deklaration der Variablennummern auch in Form einer Aufzählung erfolgen.
2. Die Aufzählung der Variablennummern kann auch Bereichsangaben enthalten. Eine Bereichsangabe hat 3 Komponenten: niedrigster Wert - Bereichsoperator - höchster Wert. Bereichsoperator ist das Wort THRU (Großschreibung erforderlich!).
3. Numerischen Variablen können auch mehrere Missing-Values\* zugeordnet werden, Stringvariablen dürfen jedoch nur einen einzigen Missing-Value besitzen.
4. Missing-values für numerische Variablen müssen immer ganzzahlig sein.

---

\* SPSS für Windows akzeptiert entweder bis zu 3 einzelne Werte oder 1 Bereichsangabe und und 1 Einzelwert.

## [HELP]-Abschnitt

Der [HELP] - Abschnitt (sowie der [HELP#]- Abschnitt, siehe Seite 43) stellt bei der Dateneingabe eine praktische Online-Hilfe zur Verfügung. WEingabe präsentiert dem Benutzer ein ständig sichtbares Hilfenfenster. Im [HELP] - Abschnitt kann jedem Eingabefeld ein passender Hilfetext zugewiesen werden, der automatisch im Hilfenfenster erscheint, solange das zugehörige Eingabefeld aktiviert ist. Hinweise und Anleitungen für die Dateneingabe, z.B. Informationen darüber, wie die Daten eines Fragebogens bei der Eingabe zu verschlüsseln sind, lassen sich auf diese Weise optimal vermitteln.

Der [HELP]-Abschnitt muß Eintragungen gemäß der folgenden Regel enthalten:

---

**<Nummer des Eingabefeldes>=<beliebiger Text>**

Beispiel:

[HELP]

1=Rechtschreibfehler bitte nicht übernehmen

3,4,6,17=Kodierung: JA =1, NEIN = 0, FEHLT = 9

Zuweisungen, die im [HELP]- Abschnitt stehen, beziehen sich immer auf das ganze Eingabefeld. Soll sich ein Hilfetext nur auf einzelne Elemente eines Eingabefeldes beziehen, muß dazu der Abschnitt [HELP#] verwendet werden (S. 43). Dieser Fall ist normalerweise dann gegeben, wenn das Eingabefeld mehrere Variablen einschließt, weil es vom Typ "multipel-numerisch" ist, und wenn diesen Variablen unterschiedliche Hilfetexte zugeordnet werden sollen.

Zusätzlich gelten folgende Regeln:

1. Der Hilfetext für ein Eingabefeld kann bis zu 2046 Zeichen umfassen. Darüber hinausgehender Text erscheint nicht im Hilfenfenster.
2. Zum Hilfetext gehören sämtliche Zeichen zwischen dem Zuweisungsoperator = und dem <RETURN>-Zeichen. Längere Hilfetexte müssen also ohne dazwischen geschaltetes <RETURN> - Zeichen geschrieben werden, wenn sie vollständig ausgegeben werden sollen. Der Zeilenumbruch im Hilfenfenster erfolgt automatisch in Ausrichtung am rechten Fensterand. Die Breite des Hilfenfensters kann vom Benutzer vergrößert oder verkleinert werden.
3. Falls der automatische Zeilenumbruch eine unübersichtliche optische Gliederung des Hilfetextes mit sich bringt, können zusätzlich auch erzwungene Zeilenumbrüche festgelegt werden. Dazu sind die Stellen, an denen ein Zeilenumbruch erfolgen soll, durch zwei hintereinander gestellte Schrägstriche // zu kennzeichnen. (das <RETURN>-Zeichen kann zu diesem Zweck nicht verwendet werden, da es - siehe oben - das Ende des ganzen Hilfetextes markiert).
4. Ist der Hilfetext für mehrere Eingabefelder der gleiche, kann die Zuweisung auch pauschal in Form einer Aufzählung der Eingabefelder durchgeführt werden (die Aufzählung darf keine Leerzeichen enthalten), also:  
2,5,6,7,8,10,122,155=HILFETEXT
5. Die Aufzählung der Eingabefelder kann auch Bereichsangaben enthalten. Eine Bereichsangabe hat 3 Komponenten: niedrigster Wert - Bereichsoperator - höchster Wert. Bereichsoperator ist das Wort THRU (Großschreibung erforderlich !). Also:  
2,5THRU8,10,122,155=HILFETEXT

## **[HELP#]-Abschnitt**

Der [HELP#] - Abschnitt erweitert die im [HELP]-Abschnitt (S. 42) zur Verfügung stehenden Möglichkeiten der Online-Hilfe bei der Dateneingabe. Im [HELP#] - Abschnitt kann jeder beliebigen Eingabeposition innerhalb eines Eingabefeldes ein eigener Hilfetext zugeordnet werden. Das Zeichen # ist ein Platzhalter und ist durch die Nummer des betreffenden Eingabefeldes zu ersetzen. Der Hilfetext erscheint automatisch im Hilfefenster, solange sich der Eingabefokus an der bezeichneten Position des betreffenden Eingabefeldes befindet.

Der Sinn und Zweck des [HELP#]-Abschnittes wird besonders deutlich bei multipelnumerischen Eingabefeldern mit mehreren Variablen, für die separate Hilfetexte zur Verfügung stehen sollen.

Der [HELP#]-Abschnitt muß Eintragungen gemäß der folgenden Regel enthalten:

---

**<laufende Nummer der Eingabeposition>=<beliebiger Text>**

Beispiel:

[HELP12]

1=Kodierung: linkes Kästchen = 1, dann 2,3,4, rechtes Kästchen = 5

2=Kodierung: JA =1, NEIN = 0, FEHLT = 9

3=Kodierung: JA =1, NEIN = 0, FEHLT = 9

4=Kodierung: JA =1, NEIN = 0, FEHLT =

5=Kodierung: Angaben < 10 = 0; Angaben > 10 = 1

6=Kodierung: Angaben in Schritten von 100 aufrunden

7=Kodierung: Angaben in Schritten von 100 aufrunden

8=Kodierung: Angaben in Schritten von 100 aufrunden

9=Kodierung: JA =1, NEIN = 0, FEHLT = 9

oder:

[HELP12]

1=Kodierung: linkes Kästchen = 1, dann 2,3,4, rechtes Kästchen = 5

2THRU4,9=Kodierung: JA =1, Nein = 0, FEHLT = 9

5=Kodierung: Angaben < 10 = 0; Angaben > 10 = 1

6THRU8=Kodierung: Angaben in Schritten von 100 aufrunden

Auch ein Mischverfahren ist möglich:

Beispiel:

[HELP]

12=Kodierung: JA =1, NEIN = 0, FEHLT = 9

[HELP12]

1=Kodierung: linkes Kästchen = 1, dann 2,3,4, rechtes Kästchen = 5

5=Kodierung: Angaben < 10 = 0; Angaben > 10 = 1

6THRU8= Kodierung: Angaben in Schritten von 100 aufrunden

Es gelten folgende Regeln:

1. Zuweisungen im [HELP#]-Abschnitt haben Vorrang vor Zuweisungen im [HELP] - Abschnitt.
2. An Eingabepositionen, für die es im [HELP#] - Abschnitt keinen eigenen Eintrag gibt, gelten die pauschalen Zuweisungen im [HELP]-Abschnitt.
3. Der Hilfetext für ein Eingabefeld kann bis zu 2046 Zeichen umfassen. Darüber hinausgehender Text erscheint nicht im Hilfefenster.

4. Zum Hilfetext gehören sämtliche Zeichen zwischen dem Zuweisungsoperator = und dem <RETURN>-Zeichen. Längere Hilfetexte müssen also ohne dazwischen geschaltetes <RETURN> Zeichen geschrieben werden, wenn sie vollständig ausgegeben werden sollen. Der Zeilenumbruch im Hilfefenster erfolgt automatisch in Ausrichtung am rechten Fenster-  
rand.
5. Um den Hilfetext nach eigenen Vorstellungen zu strukturieren, können Zeilenumbrüche auch erzwungen werden. Dazu sind die Stellen, an denen ein Zeilenumbruch erfolgen soll, durch zwei hintereinander gestellte Schrägstriche // zu kennzeichnen. (das <RETURN>-Zeichen kann zu diesem Zweck nicht verwendet werden, da es das Ende des Hilfetextes markiert).
6. Soll bei mehreren Variablen eines Eingabefeldes der gleiche Hilfetext erscheinen, kann die Deklaration der Variablennummern auch in Form einer Aufzählung erfolgen. Die Aufzählung kann auch Bereichsangaben enthalten (siehe obiges Beispiel).

## **[MESSAGE]-Abschnitt**

Manchmal kann es zweckmäßiger sein, wenn man einen wichtigen Hinweis zur Dateneingabe nicht im Hilfefenster präsentiert, sondern den Benutzer gezielt darauf aufmerksam macht. Diesem Zweck dient der Abschnitt [MESSAGE].

Er ermöglicht Eingabefelder auszuwählen, bei deren Aufruf zunächst ein (modales) Meldungsfenster erscheint, das die gewünschte Hinweisse enthält. Erst wenn der Benutzer die Meldung (hoffentlich) gelesen und dies durch Tastendruck rückgemeldet hat, verschwindet das Fenster vom Bildschirm und die Dateneingabe kann fortgesetzt werden.

Die Syntax lautet:

---

**<Nummer des Eingabefeldes{,Nummer des Eingabefeldes}>=<Text>**

Folgende Einträge beispielsweise bewirken, daß immer dann ein Meldungsfenster geöffnet wird, wenn ein Eingabefeld mit einer der links vom Gleichheitszeichen stehenden Nummern aufgerufen wird. Inhalt der Meldung ist der rechts vom Gleichheitszeichen stehende Text.

[MESSAGE]

10=Folgende 5 Fragen besonders sorgfältig codieren!//Bitte die Fragen erst durchlesen !

20=Weiter mit Satzergänzungstest ab Seite 4

50,71,115,117,158,168=Bitte Umblättern

131=Weiter auf Seite 19

170=Bald ist es geschafft !

Zusätzlich gelten folgende Regeln:

1. Zur Meldung gehören sämtliche Zeichen zwischen dem Zuweisungsoperator = und dem <RETURN>-Zeichen. Längere Texte müssen also ohne dazwischen geschaltetes <RETURN>-Zeichen geschrieben werden, wenn sie vollständig im Meldungsfenster erscheinen sollen.
2. Soll der Text mit Zeilenumbrüchen dargestellt werden, sind die gewünschten Stellen durch zwei unmittelbar aufeinanderfolgende Schrägstriche zu kennzeichnen (siehe obiges Beispiel, erste Zeile).
3. Text, der länger als 2046 Zeichen ist, wird bei der Wiedergabe abgeschnitten.
4. Ist die Meldung für mehrere Eingabefelder die gleiche, können die Nummern der Eingabefelder auch in einer einzigen Zeile, getrennt durch einzelne Kommata, zusammengefaßt werden, z.B.  
2,5,6,8,10,122,155=MELDUNG  
Die Aufzählung kann auch Bereichsangaben enthalten, z.B.  
2,5THRU8,10=MELDUNG

## [ALIAS]-Abschnitt

Bei umfangreichen Vorhaben mit sehr vielen Eingabefeldern, aber sich oft wiederholenden Codierungsregeln, kann das Schreiben der Zuweisungen im Abschnitt [CODES], auch wenn man sich der Hilfsmittel Kopieren und Einfügen bedient, eine monotone Tätigkeit sein. Eine Möglichkeit der Abhilfe liegt in der Benutzung von Aufzählungen (siehe S. 21). Eine andere bzw. zusätzliche Möglichkeit bietet der Abschnitt [ALIAS]. Hier können Sie Zeichenmengen mit einem Kürzel belegen. Dieses Kürzel kann dann in den Abschnitten [CODES] und [CODES#] rechts vom Gleichheitszeichen anstelle der originalen Zeichenfolge als Zuweisung eingesetzt werden.

Betrachten wir das folgende Beispiel:

```
[CODES]
1=123456789
2=ABCDEFGHJKLMNOPQRSTUWXYZ ÄÖÜ-
3=ABCDEFGHJKLMNOPQRSTUWXYZ ÄÖÜ-
4=123456789
5=123456789
6=ABCDEFGHJKLMNOPQRSTUWXYZ ÄÖÜ-
7=ABCDEFGHJKLMNOPQRSTUWXYZ ÄÖÜ-
8=123456789
9=ABCDEFGHJKLMNOPQRSTUWXYZ ÄÖÜ-
```

Wenn Sie aber einen **Abschnitt [ALIAS]** mit folgenden Einträgen in die Initialisierungsdatei einfügen:

```
[ALIAS]
Z=123456789
X= ABCDEFGHJKLMNOPQRSTUWXYZ ÄÖÜ-
```

kann der Abschnitt [CODES] folgendermaßen aussehen:

```
[CODES]
1=Z
2=X
3=X
4=Z
5=Z
6=X
7=X
8=Z
9=X
```

oder sogar:

```
[CODES]
1,4,5,8=Z
2,3,6,7,9=X
```

Es gelten folgende Regeln:

1. Jede Zeichenfolge im Abschnitt [CODES(#)] rechts des Zuweisungsoperators, die identisch ist mit einer im Abschnitt [ALIAS] links des Zuweisungsoperators vorkommenden Zeichenfolge, wird als Alias behandelt. Leerzeichen gelten als Bestandteil einer Zeichenfolge.
2. Ein Aliasname darf aus beliebigen Zeichen bestehen.
3. Die Bezeichnung zulässiger Codes im Abschnitt [ALIAS] folgt den gleichen Regeln wie im Abschnitt [CODES(#)]. Leerzeichen gelten also immer als Code.
4. Dem Eintrag **0=** im Abschnitt [CODES] kann kein Alias zugewiesen werden.

## [AUTOFILL]-Abschnitt<sup>3</sup>

Je nach Datenstruktur kann es Eingabefelder geben, bei denen immer ein gleicher, vorher festgelegter Wert eingegeben oder der Wert aus einem vorangegangenen Datensatz übernommen werden kann. Man denke an eine ID, die über mehrere Datensätze hinweg die gleiche bleibt.

Mit WEingabe kann solchen Feldern ein Wert zugewiesen werden, der automatisch eingefügt wird, wenn der Benutzer die **Taste F11** drückt. Die Zuweisung kann entweder explizit erfolgen oder implizit durch einen Verweis auf den Wert eines anderen Feld im gleichen oder in früheren Datensätzen.

Der [AUTOFILL]-Abschnitt enthält die Nummern von Eingabefeldern zusammen mit einer Wertangabe *oder* einem Verweis auf ein Feld, das den einzugebenden Wert liefert. Der Verweis kann entweder ein anderes (vorhergehendes) Feld im gleichen Datensatz oder das gleiche Feld in einem früheren Datensatz betreffen.

Eine Anweisung im [AUTOFILL]-Abschnitt muß folgenden Aufbau besitzen:

---

**<Nummer des Eingabefeldes{, Nummer des Eingabefeldes}>=<Wert>oder<Feldverweis>**

Es folgen einige elementare Demonstrationsbeispiele:

1=22

5=OHNE

Hier handelt es sich um eine Wertzuweisung für die Felder 1 und 5. Dem (numerischen) Feld 1 wird der Wert 22 zugewiesen, dem alphanumerischen Feld 5 die Zeichenkette OHNE.

Verweise auf andere Eingabefelder müssen immer mit dem **Zeichen @** beginnen

35=@15

Hier erkennt WEingabe, dass es sich um keine Wertzuweisung handelt, sondern um einen Verweis auf das Eingabefeld Nr. 15 im **gleichen** Datensatz. Daher wird im Eingabefeld 35, wenn die Taste F11 gedrückt wird, automatisch der Wert von Eingabefeld 15 eingefügt.

40=@-1

Eine **negative** Zahl hinter dem @ bezeichnet keine Feldnummer, sondern gibt die relative Position des **Datensatzes** an, auf den WEingabe zugreifen soll, um den Wert zu holen. In diesem Fall füllt WEingabe, sobald Taste F11 gedrückt wird, Feld 40 mit dem Wert von Feld 40 im N – 1ten, also unmittelbar vorausgehenden Datensatz aus. Auch Angaben wie @-2 (Datensatz N - 2), @-3 (Datensatz N - 3) usw. sind gültig.

Es gelten außerdem folgende Regeln:

5. Der AUTOFILL-Mechanismus funktioniert nur im letzten Datensatz und **nicht** bei Eingabefeldern, die mehr als eine Variable enthalten, also vom Typ "multipel-numerisch" sind.
6. Logischerweise kann nur auf vorangehende Felder oder vorangehende Datensätze verwiesen werden.
7. Bei fehlerhaften Zuweisungen (falsche Feldnummern) ist dafür gesorgt, dass numerische Eingabefelder nicht mit Werten aus String-Feldern gefüllt werden, weitere Typen-Checks erfolgen nicht.
8. Ist mehreren Eingabefeldern der gleiche AUTOFILL-Wert oder Feldverweis zugeordnet, können die Nummern der Eingabefelder auch in einer einzigen Zeile, getrennt durch einzelne Kommata (1,4,17,33=) zusammengefasst werden, auch Bereichsangaben (z.B. 3THRU7) sind möglich.

---

<sup>3</sup> Nur Premium-Version

## [AUTOFILL#]-Abschnitt<sup>4</sup>

Der Abschnitt [AUTOFILL#] erfüllt im Prinzip die gleiche Funktion wie der Abschnitt [AUTOFILL], spezifiziert jedoch Wertzuweisungen für nur ein **einzelnes** Eingabefeld. Das Zeichen # ist ein Platzhalter und ist in der Initialisierungsdatei durch die Nummer eines Eingabefeldes zu ersetzen. Mit den Einträgen in diesem Abschnitt kann man einem Eingabefeld **datensatzspezifische** Werte zuweisen, die beim Drücken der Taste **F11** automatisch eingefügt werden. Alle in diesem Abschnitt stehenden Wertzuweisungen beziehen sich also nicht auf Eingabefelder, sondern auf Datensätze. Für jedes Eingabefeld, das eine automatisierte Wertzuweisung erhalten soll, muss ein eigener [AUTOFILL#]Abschnitt vorhanden sein.

Der [AUTOFILL#]-Abschnitt enthält die Nummern von Datensätzen zusammen mit einer Wertangabe. Eine Anweisung im [AUTOFILL#]-Abschnitt muß folgenden Aufbau besitzen:

---

**<Nummer des Datensatzes{, Nummer des Datensatzes}>=<Wert>**

Ein Beispiel:

[AUTOFILL1]

1=Life of Brian

2=A Fish named Wanda

3=Road To Welville

Hier enthält Eingabefeld 1 den Titel eines Filmes, die Eingabe könnten Filmexperten sein, die die 3 Filme angeschaut haben und nun ein Rating durchführen sollen, wobei für jeden Film ein anderer Datensatz angelegt wird.

Es gelten außerdem folgende Regeln:

1. Der [AUTOFILL#] – Abschnitt kann keine Verweise auf andere Eingabefelder enthalten. Alle Einträge, auch wenn sie mit @ beginnen, werden als explizite Wertzuweisungen interpretiert.
2. Der Mechanismus funktioniert nur im letzten Datensatz und **nicht** bei Eingabefeldern, die mehr als eine Variable enthalten, also vom Typ "multipel-numerisch" sind.
3. Existiert ein Abschnitt [AUTOFILL#], so hat dieser Vorrang vor Einträgen im Abschnitt [AUTOFILL]
4. Ist mehreren Datensätzen der gleiche AUTOFILL-Wert zugeordnet, können die Nummern der Datensätze auch in einer einzigen Zeile, getrennt durch einzelne Kommata (1,4,17,33=) zusammengefasst werden, auch Bereichsangaben (z.B. 3THRU7) sind möglich.

---

<sup>4</sup> Nur Premium-Version

## [CASELABEL]-Abschnitt

Im Abschnitt [CASELABEL] kann jedem Datensatz ("case" in der Terminologie von SPSS) oder einigen Datensätzen ein spezifischer Text, z.B. eine Erläuterung oder Charakterisierung des Datensatzes zugewiesen werden, der während der Eingabe im **Hilfefenster** an oberster Stelle erscheint und dort permanent sichtbar bleibt. Der Unterschied zum [AUTOFILL#]-Abschnitt liegt darin, dass hier die Zuweisung feldunspezifisch erfolgt und nicht Bestandteil der Daten wird..

Der [CASELABEL]-Abschnitt enthält die Nummern von Datensätzen zusammen mit einem Text. Eine Anweisung im [CASELABEL]-Abschnitt muß folgenden Aufbau besitzen:

---

**<Nummer des Datensatzes{, Nummer des Datensatzes}>=<Text>**

Ein Beispiel:

[CASELABEL]

10,20,30,40,50,60=Uhrzeit aufschreiben

Hier sollen die Eingabe daran erinnert werden, bei jedem zehnten Datensatz die Uhrzeit aufzuschreiben, beispielsweise weil man den Zeitbedarf für die Dateneingabe ermitteln will.

Es gelten außerdem folgende Regeln:

1. Einträge im Abschnitt [CASELABEL] werden nur berücksichtigt, wenn im Abschnitt [SET] der Eintrag CASELABEL=1 vorhanden ist.
2. Wie das obige Beispiel zeigt, können die Nummern der Datensätze auch in einer einzigen Zeile, getrennt durch einzelne Kommata zusammengefasst werden, auch Bereichsangaben (z.B. 5THRU10) sind möglich.

**Alternativ** kann der Text auch aus einer externen Datei im ASCII-Format eingelesen werden. Dazu muss der Eintrag CASELABEL= im Abschnitt [SET] den **Dateinamen ohne Pfadangabe** aufweisen. Die Datei selber muss sich im **gleichen** Verzeichnis befinden wie die Initialisierungsdatei. Sie muss in Abschnitte unterteilt sein, an deren Anfang in einer eigenen Zeile ein in eckigen Klammern geschriebener, beliebiger **Abschnittsname** steht.

Daher bezeichnen in dieser Variante die Einträge im [CASELABEL]-Abschnitt nicht den auszugebenden Text, sondern den Namen des Textabschnittes, aus dem der Text für einen Datensatz entnommen werden soll.

---

**<Nummer des Datensatzes{, Nummer des Datensatzes}>=<Abschnittsname>**

Ein Beispiel:

[CASELABEL]

1=VP123

2=XX235

3=VP123

.....

Der dazu gehörige Eintrag im Abschnitt [SET] könnte lauten:

[SET]

CASELABEL=Caselabels.txt

Der Text, der je Datensatz im Hilfefenster eingeblendet werden soll, befindet sich hier in der Datei Caselabels.txt im Verzeichnis der INI-Datei, und zwar für den ersten Datensatz im Abschnitt [VP123], für den zweiten Datensatz im Abschnitt [XX235] usw.

Mit dieser Variante könnte man z.B. direkt in WEingabe die offenen Antworten eines Fragebogens einem Rating unterziehen, man muss dazu die INI-Datei nur so anlegen, dass jede Antwort als eigener Datensatz kodiert wird.

## [LEVEL]<sup>5</sup>-Abschnitt

Hierarchisch aufgebaute Eingabe-Daten sind dadurch gekennzeichnet, dass sich die Datensätze nur hinsichtlich solcher Daten unterscheiden, die in und unterhalb der Ebene liegen, auf der sich die Datensätze selbst unterscheiden, bei den Daten höherer Ebenen aber identisch sind. Man betrachte etwa folgendes Eingabeschema:

Ebene 1	Schule <i>ID</i> <i>Typ</i> <i>Schülerzahl</i> .....	N = 50
Ebene 2	Klasse <i>ID</i> <i>Schülerzahl</i> <i>Mittlerer Status</i> .....	N = 5
Ebene 3	Fach <i>ID</i> <i>Wochenstunden</i> .....	N = 2
Ebene 4	Klassenarbeit <i>ID</i> <i>Reihenfolge</i> <i>Zahl der Aufgaben</i> .....	N = 3
Ebene 5	Hauptaufgabe <i>ID</i> <i>Reihenfolge</i> <i>Art</i> .....	N = 2
Ebene 6	Teilaufgabe <i>ID</i> <i>Lösungswahrscheinlichkeit</i> <i>Wissensstufe</i> ..... .....	N = 3

Es gibt 6 Kodiereinheiten (Ebenen), das N jeder Ebene gibt die Zahl der Fälle innerhalb der nächsthöheren Ebene an, die kursiv gesetzten Zeilen repräsentieren Variablen (Eingabefelder). Es sind insgesamt  $50 \cdot 5 \cdot 2 \cdot 3 \cdot 2 \cdot 3 = 9000$  Datensätze zu bearbeiten, wobei nur auf Ebene 6 in jedem Datensatz neue Daten eingegeben werden müssen. Dagegen gibt es z.B. je Schule 180 Datensätze, die auf Ebene 1 identisch sind, oder je Klasse 36 Datensätze, die auf Ebene 1 und 2 identisch sind, d.h. die ID für die Klasse und die Werte für die Variablen *Schülerzahl* und *mittlerer Status* (Ebene 2) ändern sich nur in jedem 36ten Datensatz. Es wäre deswegen ein SKIP-Mechanismus nützlich, der es bei der Eingabe erlaubt, ganze Ebenen zu überspringen, und gleichzeitig automatisch für das Ausfüllen der übersprungenen Datenfelder sorgt.

Diese Funktion erfüllt der Abschnitt [LEVEL]. Im Unterschied zum [SKIP]-Abschnitt, in dem Werte von Eingabefeldern als Skip-Bedingung festgelegt sind (siehe Seite 36), kann man im [LEVEL]-Abschnitt die Eingabefelder direkt bestimmen, die Ausgangspunkt einer Skip-Aktion sein können.

---

<sup>5</sup> nur Premium-Version

Der [LEVEL]-Abschnitt enthält die Ebenen eines Datensatzes zusammen mit der Nummer des Eingabefeldes, das erstes Element dieser Ebene ist. Aus allen im Abschnitt [LEVEL] aufgeführten Eingabefeldern kann jederzeit ein Sprung zu einem der anderen in diesem Abschnitt stehenden Eingabefelder durchgeführt werden, sofern dieses zu einer Ebene unterhalb der derzeitigen Ebene gehört. Der Sprungbefehl erfolgt durch Drücken der Taste **F12** oder **STRG+F12**. Zunächst kann man in einem Dialogfenster das Sprungziel auswählen, danach erfolgt der Sprung. Der Befehl ist nur wirksam, wenn die im [SKIP]-Abschnitt beschriebenen Regeln eingehalten werden, insbesondere muss für alle Eingabefelder der übersprungenen Ebenen im [MISSING]-Abschnitt ein Wert oder ein Verweis auf einen anderen Datensatz bzw. anderes Eingabefeld zu finden sein (Siehe Seite 39).

Eine Anweisung im [LEVEL]-Abschnitt muß folgenden Aufbau besitzen:

---

**<Nummer des Ebene>=<Nummer des Eingabefeldes>**

Das folgende Beispiel demonstriert, wie man die Eingabe für einen Datensatz der eingangs dargestellten Art konfigurieren könnte. Angenommen werden 6 Ebenen mit je 1 Eingabefeld für die ID und 2 Feldern für andere Variablen.

```
[LEVEL]
1=1
2=4
3=7
4=10
5=13
6=16
```

Die Eingabefelder, mit denen die 6 Ebenen beginnen, haben die Nummern 1,4,7,10,13,16. Aus Feld 1 kann unmittelbar zu den Feldern 4,7,10,13,16, aus Feld 7 zu den Feldern 10,13,16 gesprungen werden usw.

```
[MISSING]
1THRU16=@-1
```

Hier wird festgelegt, dass übersprungene Eingabefelder immer mit dem Wert aus dem unmittelbar vorangehenden Datensatz ausgefüllt werden. Beachten Sie, dass nach dieser Logik im ersten Datensatz keine Sprünge möglich sind,

Außerdem gelten folgende Regeln:

Es kann auch jederzeit an den Anfang des nächsten Datensatzes gesprungen werden, das Dialogfenster zur Wahl des Sprungzieles enthält außer den Nummern der Eingabefelder auch die Option "nächster Datensatz" .

## [SET]-Abschnitt

Der [SET]-Abschnitt ist für Eintragungen bestimmt, die sich auf den allgemeinen Programmablauf auswirken:

Der [SET]-Abschnitt kann die folgenden Einstellungen enthalten:

---

### TITEL

Syntax:

**TITEL=<beliebiger Text>**

**Standard:** ohne

**Zweck:** Legt eine kurze Überschrift fest, die während der gesamten Dateneingabe oben links im Programmfenster angezeigt wird. Die Überschrift kann beliebigen Text enthalten, sollte aber aus nicht mehr als **128** Zeichen (einschließlich Leerzeichen) bestehen, da ein zu breiter Text bei der Ausgabe abgeschnitten wird.

---

### DELIMITER

Syntax:

**DELIMITER=<Zeichen>**

**Standard:** Tabulatorzeichen

**Zweck:** Legt das Trennzeichen fest, mit dem die einzelnen Datenfelder voneinander abgegrenzt werden, wenn die Exportdatei \*.DAT erstellt wird. Fehlt ein solcher Eintrag, wird standardmäßig das Tabulatorzeichen benutzt. Ändern Sie diesen Vorgabewert nur, wenn Sie ein Importprogramm verwenden, das einen anderen Feldbegrenzer als das Tabulatorzeichen verlangt. Ändern Sie diesen Vorgabewert nicht, wenn Sie einen Datenexport mittels einer \*.DAT-Datei nach **SPSS** für **WINDOWS** beabsichtigen.

---

### DEZIMAL

Syntax:

**DEZIMAL=<Dezimaltrennzeichen>**

**Standard:** Komma

**Zweck:** Legt das Symbol fest, mit dem in der Datendatei Nachkommastellen von der ganzen Zahl getrennt werden sollen. Sie können somit das Komma durch den Punkt ersetzen, wenn Sie folgenden Eintrag einfügen:

DEZIMAL=.

**ACHTUNG:** Bei Datenexport in eine SAV-Datei (SAV=1) **muss** als Dezimalzeichen der Punkt (.) verwendet werden, es muss also der Eintrag **DEZIMAL=.** vorhanden sein.

Ansonsten richten Sie sich bei der Wahl des Dezimaltrennzeichens nach den aktuellen Einstellungen des Systems oder des Programms, wohin die Daten später exportiert werden sollen. Welches Dezimaltrennzeichen ein Programm verwendet, hängt meistens von den Ländereinstellungen in der Windows-Systemsteuerung ab. Ist ein Datenexport nach SPSS für WINDOWS vorgesehen, und Sie verwenden in Ihrer Datendatei das Komma als Dezimaltrennzeichen, müssen sie dafür sorgen, daß auch beim Daten**import** in der Windows-Systemsteuerung das Komma als Dezimalzeichen eingetragen ist.

---

## AUTO

Syntax:

**AUTO=<0 oder 1>**

**Standard:** 0 (= nein)

**Zweck:** Simulation der Dateneingabe. Wird AUTO auf den Wert **1** gesetzt, lassen sich mit den Tasten **F11** und **F12** zufällige Datenwerte erzeugen. Drücken Sie **F11**, um ein einzelnes Eingabefeld auszufüllen, und **F12**, um einen ganzen Datensatz zu erzeugen.

Verwenden Sie diese Option, wenn Sie die Initialisierungsdatei auf einwandfreies Funktionieren testen und die dazu notwendige lästige Tipparbeit verringern wollen. Eine fehlerhafte Initialisierungsdatei kann nicht nur die Dateneingabe behindern, sondern auch die Lauffähigkeit von WEingabe blockieren. Errechnet werden die Zufallswerte immer (und nur) nach Maßgabe der in den Abschnitten [ITEMS], [CODES], [CODES#], [SUB], [NAMES] und [NAMES#] festgelegten Vorschriften. Andere Abschnitte, z.B. der [SKIP]-Abschnitt, bleiben unberücksichtigt und können systematisch nur getestet werden, indem man selber Daten eintippt. Insbesondere erfolgt im AUTO-Modus keine Range-Überprüfung. Um Einträge im Abschnitt [RANGE] zu testen, muß der Wert für AUTO auf **0** gesetzt werden.

Bei eingeschaltetem AUTO-Modus erscheint in der Titelleiste des Programmfensters der Name der Initialisierungsdatei zusammen mit dem Zusatz "im Auto-Modus!!!", also z.B. "[FRAGEBOGEN.INI im Auto-Modus!!!!]". Achten Sie darauf, den AUTO-Modus vor Beginn der eigentlichen Dateneingabe zu deaktivieren.

Die Option AUTO ersetzt die Kommandozeilenparameter TEST und SIMU, die in der Vorgängerversion **WEingabe für Windows 3.1** einen ähnlichen Zweck erfüllten.

---

## NEXT

Syntax:

**NEXT=<0 oder 1>**

**Standard:** 0 (= nein)

**Zweck:** Aktiviert die Funktionstaste <**F12**> als **Hotkey**, der benutzt werden kann, um zum nächsten Datensatz zu springen, wobei den übersprungenen Eingabefeldern bzw. Variablen des aktuellen Datensatzes automatisch Missing-Values zugeordnet werden.

Der Sprung kann aus jedem beliebigen Eingabefeld heraus erfolgen, vorausgesetzt es ist für jede der zu überspringenden Variablen mindestens ein Missing-Value, entweder im Abschnitt [MISSING] (S. 39) oder im Abschnitt [MISSING#] (S. 41) definiert. Nach welchen Regeln die Definition von Missing-Values zu geschehen hat, kann an den oben bezeichneten Stellen ausführlich nachgelesen werden. Das Sprungkommando ist außerdem nur wirksam, wenn das Eingabefeld, von dem aus der Sprung erfolgen soll, entweder korrekt ausgefüllt oder aber leer ist.

Die Sprung erfolgt erst, wenn der Benutzer eine Sicherheitsabfrage ("Weiter mit Feld 1 im nächsten Datensatz?") mit JA beantwortet.

Ein solcher Hotkey kann sehr praktisch sein, beispielsweise bei Fragebögen, bei denen ein Teil der Befragten die Beantwortung vorzeitig abgebrochen hat. Auf der anderen Seite birgt er auch die Gefahr des Mißbrauchs in sich.

Der Eintrag NEXT=1 ist **unwirksam**, wenn ein [LEVEL]-Abschnitt existiert. In diesem Fall ändert sich die Funktion von F12, näheres erfahren Sie auf Seite 50.

---

## SKIPMESSAGE

Syntax:

**SKIPMESSAGE=<0 oder 1>**

**Standard:** 0 (= nein)

**Zweck:** Enthält der Abschnitt [SET] den Eintrag SKIPMESSAGE=1, wird vor jeder Ausführung einer SKIP-Anweisung (siehe Seite 36) eine standardisierte Instruktion ausgegeben, mit welchem Eingabefeld die Eingabe fortzusetzen ist. Beispiel: "Weiter mit Feld 1 im nächsten Datensatz". Sofern Felder mit einem Label (siehe dazu die Abschnitte [LABEL] auf Seite 34 und [LABEL#] auf Seite 35) versehen sind, wird dieses in den Text mit eingebunden, also z.B. "Weiter mit Feld 1 ("Bundesland") im nächsten Datensatz". Das Fenster, in dem die Instruktion erscheint, enthält außerdem zwei mit "**JA**" und "**NEIN**" bezeichnete Schaltflächen. Wählt der Benutzer den **JA**-Button, wird der Sprung ausgeführt, im anderen Falle unterbleibt er.

Für SKIP-Zwecke, die spezifischere Instruktionen erfordern, als sie der obige Standardtext bereitstellt, können auch variable Anweisungen formuliert werden. Diese Möglichkeit bietet der Abschnitt [ SKIPMESSAGE] auf Seite 38.

---

## MINUS

Syntax:

**MINUS=<0 oder 1>**

**Standard:** 0 (= nein)

**Zweck:** Enthält der Abschnitt [SET] den Eintrag MINUS=1, können bei Eingabefeldern, die mit =@ deklariert sind (siehe dazu Abschnitt [CODES], S. 21), Ganzzahlen auch mit **negativem** Vorzeichen eingetippt werden. Standardmäßig sind bei der Dateneingabe die Tasten, mit denen ein Minuszeichen eingegeben werden kann, gesperrt, um Tippfehlern vorzubeugen (auf dem numerischen Tastaturblock z.B. liegen die Tasten für die 9 und das Minuszeichen in unmittelbarer Nachbarschaft). Der Eintrag MINUS=1 hebt diese Sperre auf.

## SPS

Syntax:

**SPS=<0 oder 1>**

**Standard:** 0 (= nein)

**Zweck:** Legt fest, ob zusammen mit der Datendatei eine **Syntax**-Datei **\*.SPS** in der Kommando-sprache von SPSS für WINDOWS erstellt werden soll. Die **Syntax**-Datei dient der Variablen-  
definition innerhalb von SPSS und beschreibt für die im eingelesenen Datenfile vorkommenden  
Variablen deren **Typ**, **Label** und **Missing-Values**. Für genauere Informationen zur Handhabung  
von **Syntax**-Dateien sei auf die Handbücher von **SPSS** für **WINDOWS** verwiesen.

Das folgende Beispiel verdeutlicht den Sachverhalt. Angenommen, eine Variable sei in der  
Initialisierungsdatei folgendermaßen deklariert:

```
[CODES]
```

```
1=019
```

```
[SUB]
```

```
1=10
```

```
[NAMES1]
```

```
3=VARNAME
```

```
[LABEL1]
```

```
3=VARLABEL
```

```
[MISSING]
```

```
1=9
```

WEingabe erkennt anhand dieser Deklarationen, daß die Variable mit dem Namen "VARNAME"  
einstellig-numerisch ist, daß sie als Label die Bezeichnung "VARLABEL" und als "Missing-  
Value" den Wert 9 zugeordnet bekommen soll. Infolgedessen schreibt es folgende Kommandos  
in die Syntax-Datei \*.SPS :

```
VARIABLE LABELS VARNAME "VARLABEL".
```

```
FORMATS VARNAME (F1).
```

```
MISSING VALUES VARNAME (9).
```

Es sind folgende Regeln zu beachten:

1. Damit eine Syntax-Datei erstellt wird, muß im Abschnitt [SET] die Zeile **SPSS=1** vorhanden sein. Die Datei wird automatisch immer dann erzeugt, wenn WEingabe gestartet, die Initialisierungsdatei geöffnet und das Programm anschließend wieder beendet wird, ohne daß Daten eingegeben werden.
2. Als Dateiname wird der Name (ohne Dateierweiterung) der bei der Datenspeicherung erzeugten Exportdatei verwendet. Heißt z.B. die Exportdatei DATEN.DAT, heißt die Syntax-Datei DATEN.**SPS**. Die Datei wird immer im aktuellen Arbeitsverzeichnis angelegt. Arbeitsverzeichnis ist **immer** das Verzeichnis, in dem sich die Initialisierungsdatei befindet.
3. Eine Syntax-Datei wird nur dann erstellt, wenn die Datendatei Variablenamen enthält. Mindestvoraussetzung dafür ist der Eintrag [NAMES] (S. 29) in der Konfigurationsdatei. Über die Bildung von Variablenamen siehe ansonsten die Erläuterungen zum Abschnitt [NAMES].

4. Der Befehl **VARIABLE LABELS** wird für eine Variable nur dann erzeugt, wenn ein entsprechender Eintrag in den Abschnitten [LABEL] (S. 34) oder [LABEL#] (S. 35) vorhanden ist, wobei gilt:
  - a) Einträge für Variablen aus Eingabefeldern, die mit mehreren Variablen belegt sind, müssen immer im Abschnitt [LABEL#] stehen.
  - b) Einträge für Variablen aus Eingabefeldern, die mit nur einer Variable belegt sind, müssen immer im Abschnitt [LABEL] stehen.Ausführlichere Informationen über die Abschnitte [LABEL] und [LABEL#] erhalten Sie an anderer Stelle (S. 34, S. 35).
5. Der Befehl **MISSING VALUES** für eine Variable wird nur dann erzeugt, wenn ein diese Variable betreffender Eintrag in den Abschnitten [MISSING] (S. 39) oder [MISSING#] (S. 41) vorhanden ist, wobei gilt:

Im [MISSING#]-Abschnitt aufgeführte Missing-Values haben Vorrang vor solchen im [MISSING]-Abschnitt.

Eine genaue Beschreibung der Verwendungsregeln für die Abschnitte [MISSING] und [MISSING#] findet sich an den oben bezeichneten Stellen.
6. Bei Stringvariablen entfällt das Kommando **FORMATS** (die String-Länge wird bereits beim Einlesen der Datendatei bestimmt).
7. Bei Variablen aus einfach-numerischen Eingabefeldern, die in der Initialisierungsdatei durch die Anweisung =@, definiert sind, d.h. Variablen, zu deren Kodierung auch Zahlen zulässig sind, die Nachkommastellen enthalten, beinhaltet die Formatangabe 1 Nachkommastelle, sofern der im Abschnitt [SUB] (siehe S. 26) eingetragene Wert kleiner 3 ist, und **2 Nachkommastellen** in allen anderen Fällen. Für eine Variable, deren Stellenzahl im Abschnitt [SUB] mit 4 angegeben ist, so daß also maximal 3 Dezimalstellen eingegeben werden können, wird beispielsweise folgendes Kommando erzeugt:  
FORMATS VARIABLE (F3.2).

---

## SAV

Syntax:

**SAV=<0 oder 1>**

**Standard:** 0

**Zweck:** Legt fest, ob zusätzlich zur Exportdatei \*.DAT ein SPSS-Datenfile (\*.SAV) erstellt werden soll. Falls dies gewünscht wird, muss der Abschnitt [SET] folgenden Eintrag aufweisen:

SAV=1

Ist gleichzeitig der Eintrag SPS=1 vorhanden, so bleibt dieser wirkungslos, d.h. WEingabe erstellt keine Syntaxdatei \*.SPS (siehe oben).

---

## CASELABEL

Syntax:

**CASELABEL=<0> oder <1> oder <Dateinamen ohne Pfadangabe>**

**Standard:** 0

**Zweck:** Legt fest, ob und wie die Einträge im Abschnitt [CASELABEL] ausgewertet werden (0 = nicht auswerten).

## LAUNCH<sup>6</sup>

Syntax:

**LAUNCH=<0 oder 1>**

**Standard:** 0

**Zweck:** In Eingabefeldern vom Typ STRING können auch externe Programme gestartet werden, wenn das Feld den Programm-Namen zusammen mit einer Pfadangabe enthält, sofern letztere erforderlich ist. Durch Drücken der **Taste F5** wird das Programm gestartet und ausgeführt. Voraussetzung dafür ist aber, daß im Abschnitt [SET] der Eintrag LAUNCH=1 vorhanden ist.

Man kann den Programm-Namen manuell eingeben, oder was der einfachere Weg ist, die AUTOFILL-Funktion nutzen (siehe Seite 47), um mit der Taste F11 den Programmnamen in das Eingabefeld zu schreiben. Dazu ein Beispiel:

[AUTOFILL1]

1=c:\programme\winamp\winamp.exe "C:\Audio\Titel 1.mp3"

2=c:\programme\winamp\winamp.exe "C:\Audio\Titel 2.mp3"

3=c:\programme\winamp\winamp.exe "C:\Audio\Titel 3.mp3"

Hier sollen Audio-Files, die als MP3-Dateien auf der Festplatte liegen, mit dem Programm Winamp angehört und dabei beurteilt werden. Das erste Eingabefeld dient dazu, Winamp zu starten. Die dafür erforderliche Kommandozeile steht im Abschnitt [AUTOFILL1] und wird mit F11 in das Eingabefeld übertragen. Im ersten Datensatz soll "Titel 1" gespielt und beurteilt werden, im zweiten Titel 2 usw.

Es gelten folgende Regeln:

1. Werden in der Kommandozeile dem aufzurufenden Programm Argumente (Parameter) übergeben, wie im obigen Beispiel der Name und Pfad der abzuspielenden Musikdatei, muss dieser Teil der Kommandozeile insgesamt in **doppelte Anführungszeichen** eingeschlossen sein.

---

<sup>6</sup> Nur Premium-Version

2. Statt wie oben den Pfadnamen des Programms in jeder Zeile zu wiederholen, kann man einen Vorgabewert festlegen, indem Sie folgender Abschnitt eingefügt wird:

**[EXEDIR1]**

0=c:\programme\winamp

mit der Implikation:

**[AUTOFILL1]**

1=winamp.exe "C:\Musik\Titel 1.mp3"

2=winamp.exe "C:\Musik\Titel 2.mp3"

3=winamp.exe "C:\Musik\Titel 3.mp3"

3. Dasselbe gilt, wenn es mehrere Eingabefelder gibt, in denen externe Programme aufgerufen werden sollen, hier kann folgendermaßen vorgegangen werden :

**[EXEDIR]**

0=c:\programme\winamp

oder

**[EXEDIR]**

1,5=c:\programme\winamp

2,7,13=c:\programme\notepad

Man beachte, daß sich hier die Einträge nicht auf Datensätze, sondern auf Eingabefelder innerhalb eines Datensatzes beziehen

# Index

---

## **D**

Datenexport 14, 31, 53, 54  
Dezimaltrennzeichen 22, 23, 27, 54, 57

---

## **E**

einfach-numerisches Feld 22, 27, 34, 35, 36  
einstellig-numerische Variable 18, 19, 22, 24, 25, 27, 42  
Export, siehe Datenexport 8, 14, 21

---

## **F**

Feldbreite 14, 19, 20

---

## **G**

ganzzahlige Werte 16, 18, 34, 40, 42  
Gleichheitszeichen 16, 18, 19, 23, 24, 39, 40, 46, 47  
Großschreibung 24, 28, 34, 37, 40, 42, 43

---

## **H**

Hilfefenster 9, 43, 44, 45, 46  
Hilfetext 43, 44, 45  
Hotkey 55

---

## **K**

Komma 23, 37, 53, 54, 57, 58

---

## **L**

Labels 8, 9, 21, 35, 36, 55, 56  
Leerzeichen 14, 15, 18, 19, 23, 24, 34, 35, 36, 37, 38, 40, 47, 53, 64

---

## **M**

mehrstellig-numerische Variable 24, 27

Meldungsfenster 21, 39, 46  
Minuszeichen 22, 23, 55  
Missing-Values 8, 9, 11, 15, 18, 20, 21, 24, 37, 38, 40, 41, 42, 55, 56, 57  
mehrfach-numerisches Feld 22, 23, 24, 27, 30, 32, 35, 36, 38, 43, 44

---

## **N**

Nachkommastellen 23, 54, 57  
numerisches Feld 15, 18, 19, 20, 22, 23, 24, 25, 27, 30, 34, 35, 36, 38, 43, 56

---

## **R**

realzählige Werte 23

---

## **S**

Speichern 15, 16  
SPSS 8, 9, 11, 14, 15, 20, 21, 30, 31, 35, 36, 40, 42, 53, 54, 56  
**SPSS-Format** 57, 58  
Stringvariable 11, 15, 18, 19, 22, 23, 24, 25, 27, 28, 35, 36, 37, 40, 57  
Syntax-Datei (\*.SPS) 9, 15, 56, 57, 58, 64

---

## **T**

THRU, Operator 19, 20, 24, 28, 34, 37, 40, 42, 43  
Trennzeichen 14, 53

---

## **V**

variable labels 35  
Variablenamen 9, 14, 15, 20, 21, 30, 32, 53, 56  
Variablentyp 9, 18, 19, 22, 24, 25  
Vorzeichen 23, 27, 55

---

## **Z**

Zeilenendezeichen 11, 14, 16, 22, 34, 43, 45, 46

## Anhang

### Fragebogen-Beispiel

#### FRAGE 1

(1) Bist Du ein Mädchen oder ein Junge ?  ein Mädchen  ein Junge

#### FRAGE 2

(2) In welchem Jahr bis Du geboren ? 19 \_\_\_\_\_

#### FRAGE 3

Wieviele Geschwister hast Du ? \_\_\_\_\_ Schwestern (3) und \_\_\_\_\_ Brüder (4)

#### FRAGE 4

Welchen Schulabschluß haben Deine Eltern ? (Kreuze bitte jeweils den höchsten Abschluß an!)

##### (5) MUTTER

- keinen Schulabschluß
- Hauptschulabschluß
- Realschulabschluß
- Abitur/Fachhochschulreife
- Ich weiß es nicht

##### (6) VATER

- keinen Schulabschluß
- Hauptschulabschluß
- Realschulabschluß
- Abitur/Fachhochschulreife
- Ich weiß es nicht

#### FRAGE 5

Welchen Berufe haben Deine Eltern erlernt ?

(7) MUTTER \_\_\_\_\_ (8) VATER \_\_\_\_\_

#### FRAGE 6

(9) Hast Du Dich schon um einen Ausbildungs-oder Arbeitsplatz beworben ?  ja  nein

**FRAGE 7**

Wenn ja: für welche Berufe hast Du dich beworben ?

- 1. \_\_\_\_\_ (10)
- 2. \_\_\_\_\_ (11)
- 3. \_\_\_\_\_ (12)

**FRAGE 8**

**(13)** Auf dieser Seite sind eine Reihe von Eigenschaften aufgeführt. Du sollst Dich mit Hilfe dieser Eigenschaften selbst beschreiben. Gib bitte bei jeder Eigenschaft an, wie oft die jeweilige Eigenschaft auf Dich zutrifft.

	sehr oft	oft	manchmal	selten	nie
1. romantisch	<input type="checkbox"/>				
2. bemüht, verletzte Gefühle zu besänftigen	<input type="checkbox"/>				
3. feinfühlig	<input type="checkbox"/>				
4. unerschrocken	<input type="checkbox"/>				
5. hartnäckig	<input type="checkbox"/>				
6. kraftvoll	<input type="checkbox"/>				
7. empfindsam	<input type="checkbox"/>				
8. sicher	<input type="checkbox"/>				
9. konsequent	<input type="checkbox"/>				
10. herzlich	<input type="checkbox"/>				

**FRAGE 9**

**(14)** Bitte trage in die Kästchen Dein Kennwort ein:

Die beiden ersten Buchstaben des Vornamens Deiner Mutter:

 

Dein eigener Geburtstag (nur der Tag):

 

Die beiden ersten Buchstaben des Vornamens Deines Vaters:

## Die Initialisierungsdatei BEISPIEL.INI

;Die Datei beschreibt eine Eingabemaske für das vorstehende Fragebogen-Beispiel

[ITEMS]

N=14

[CODES]

1=019

2,3,4=@

5,6=123459

7,8,10THRU12=ÄÖÜABCDEFGHIJKLMNOPQRSTUVWXYZ \_.

9=109

13=123459

14=

[CODES14]

1=ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ

2=ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ

3=0123456789X

4=0123456789X

5=ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ

6=ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ

[SUB]

0=1

2,3,4=2

7,8,10THRU12=40

13=10

14=6

[NAMES]

1=SEX

2=GEBOREN

3=GESCHW\_S

4=GESCHW\_B

5=BILD\_MUT

6=BILD\_VAT

7=BERUF\_M

8=BERUF\_V

9=BEWERB

10=BBEWERB1

11=BBEWERB2

12=BBEWERB3

13=SELBST

14=KENNWORT

[NAMES13]

4=SELBST22

[LABEL]

1=Junge oder Mädchen

2=Geburtsjahr

3=Anzahl der Schwestern

4=Anzahl der Brüder

5=Schulabschluß der Mutter

6=Schulabschluß des Vaters

7=erlernter Beruf der Mutter

8=erlernter Beruf des Vaters

9=Bewerbung um Ausbildungsplatz

10=Frage nach Bewerbungsberufen: 1. Nennung

11=Frage nach Bewerbungsberufen: 2. Nennung  
12=Frage nach Bewerbungsberufen: 3. Nennung  
13=Beschreibe Dich selbst  
14=Schülercode

[LABEL13]

1= romantisch  
2= besänftigen  
3= feinfühlig  
4= unerschrocken  
5= hartnäckig  
6= kraftvoll  
7= empfindsam  
8= sicher  
9= konsequent  
10=herzlich

[RANGE]

2=73THRU80,99  
3,4=0THRU12,99

[SET]

TITEL=Beispiel einer Eingabemaske  
SPS=1

[SKIP]

9=13(0)  
10=13(.)  
11=13(.)  
12=13(.)

[SKIPMESSAGE]

9,10TRU12=Folgende Frage(n) überspringen ?// Wenn ja,weiter mit:// Beschreibe Dich selbst...

[MISSING]

2,3,4=99  
1,5,6,13,9=9  
7,8,10THRU12=.  
14=XXXXXX

[MESSAGE]

14=Weiter mit 6-stelligem Code.//Bitte Sorgfalt bei der Eingabe !

[HELP]

1=1=Junge//2=Mädchen//fehlt=9  
2=Zulässige Angaben: 73-80//keine Angabe = 99  
3,4=Zulässige Angaben: 0-12//Keine Angabe = 99  
5,6=Codierung://einzugeben ist die Nummer des//angekreuzten Kästchens.//Nummer 1 ist bei senkrechter Anordnung//das oberste Kästchen, bei waagerechter//Anordnung das linkeste Kästchen.//Ist kein Kästchen angekreuzt, wird eine 9 eingegeben  
7,8,10THRU12=1. Keine Leerzeichen, stattdessen Unterstrich verwenden//2. Bei fehlender Angabe einen Punkt (.) eingeben.//3. Schreibfehler nicht übernehmen.  
9=Codierung://1=ja//0=nein//9=fehlt  
13=Codierung://Die Ausprägungen der Skala://sehr oft-oft-manchmal-selten-nie//werden immer von links nach rechts,//mit 1 beginnend,durchnumeriert.//9 = Ankreuzung fehlt  
14=Zusammensetzung des Kennwortes://2 Buchstaben - 2 Ziffern - 2 Buchstaben//Fehlende Zeichen durch X ersetzen

