



TUSTEP

Erste Schritte



Michael Trauth

März 2018





Inhalt

1. Installation von Tustep	3
2. Organisatorisches	4
a) Einrichtung von Tustep-Datenordnern	4
b) Einrichtung von Tustep-Sitzungen	5
3. Kommandos und Kommandoebene	7
a) Beenden und Unterbrechen einer TUSTEP-Sitzung	11
b) Dateien und Datei-Kommandos	11
4. Tustep-Editor-Anweisungen	15
a) Vorbemerkung	15
b) Einfache Anweisungen	16
c) Tastenfunktionen	18
d) Erweiterte Anweisungen	19



Erste Schritte mit Tustep

1. Installation von Tustep

Zu diesem Punkt muß nicht viel gesagt werden, denn abgesehen davon, daß das Thema kein schwieriges ist, gibt es eine sehr gute und reich bebilderte Anleitung dazu unter den Handbüchern, die TUSTEP von Haus aus mitbringt. Der erste TUSTEP-Befehl, mit dem Sie sich hier konfrontiert sehen, lautet deshalb

– #*zebe, config

Er sorgt dafür, daß besagte Anleitung geöffnet wird – wenn auf Ihrem System der Adobe Acrobat oder ein anderer pdf-Reader installiert ist.

Einziges kleines Problem: Diese Anleitung steht Ihnen erst zur Verfügung, wenn Sie TUSTEP bereits installiert *haben*. Macht nichts: Sie finden die Datei CONFIG.PDF, um die es hier geht, deshalb sowohl (a) im WWW unter dem URL <http://www.tustep.uni-tuebingen.de/inst.html> wie auch (b) unter den Materialien, die Ihnen für diesen Kurs zur Verfügung gestellt werden.

Für Windows-Anwender sei hier gleichwohl noch in aller Kürze resümiert, daß die exe-Datei, die das Installationspaket enthält, lediglich mit einem Doppelklick im Explorer aufgerufen werden muß. Danach werden Sie begrüßt und über die TUSTEP-Version, die installiert werden soll, informiert. Nach einem Klick auf →CONTINUE wird Ihnen angezeigt, in welchem Programmordner TUSTEP installiert werden soll: Diese Option kann geändert werden, aber in aller Regel spricht nichts dagegen, den vorgeschlagenen Ordner mit ›OK‹ zu akzeptieren. Ein weiterer Klick auf →START läßt die Installation beginnen, die dann schon nach wenigen Sekunden beendet ist. Zum Schluß können Sie sich mit Klick auf →INFO weitere Informationen geben lassen oder mit Klick auf →FINISH die Installation beenden. – Hin und wieder zeigt sich Windows besorgt, daß die Installation »eventuell nicht richtig installiert« worden sei. Das muß Sie aber nicht ebenfalls in Unruhe versetzen: In solch einem Fall können Sie ohne weiteres bestätigen, daß die Installation fehlerfrei abgelaufen sei.

2. Organisatorisches

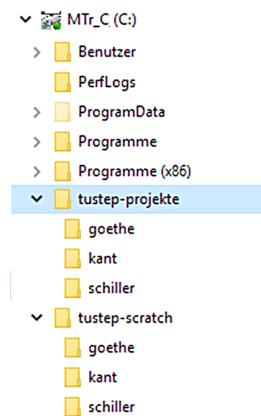
a) Einrichtung von Tustep-Datenordnern

TUSTEP erzieht Sie nachdrücklich dazu, alle Daten Ihrer verschiedenen Projekte an exakt definierten Stellen unterzubringen. Im Gegensatz dazu sind viele Windows-Anwender noch immer gewohnt, den ganzen riesigen Speicherplatz, der ihnen auf modernen Festplatten zur Verfügung steht, zur Ablage zu nutzen – nicht selten mit den fatalen Konsequenzen des sog. Eichhörnchen-Syndroms: Man hat die Nüsse irgendwo versteckt und weiß nicht mehr, wo und unter welchem Namen ... oder hat womöglich gar vergessen, daß man sie hat. Die damit einhergehende Konfusion hat sich etwas gebessert, seit Windows wenigstens für Dokumente, Bilder, Musikdateien etc. Default-Ablageordner im Profil des jeweiligen Benutzers zur Verfügung stellt, aber noch immer sind der Planlosigkeit des Benutzers Tür und Tor geöffnet.

Ganz anders als Windows geht TUSTEP jedenfalls davon aus, daß *alle* Ihre Daten, die zu einem *bestimmten* thematischen Projekt gehören, in genau *einem* Verzeichnis zu finden sind. Meine persönliche Empfehlung lautet, daß Sie sich für Ihre Tustep-Projekte (die im Laufe der Zeit sehr zahlreich werden können) *ein* übergeordnetes Verzeichnis (z. B. mit dem frei wählbaren Namen ›tustep‹ oder ›tustep-projekte‹) anlegen und *unterhalb* dieses Mutterverzeichnisses für jedes Projekt einen separaten Ordner anlegen. Ein Beispiel dazu folgt gleich.

Zunächst jedoch gebe ich schon hier eine weitere Empfehlung, die auf Einsteiger möglicherweise etwas ›sophisticated‹ wirkt, aber einen sehr nützlichen Hintergrund hat: Die TUSTEP-Prozeduren, mit denen Sie später arbeiten und die Sie vielleicht auch selbst erstellen werden, funktionieren in der Regel sequenziell, d. h. daß die Ausgangsdateien stets unverändert bleiben und Zwischenstadien der Verarbeitung in sog. ›temporären Dateien‹ (auch ›Scratchdateien‹ genannt) abgelegt werden; das ist – unter anderem – sehr hilfreich, wenn man einem Fehler in der Logik der Prozedur auf die Schliche kommen will. Diese Dateien dienen gewissermaßen als Schmier- oder Konzeptpapier, das ebenfalls nur kurzzeitig gebraucht wird, und haben deshalb, wie leicht ersichtlich, einen geringeren Wert als die ›permanenten‹ resp. ›katalogisierten‹ Daten im eigentlichen Projektverzeichnis. Zumal die Prozedur sie immer wieder aufs neue generiert, müssen sie auch nicht unbedingt per Backup gesichert werden. TUSTEP bietet Ihnen deshalb die Möglichkeit, die Scratchdateien an anderer Stelle unterzubringen als die eigentlichen Projektdaten, damit Wichtiges von weniger Wichtigem getrennt bleiben kann. Das funktioniert genauso wie oben beschrieben, nur daß für die Scratchdateien der jeweiligen Projekte ein weiteres, neues Mutterverzeichnis angelegt wird.

Das folgende Beispiel geht davon aus, daß Sie drei wissenschaftliche Projekte zu den Themen ›Goethe‹, ›Kant‹ und ›Schiller‹ *in petto* haben und die Daten dazu auf Ihrem Festplattenvolume ›C:\‹ unterbringen wollen. Wenn Sie meinem Vorschlag folgen möchten, legen Sie also die beiden Mutterverzeichnisse C:\tustep-projekte und C:\tustep-scratch an und richten in beiden(!) die Unterverzeichnisse ›goethe‹, ›kant‹ und ›schiller‹ ein. Die Anzeige in Ihrem Explorer-Fenster sollte demnach in etwa so aussehen wie in der Abbildung rechts: Ihre eigentlichen Projektdaten, die ›katalogisierten‹ bzw. ›permanenten‹ Dateien, gehören dann in die Verzeichnisse unterhalb von C:\tustep-projekte, und um die Verwaltung der temporären Dateien soll sich TUSTEP in den Verzeichnissen mit denselben Namen unterhalb von C:\tustep-scratch kümmern.



Bleibt nur noch die Aufgabe, TUSTEP von dieser Absicht in Kenntnis zu setzen – und das geschieht durch das ›Einrichten einer Sitzung‹:

b) Einrichten von Tustep-Sitzungen

Was sind ›Tustep-Sitzungen‹ überhaupt? Im Grunde handelt es sich dabei um die vom Benutzer *ein einziges Mal* vorzunehmende Definition der Absicht, mit TUSTEP an den Daten eines *bestimmten* Projektes arbeiten zu wollen; diese Definition wird in Gestalt einer sog. ›TSD-Datei‹ abgelegt (TSD = Tustep-Start-Datei und mit einem Tustep-Icon sichtbar gemacht. Von da an können solchermaßen definierte Sitzungen durch Doppelklick auf das Icon gestartet werden.

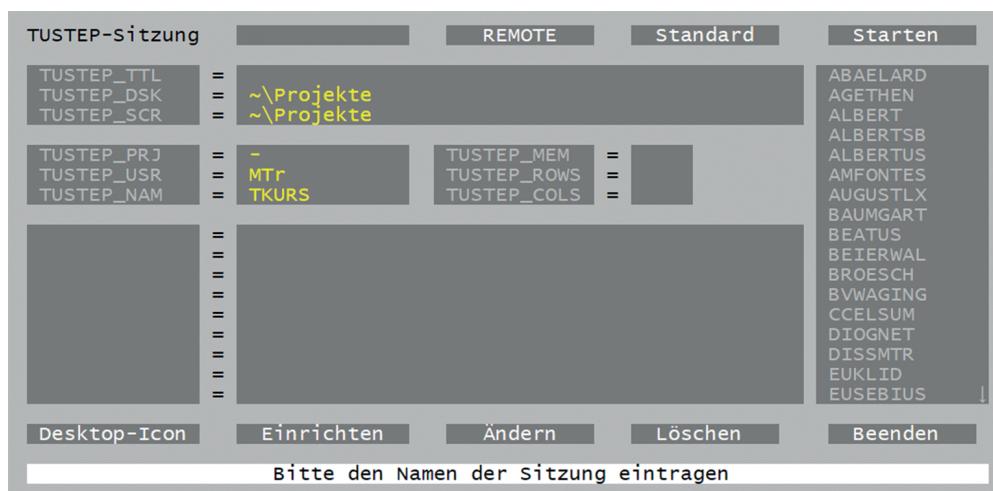
An einer solchen Sitzungsdefinition ist nichts geheimnisvolles. Wenn Sie möchten, können Sie die TSD-Dateien mit einem beliebigen Textprogramm öffnen und sich davon überzeugen, daß nur einige wenige Textinformationen darin stehen; z. B. sieht eine mögliche Definition für die Sitzung unseres Kurses folgendermaßen aus:

```
TUSTEP-Session: KURS18TR
TUSTEP_TTL=[T] Tustep Spring School 2018
TUSTEP_DSK=C:\tustep
TUSTEP_SCR=C:\tustep-scratch
TUSTEP_PRJ=kurs18tr
TUSTEP_MEM=00100000
TUSTEP_USR=trauth
TUSTEP_NAM=KURS18TR
TUSTEP_COLS=
TUSTEP_ROWS=
```

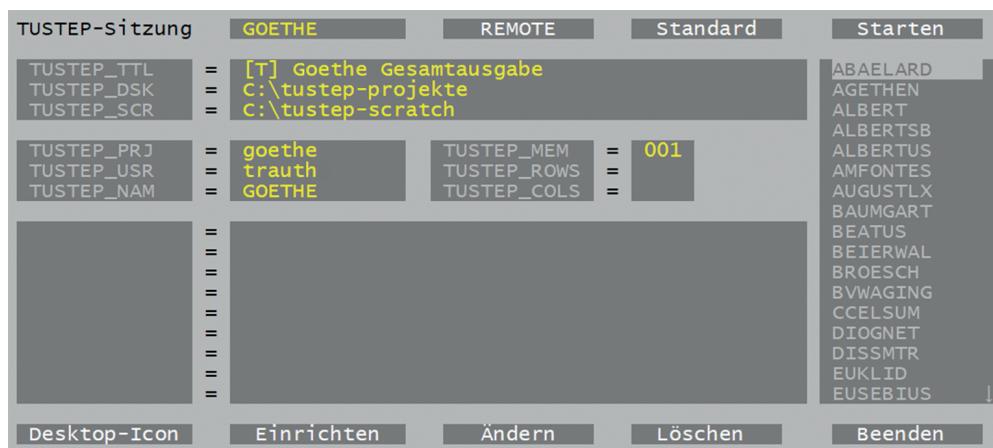
Was diese dünnen Zeilen, eine Handvoll Variablendefinitionen, besagen, können Sie sich unschwer selbst zusammenreimen: In der ersten steht der Name der Sitzung ›kurs18tr‹, in der zweiten ein beliebiger Titel (der später in der Kopfzeile

des Tustep-Fensters angezeigt werden soll), in der dritten und vierten stehen die Namen der oben (S. 4f) beschriebenen Mutterverzeichnisse, in der fünften der Name des Projektverzeichnisses und in der sechsten die Nummer der Sitzung.

Nicht schwierig also – und es wäre ein leichtes, eine solche Sitzungsdefinition ggf. selbst von Hand zu erstellen. Aber es geht einfacher: Nachdem Sie TUSTEP installiert haben, finden Sie das TUSTEP-Programm-Icon im Windows-Startmenü. Dessen Aufruf startet das Programm #*DESI (DESI = DEfiniere Sitzung), das Sie beim Definieren einer solchen TUSTEP-Sitzung unterstützt. Dieses #*DESI-Fenster hat in etwa folgendes Aussehen:



Bitte nicht erschrecken – die Chose ist sehr viel einfacher, als es auf den ersten Blick erscheinen mag. Um die erste Sitzung für das oben als Beispiel herangezogene Projekt ›goethe‹ zu definieren, müssen Sie nur die folgenden Angaben in die Maske eintragen. Das sind im Prinzip dieselben Angaben, die ich Ihnen bereits auf der vorangehenden Seite vorgestellt habe.



Nachdem Sie die Felder der Eingabemaske ausgefüllt haben, klicken Sie auf → Einrichten – und schon ist Ihre erste Sitzungsdefinition erledigt. Wenn Sie möchten, können Sie außerdem noch den Button → Desktop-Icon anklicken, dann wird Ihnen ein Link zur Sitzungsdefinition als Tustep-Icon auf den Desktop gelegt. Ein Doppelklick auf dieses Icon startet dann die betr. Sitzung.

Die folgenden Ausführungen gehen davon aus, daß Sie Ihre erste TUSTEP-Sitzung erfolgreich eingerichtet und gestartet haben. Ihr TUSTEP-Fenster sollte jetzt also etwa das folgende Aussehen haben:



Das spartanische Outfit, mit dem Sie sich konfrontiert sehen, ist darauf zurückzuführen, daß TUSTEP auf sehr unterschiedlichen Betriebssystemen (Windows, Macintosh, Linux) im Biotop eines Terminal-Fensters bzw. der DOS-Box läuft und überall, auch in sog. Remote-Sitzungen, mit stets gleicher Funktionalität benutzbare sein muß. Diese Prämisse schränkt die Möglichkeiten zur Gestaltung grafischer Benutzeroberflächen weitgehend ein. Lassen Sie sich aber durch das nüchterne Auftreten von TUSTEP nicht täuschen: Auf die außerordentliche Mächtigkeit des Programmsystems hat das keinen Einfluß.

► Kommandos und Kommando-Ebene

Wie Sie sehen, erwartet TUSTEP von Ihnen die Eingabe von ›Kommandos‹. (Die etwas barsch wirkende Aufforderung geht auf die historisch weit zurückreichenden Anfänge von TUSTEP in der Welt der damaligen ›Großrechner‹ zurück.) Im folgenden werden Ihnen deshalb einige ausgewählte elementare Kommandos vorgestellt, die Sie für Ihre weitere Arbeit mit TUSTEP brauchen können.

Kommandos werden hinter jenes "> Gib Kommando" geschrieben und mit der RETURN(\leftarrow) oder der ENTER-Taste zur Ausführung ›weggeschickt‹.

Alle Kommandos werden mit ihrem Namen und dem vorangestellten Befehlszeichen ›#‹ aufgerufen (z. B. #ZEIT); die Kommandonamen dürfen dabei so weit abgekürzt werden, als sie eindeutig bleiben (z. B. #ZE).

Die meisten Kommandos haben darüber hinaus Erweiterungen, die ›Spezifikationen‹ genannt werden. So benötigt z. B. das Kommando #KOPIERE natürgemäß ergänzende Angaben, *was* kopiert werden soll und *wohin* kopiert werden soll; das sind die Spezifikationen QUELLE und ZIEL. Die kanonische Form des Kommandos ist deshalb

- #kopiere,quelle=q,ziel=z

Der Befehl besagt, daß die Datei Q in die Datei z kopiert werden soll.

Auch hier gilt, daß das Kommando so weit abgekürzt werden darf, als es dabei eindeutig bleibt. Für alle Spezifikationen ist eine Default-Reihenfolge vorgegeben, und wenn diese beim Schreiben des Befehls eingehalten wird, dürfen zur Erleichterung und Beschleunigung der Notation auch die Spezifikationsnamen weggelassen werden. Das Kommando kann also bis zu der Form

- #ko,q,z

abgekürzt werden.

Das Kommando #KOPIERE hat freilich – wie andere Kommandos auch – noch weitere Spezifikationen, etwa als nächstes MODUS=... (was es damit auf sich hat, erfahren Sie später) und LOESCHEN=... Mit dieser zweiten Angabe kann gesteuert werden, ob ein eventuell in der Zielfile z vorhandener Inhalt beim Kopieren überschrieben (=gelöscht) werden soll oder nicht.

Beachten Sie bitte hier schon, daß striktes Bedachtsein auf Datensicherheit ein Wesenszug von TUSTEP ist: Daten in Zielfilen werden *nur* dann überschrieben, wenn der Benutzer es ausdrücklich verlangt. Konkret bedeutet das, daß der o. a. Befehl #ko,q,z den Inhalt der Quelldatei Q *ans Ende* des in der Zielfile z enthaltenen Inhalts kopiert. Dagegen kopiert das ausführlich notierte Kommando

- #kopiere,quelle=q,ziel=z,modus=-std-,loeschen=+

den Inhalt der Datei Q vollständig *an die Stelle* dessen, was zuvor in z enthalten war, *löscht* also diesen früheren Inhalt!

Weil die Spezifikation MODUS=-STD- voreingestellt ist, kann sie auch weggelassen werden. Die folgende Kurznotation ist demzufolge gleichwertig mit der ausführlichen Notation:

- #ko,q,z,,+

Beachten Sie dabei, daß die *Position* der Modus-Angabe (zwischen dem dritten und dem vierten Komma) freigehalten werden muß, damit das Plus-Zeichen auf der letzten Position korrekt als LOESCHEN=+ interpretiert wird. Im folgenden Screenshot sehen Sie, daß das Resultat *beider* Befehle exakt *dasselbe* ist:

```
Gib Kommando >#kopiere,quelle=q,ziel=z,modus=-std-,loeschen=+
Start KOPIERE auf: YGGDRASIL am: 08.04.17 um: 18:37:49
**** Eingabe: 153 Sätze von Scratch-Datei WEIGEL10*Q
Satzlänge: 0/98/384 Satznummern: 1.1 - 1.153
**** Ausgabe: 153 Sätze auf Scratch-Datei WEIGEL10*Z
Satzlänge: 0/98/384 Satznummern: 1.1 - 1.153
Ende KOPIERE auf: YGGDRASIL am: 08.04.17 um: 18:37:49
=====
Gib Kommando >#ko,q,z,,+
Start KOPIERE auf: YGGDRASIL am: 08.04.17 um: 18:38:05
**** Eingabe: 153 Sätze von Scratch-Datei WEIGEL10*Q
Satzlänge: 0/98/384 Satznummern: 1.1 - 1.153
**** Ausgabe: 153 Sätze auf Scratch-Datei WEIGEL10*Z
Satzlänge: 0/98/384 Satznummern: 1.1 - 1.153
Ende KOPIERE auf: YGGDRASIL am: 08.04.17 um: 18:38:05
=====
Gib Kommando >
```

Ich könnte Sie jetzt bitten, die beiden Kommandos einmal probeweise genau wie im Bild einzugeben. Aber wenn Sie es täten, dürften Sie wohl kaum Erfolg haben. Der Grund dafür ist offensichtlich: Die beiden Dateien *Q* und *Z* sind in Ihrem Projekt ganz einfach (noch) nicht vorhanden. – Das ist, wie ich meine, die Gelegenheit für die Vorwegnahme einer wichtigen (und hilfreichen) Erfahrung, die Ihnen als Einsteiger zweifellos noch bevorsteht: die berüchtigte **rote TUSTEP-Fehlermeldung**, die schon viele TUSTEP-Novizen in Panik versetzt hat. Aber für Aufregung besteht kein Anlaß: Es ist im Gegenteil völlig normal, daß Sie sich bei einer Eingabe auch einmal vertippen oder einen anderen Fehler begehen. Und bevor dieser Fehler womöglich unheilvolle Konsequenzen hat, bricht TUSTEP die Ausführung ab und warnt Sie in dieser auffälligen Weise:

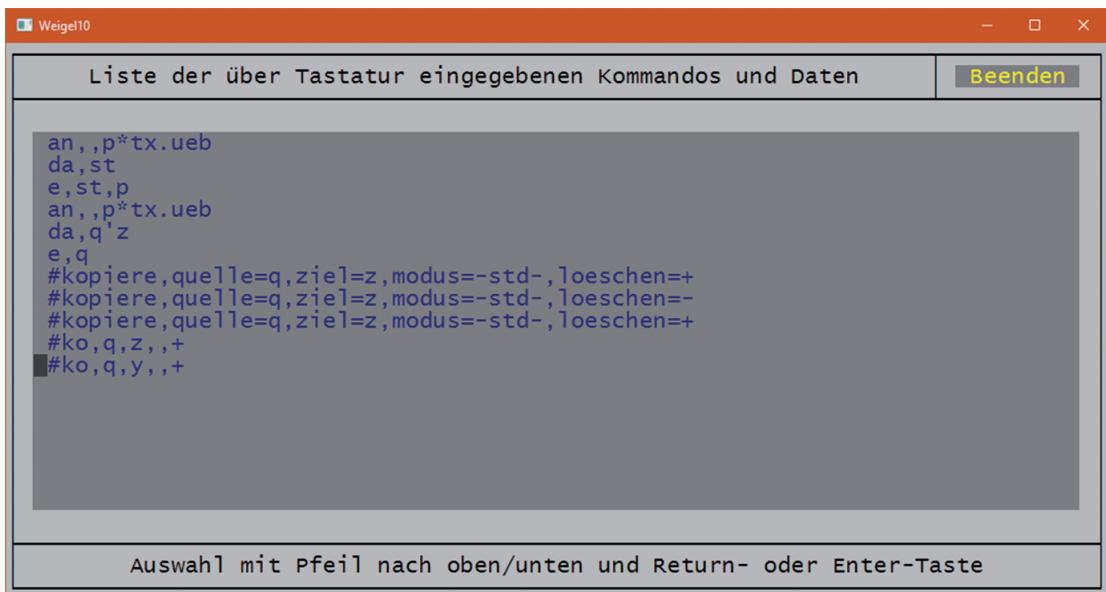
```
Gib Kommando >#ko,q,z,,+
Start KOPIERE auf: YGGDRASIL am: 08.04.17 um: 18:38:05
**** Eingabe: 153 Sätze von Scratch-Datei WEIGEL10*Q
Satzlänge: 0/98/384 Satznummern: 1.1 - 1.153
**** Ausgabe: 153 Sätze auf Scratch-Datei WEIGEL10*Z
Satzlänge: 0/98/384 Satznummern: 1.1 - 1.153
Ende KOPIERE auf: YGGDRASIL am: 08.04.17 um: 18:38:05
=====
Gib Kommando >#ko,q,y,,+
Start KOPIERE auf: YGGDRASIL am: 09.04.17 um: 01:32:22
##### ZIEL-Datei WEIGEL10*Y wurde nicht gefunden #####
#####
##### PROGRAMABBRUCH: Kommando-, Datei- oder Parameterfehler #####
#####
Ende KOPIERE auf: YGGDRASIL am: 09.04.17 um: 01:32:22
=====
Gib Kommando >
```

Sehen Sie denn gleich, was den Fehler verursacht hat? Wenn nein, brauchen Sie nur den Text der Fehlermeldung zur Kenntnis zu nehmen:

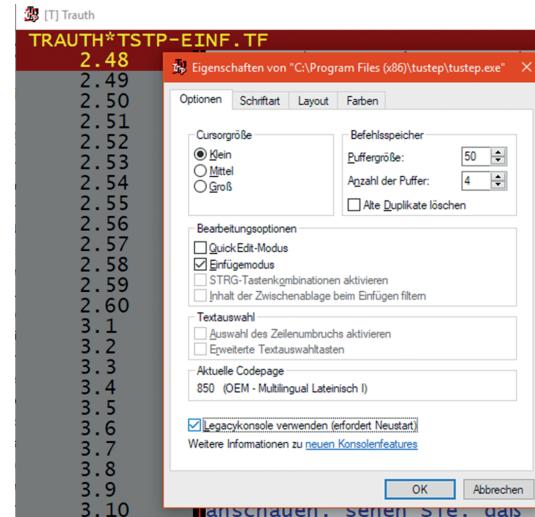
ZIEL-Datei <PROJEKT>*Y wurde nicht gefunden.

Wenn Sie sich dann das protokolierte Kommando `#ko,q,y,,+` anschauen, sehen Sie, daß dort statt der Zielfile *z* die (falsche) Datei *y* angegeben wurde.

Das fehlerhafte Kommando muß also korrigiert werden. Statt es mit der richtigen Angabe der Zielfile noch einmal vollständig einzutippen (was bei komplexen Befehlen durchaus mühselig sein kann) können Sie es – weitaus einfacher – durch Druck auf die Cursor-Taste ↑ in die Kommandozeile zurückholen und dort allein die falsche Angabe verbessern. – *Hinweis:* Die Cursor-Tasten ↑ und ↓ navigieren rückwärts und vorwärts durch die ›History‹ der bisher gegebenen Kommandos. Ganz besonders elegant aber zeigt Ihnen die Taste F5 eine Übersicht der History, aus der Sie frühere Kommandos auswählen können:



Darin wählen Sie eine Zeile der Liste entweder mit RETURN oder Mausklick aus: Dann wird dieser Befehl sofort erneut ausgeführt. Oder aber Sie drücken die ENTER-Taste: Dann wird der Befehl, z. B. zur Korrektur, nur in die Kommandozeile zurückgebracht. – *Hinweis:* Unter Windows 10 kann es vorkommen, daß im Konsolenfenster, in dem TUSTEP läuft, der Cursor nicht sichtbar ist. Stellen Sie in diesem Fall einfach in den Eigenschaften Ihres TUSTEP-Fensters die alternative ›Legacy-Konsole‹ ein (siehe Abbildung rechts).



► Beenden und Unterbrechen einer TUSTEP-Sitzung

Als nächstes sollten Sie erfahren, wie Sie Ihre TUSTEP-Sitzung wieder beenden können. Dafür haben Sie gleich mehrere Optionen, doch ist es *vor* deren Verwendung wichtig zu wissen, daß Sie mit diesem ›Beenden‹ zwei grundsätzlich *verschiedene* Vorgänge auslösen können:

- Im einen Fall wird die Sitzung lediglich *unterbrochen*, was bedeutet, daß die Anweisungen und Kommandos, die Sie im Verlauf der Sitzung gegeben haben (die sog. ›History‹), ebenso wie die Einstellungen und Definitionen, die Sie vorgenommen haben, einschließlich der temporären Dateien, mit denen Sie evtl. gearbeitet haben, *erhalten* bleiben. Anders formuliert: Wenn Sie also – beispielsweise – die Sitzung später erneut starten und den Editor aufrufen, wird dieser in dieselbe Datei zurückkehren, die Sie zuletzt bearbeitet haben; sie werden auch alle anderen Einstellungen so vorfinden, wie Sie sie in Ihrer letzten Arbeitssitzung verlassen haben.
- Im anderen Fall wird die Sitzung tatsächlich im Sinne des Wortes ›beendet‹, d. h. daß das ›Gedächtnis‹ von TUSTEP für diese Sitzung, die ›History‹ ihrer gemerkten Kommandos und Anweisungen, auch ihrer Einstellungen und Definitionen, einschließlich aller temporären Dateien gelöscht werden.
- #beende, modus=- Das ist das kanonische Kommando zur *Unterbrechung alternativ: #be, -* der Sitzung. Aber Sie können sich den Notationsaufwand auch sparen, indem Sie die einfachere Alternative F4 verwenden. Unter Windows steht Ihnen außerdem als dritte Möglichkeit die ESC-Taste zur Verfügung.
- #beende, modus=+ Das ist das kanonische Kommando für das eigentliche *alternativ: #be, +* ›Beenden‹ der Sitzung (s. o.).

► Dateien und Datei-Kommandos

TUSTEP kennt zwei eigene Dateiformate, SEQ und RAN, kann aber auch mit Fremdformaten aus anderen Quellen umgehen. Wie man Fremddaten nach TUSTEP importiert und ggf. auch exportiert, erfahren Sie in einem eigenen Kursmodul. Auch auf die Unterschiede zwischen SEQ und RAN, die (für den Benutzer) überwiegend im Performance-Verhalten bestehen, soll hier nicht weiter eingegangen werden.¹ Die folgenden Ausführungen behandeln deshalb ausschließlich Dateien vom Typ SEQ, doch gelten die Hinweise auch für den Typ RAN.

1 Weiterführende Erläuterungen dazu finden Sie im aktuellen Handbuch 2018 auf den Seiten 31 ff.

Von der inhaltlichen Struktur her unterscheidet TUSTEP drei Dateiarten: Text-, Programm- und Segmentdateien. Sie werden alle in gleicher Weise eingerichtet, unterscheiden sich also nur in der Art, in der sie verwendet werden, und in der Art der Daten, die sie enthalten.

Dateien, die mit TUSTEP bearbeitet werden sollen, müssen außerdem eine strenge *Dateinamen-Konvention* einhalten: Die Namen dürfen maximal zwölf Zeichen lang sein und zusätzlich eine maximal vier Zeichen lange Extension haben. Als Zeichen dürfen nur Buchstaben, Ziffern, der Bindestrich und der Underscore ›_‹ verwendet werden, wobei das erste Zeichen stets ein Buchstabe sein muß. Die Gründe für diese Rigorosität sind überwiegend historische Natur und bildeten vor Jahren den größten gemeinsamen Nenner für Dateinamen auf den unterschiedlichen Plattformen, auf denen TUSTEP zuhause war. Beispiele dafür:

- GOETHE-WB-T1.TF
- CAESAR-BC-L1.TF
- SATZPROG1.P

usw. Die Extensionen sind im Rahmen der Konvention frei wählbar, doch hat es sich als empfehlenswert erwiesen, die Extension .TF für Textdateien und die Extension .P für Programmdateien zu wählen.

Um Dateien einzurichten, benötigen Sie das Kommando #DATEI, bei dem der Name der einzurichtenden Datei und ihr Typ anzugeben sind, also beispielsweise:

- #DATEI, NAME=TEST1.TF, TYP=SEQ-P
alternativ: #da, test1.tf, seq-p

Damit wird eine Datei mit dem Namen TEST1.TF vom Typ SEQ *permanent* (=›-P‹) eingerichtet.

Außer diesen dauerhaft eingerichteten (›permanenten‹) Dateien können Sie auch temporäre (Scratch-)Dateien einrichten, die ähnlich wie Schmierpapier lediglich zur Aufnahme von (unwichtigen) Zwischenergebnissen gedacht sind:

- #DATEI, NAME=TEST2.TF, TYP=SEQ-T
alternativ: #da, test2.tf

Beachten Sie bitte, daß diese temporären Dateien beim Beenden der Sitzung ohne Rückfrage gelöscht werden!

Dateinamen können selbstverständlich später ggf. geändert werden, und zwar mit dem Kommando #AENDERE:

- #AE, TEST2.TF:MAX-17.TF
womit der Dateiname TEST2.TF in MAX-17.TF geändert wird.

Um bereits existierende Dateien in TUSTEP verwenden zu können, müssen sie *angemeldet* werden, und zwar entweder zum Lesen oder zum Schreiben. Das ist für den Einsteiger i.d.R. zunächst einmal ungewohnt, hat aber vor allem in Teams und in Netzwerken seine Berechtigung, wo mehrere User gleichzeitig

lesend auf dieselbe Datei zugreifen müssen. Es stellt außerdem einen wirksamen Schutz vor eigenen Fehlern dar, denn eine Datei, die nur zum Lesen angemeldet wurde, kann weder versehentlich verändert, noch gelöscht werden.

- #ANMELDE, LESEN=TEST1.TF
alternativ: #an, test1.tf

meldet die Datei TEST1.TF zum Lesen an. Die Datei kann gelesen, aber nicht verändert oder gelöscht werden.

- #ANMELDE, SCHREIBEN=TEST1.TF
alternativ: #an, , test1.tf (beachten Sie die *beiden* Kommata!)

meldet die Datei TEST1.TF zum Schreiben an; sie kann nicht nur gelesen, sondern auch verändert werden.

Dieses Anmelden kann in Umgebungen, in denen nur *ein* einziger Benutzer administrativen Zugriff auf *alle* Dateien hat, lästig sein. Es kann deshalb ggf. mit dem einmaligen Befehl

- #AN, , +
der *sämtliche* Dateien des Projektes zum Schreiben anmeldet, umgangen werden. Bitte seien Sie sich aber der Risiken bewußt, die Sie damit eingehen!

Vor dem Anmelden ist es freilich i.d.R. nützlich, sich einen Überblick über die im Projekt vorhandenen Dateien verschaffen zu können. Das besorgt das Kommando #LISTE:

- #LISTE, DATEINAMEN
alternativ: #li, da

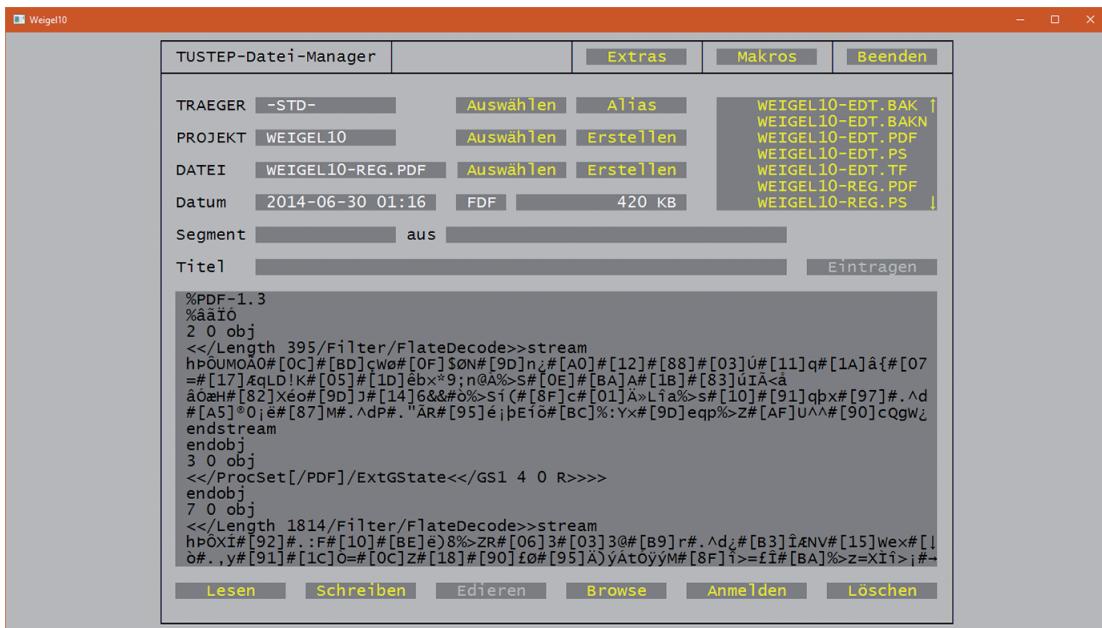
liefert Ihnen eine alphabetische Liste dieser Dateien in drei Kolumnen:

```
Gib Kommando >list dateinamen
=====
Start LISTE auf: YGGDRASIL am: 09.04.17 um: 19:44:36
**** 73 von 86 Dateien mit PROJEKT = WEIGEL10 und TRAEGER = TUSTEP_DSK
ABB01.EPS      ABB108A.EPS      WEIGEL10-EDT.BAK
ABB02.EPS      ABB108A.PNG      WEIGEL10-EDT.BAKN
ABB03.EPS      GRAFIK.P       WEIGEL10-EDT.PDF
ABB04.EPS      KONVERT.P      WEIGEL10-EDT.PS
ABB05.EPS      KONVERT2.P     WEIGEL10-EDT.TF
ABB06.EPS      TEST1.TF       WEIGEL10-REG.PDF
ABB07.EPS      TUSTEP.INI     WEIGEL10-REG.PS
ABB011.EPS     TZSTEP.INI     WEIGEL10-REG.TF
ABB021.EPS     W-EDT.P        WEIGEL10-SON.PS
ABB031.EPS     W-IMPORT.P    WEIGEL10-SON.TF
ABB041.EPS     W-REG.P        WEIGEL10-VOR.PDF
ABB051.EPS     W-REGSATZ.P   WEIGEL10-VOR.TF
ABB052.EPS     W-ROHREG.P    WEIGEL10.GR0
ABB061.EPS     W-SONDERD.P   WEIGEL10.GR1
ABB071.EPS     W-STRUKT.P    WEIGEL10EDTF.PS
ABB082.EPS     W-VOR.P       WEIGEL10R.A
ABB101.EPS     WEIGEL-REG.KRZ WEIGEL10R.FDF
ABB102.EPS     WEIGEL.TRL     WEIGEL10VOR.PDF
ABB103.EPS     WEIGEL5-VOR.TF WEIGEL10VOR.PS
ABB104.EPS     WEIGEL6-SOND.PS WEIGEL10VORF.PS
ABB105.EPS     WEIGEL6-VOR.PS WEIGELX1.FDF
ABB106.EPS     WEIGEL6-VOR.TF WEIGELX2.FDF
ABB107.EPS     WEIGEL10-ED2.PDF WEIGELXH.FDF
ABB108.EPS     WEIGEL10-ED2.PS
ABB108.PNG     WEIGEL10-ED2.TF
=====
Ende LISTE auf: YGGDRASIL am: 09.04.17 um: 19:44:36
=====
```

Gib Kommando >

Mit diesem Kommando können auch noch andere Informationen gelistet werden, aber darüber können Sie sich im Handbuch im Detail informieren.

Im selben Kontext darf auch der mächtige *Tustep-Dateimanager* nicht unerwähnt bleiben, den Sie auf der Kommandoebene mit F6 oder dem Kommando #*D aufrufen können. Ausführlich wird er im Tustep-Wiki unter dem URL <http://tustep.wikispaces.com/Datei-Manager> behandelt – sogar mit einem kleinen Video, das Sie für Ihr Selbststudium herunterladen können. Ich kann es mir deshalb erlauben, ihn hier mit einem Screenshot nur zu erwähnen:



Für das *Löschen von Dateien* gibt es das weitere Kommando #LOESCHE, das Sie aber – selbstverständlich – *nur* dann einsetzen können, wenn Sie die betreffende(n) Datei(en) zuvor zum Schreiben angemeldet haben:

- #LOESCHE, DATEN=TEST1.TF
alternativ: #lo, test1.tf

löscht den Inhalt der Datei TEST1.TF, lässt aber die (dann leere) Datei fortbestehen.

- #LOESCHE, DATEI=TEST1.TF
alternativ: #lo,,test1.tf (beachten Sie die *beiden* Kommata!)

löscht die Datei TEST1.TF als ganzes.

Der TUSTEP-Editor

(Kleine Auswahl)¹

VORBEMERKUNG:

Das zentrale Modul dieses Erste-Schritte-Kurses ist schließlich der TUSTEP-Editor, den Sie mit dem Kommando #EDIERE aufrufen können. Er hat – unter anderem – die Spezifikationen DATEI=... und MODUS=..., wo Sie den Namen der Datei, die ediert werden soll, angeben und präzisieren können, ob die Datei als Text- oder Programmdatei behandelt werden soll, z. B. für Textdateien:

- #EDIERE, DATEI=TEST1.TF, MODUS=T
alternativ: #e, test1.tf, t

The screenshot shows the TUSTEP Editor window with the title "WEIGEL10*ST". The main area displays a XML document with the title "Dateianfang". The text is a story in German about a cannibal who ate children. The editor interface includes a menu bar, toolbars at the bottom, and status bars.

Nachdem Sie den Editor einmal aufgerufen haben, merkt sich das Programm, welche Datei mit welchem Modus zuletzt bearbeitet wurde. Das bedeutet, daß Sie beim nächsten Mal den Aufruf auf das bloße #E verkürzen können: Sie landen dann automatisch an der Stelle, an der Sie Ihre Arbeit zuvor unterbrochen haben.

¹ Die nachstehende Übersicht bietet einige ausgewählte Beispiele zu dem im Handbuch (Ausg. 2018), dort in der Grundlagenbeschreibung auf S. 251–398 systematisch abgehandelten Katalog der Editor-Anweisungen.

Neben einigen nützlichen Statusinformationen sind vor allem drei unterschiedliche Zonen des Fensters zu unterscheiden:

- (a) Den größten Teil nimmt das Textfeld in Anspruch, in dem der Inhalt der Datei, i. d. R. die Textdaten, steht.
- (b) Am linken Rand stehen die Satznummern: Der Inhalt der Datei ist nämlich in sog. ›Datensätze‹ unterteilt, die jeweils einen eindeutigen, durch die Numerierung bestimmten Platz haben; die Satznumerierung folgt dabei, solange Sie es nicht anders bestimmen, dem aus Büchern bekannten Schema `seitennr.zeilennr.` (*Achtung:* Das bedeutet jedoch *nicht*, daß hier bereits eine Entscheidung über eine eventuelle spätere Formatierung getroffen wird!)
- (c) Am unteren Rand die Anweisungszeile, auch ›COMMAND-LINE‹ genannt, in der Sie eine Vielzahl verschiedener Anweisungen geben können.

EINFACHE ANWEISUNGEN:

- b Beenden des Editors. Alternativ können Sie den Editor (nach Belieben) auch durch Drücken der Tasten `ESC` und `F4` verlassen.
- za Zeige die Datei vom Anfang an, d. h. der erste Satz der Datei wird an den Anfang des Bildschirms geschrieben und die folgenden Sätze dahinter.
- za, 1.3 Ähnlich wie vor, jedoch wird Satz 1.3 an den Anfang des Bildschirms geschrieben und die folgenden Sätze dahinter.
- zb, 1.5 Zeige bis Satz 1.5, d. h. dieser wird an das Ende des Bildschirms geschrieben.
- zu, 1.4 Zeige die Umgebung von Satz 1.4, d. h. dieser wird in die Bildschirmmitte geschrieben.
- 1!, 1.6 Lösche den Satz 1.6.
- 1!, (1.4, 1.7) Desgl., jedoch gilt die Anweisung für den *Bereich* der Sätze von 1.4 bis 1.7 einschließlich. Wenn Sie mit dieser Anweisung mehr als zehn Sätze löschen wollen, verlangt das Programm sicherheitshalber eine Bestätigung von Ihnen.
- k, 1.3, 1.6 Kopiere den Satz 1.3 hinter 1.6.
- k, (+1, -3), -2 Kopiere den Bereich vom *ersten*² bis zum *drittletzten* Satz der Datei hinter den *vorletzten* Satz.

2 Sie können in allen Anweisungen die Satznummern sowohl *absolut* (durch die Nummer selbst) als auch *relativ* bezeichnen: Die Angaben +1, +5, -3 etwa stehen für den ersten, fünften, drittletzten Satz einer Datei. Es ist zulässig, absolute und relative Angaben zu mischen.

k, quelle, (12.72, 14.200), 22.100

Kopiere aus der Datei QUELLE die Sätze 12.72 bis 14.200 in die aktuelle Editor-Datei so, daß sie dort hinter Satz 22.100 zu stehen kommen.

u Umstellen: Wie Kopieren, jedoch wird der umgestellte Satz oder Bereich an seinem ursprünglichen Standort gelöscht.

u, , 1.1 Umnumerieren einer Textdatei, so daß die Satznummern – bei 1.1 beginnend – lückenlos aufsteigen.

g (wie ›Gib!‹): Damit können Sie sich die letzte *erweiterte* Anweisung (vgl. übernächsten Abschnitt) einer bestimmten Gruppe zurückgeben lassen: ›ga‹ bedeutet etwa ›Gib die letzte Austausche-Anweisung auf den Bildschirm zurück‹, ›gz‹ fordert die letzte Zeige-Anweisung zurück usw. Mit den Anweisungen ›g-‹ (alternativ: F9) und ›g+‹ (alternativ: F10) blättern Sie ferner Ihre letzten Anweisungen nach hinten bzw. nach vorne durch (und damit Sie diese Anweisungen nicht von Hand eintippen müssen, sind sie vordefiniert auf die Funktionstasten F9 und F10 gelegt).

gg Diese Anweisung zeigt Ihnen die letzten achtzig Editor-Anweisungen, die Sie gaben, in Listenform. Sie können damit z. B. verifizieren, ob und welche Fehler Sie machten, und nachsehen, ob die von Ihnen gesuchte Anweisung noch im Memory des Editors ist. Um eine bestimmte Anweisung aus dieser Liste anzuwählen, haben Sie zwei Möglichkeiten: Der Befehl ›#nn‹ (wobei nn für die Nummer der gewünschten Anweisung in der Liste steht) führt besagte Anweisung sofort aus, während ›g#nn‹ die Anweisung zur Bearbeitung auf den Bildschirm zurückgibt.

ee Eintragen in eine leere Datei bzw. Eintragen ans Ende einer Datei, die bereits Text enthält. Mit dieser Anweisung wird der Bildschirm für die Eingabe freigemacht und der Cursor an den Anfang des Bildschirms geführt. Geben Sie nun Ihren Text *fortlaufend* ein, d. h. versuchen Sie zu diesem Zeitpunkt noch *nicht*, das Aussehen des späteren Ausdrucks auf dem Bildschirm herbeizuführen. Unterlassen Sie also jegliches Einrücken, Mehrfachblanks, Sperrungen, Zentrieren etc.; Sie werden bei der Erörterung des Programmmoduls #SATZ (ebenso bei #FORMATIERE) erfahren, wie Sie Leistungen dieser Art in TUSTEP realisieren können. Mit jedem Zeilenwechsel am Bildschirm während der Eingabe wird ein neuer Datensatz mit eigener Nummer angelegt. Wenn Sie am Ende des Bildschirms angelangt sind, können Sie gleichwohl weiterschreiben: Ihre bisherige Eingabe ›scrollt‹ dann nach oben und wird dabei zugleich auf der Platte gesichert.

- **Wichtig:** Wenn Sie im Eingabemodus sind, können Sie vorübergehend keine Anweisungen geben, weil alles, was Sie auf den Bildschirm schreiben, als ›Text‹ interpretiert wird. Wollen Sie also den Eingabemodus für andere Arbeiten beenden, müssen Sie zunächst Ihre letzten Eintragungen ›wegschicken‹ (= Betätigung der ENTER-Taste); eine weitere, ›leere‹ Eingabe (= erneutes Drücken der ENTER-Taste *ohne* einzutragenden Text) beendet jetzt den Eingabemodus. Alternativ können Sie den Eingabemodus auch – beispielsweise – durch Auslösen einer Zeigeanweisung mittels F1, F2 oder F4 beenden.

TASTENFUNKTIONEN³

EINFG Mit dieser Taste schalten Sie zwischen REPLACE- und INSERT-Modus des Editors hin und her.

PLUS⁴, END Damit löschen Sie eine Bildschirmzeile ab der aktuellen Cursor-Position bis zum Ende der Zeile.

PLUS, PLUS, END Diese Verdoppelung von PLUS kehrt den vorherigen Befehl um: Der mit PLUS, END gelöschte Text wird an der aktuellen Cursor-Position wieder eingefügt.

PLUS, Pos1 Löscht den Text der Zeile ab der aktuellen Cursor-Position bis zum Anfang der Zeile.

PLUS, PLUS, Pos1 Kehrt ein zuvor gegebenes PLUS, HOME um (vgl. die Hinweise bei PLUS, PLUS, END)

PLUS, Entf Löscht den Datensatz, in dem sich der Cursor gerade befindet.

PLUS, Bild↓ Fügt oberhalb der aktuellen Datenzeile einen neuen Datensatz ein.

PLUS, Bild↑ Löscht die aktuelle Datenzeile vom Bildschirm.

PLUS, RETURN Splittet den aktuellen Datensatz an der Cursorposition und beginnt an dieser Stelle einen neuen Datensatz mit eigener Nummer.

PLUS, PLUS, Bild↓ Dupliziert die aktuelle Bildschirmzeile in einen neuen Datensatz mit eigener Nummer.

F1-F40 FUNKTIONSTASTENBELEGUNGEN: Wenn Sie komplexe Anweisungen, wie sie im folgenden beschrieben werden, häufiger benötigen, so empfiehlt es sich, sie der Einfachheit halber auf Funktionstasten zu

3 Die nachstehend für Windows-Rechner beschriebenen Funktionen können Hardware- und Betriebssystem-abhängig differieren.

4 Gemeint ist die mit dem Zeichen ›+‹ beschriftete Taste im NUM-Block am rechten Rand der Tastatur. Alternativ kann auch die normale Plus-Taste im alphanumerischen Teil der Tastatur, kombiniert mit der STRG-Taste, verwendet werden – wichtig auf Notebooks, wo es i.d.R. keinen NUM-Block in der Tastatur gibt.

legen. Beispielsweise belegen Sie die Taste F24 folgendermaßen mit einer erweiterten Zeigeanweisung:

F24=zn,,, | \Zeus | \Jupiter | \Hera | \Juno |

Zum Aufrufen der Funktionen F1-F10 genügt das Drücken der entsprechenden Funktionstasten, für F11-F20 benutzen Sie dieselben Taten in Verbindung mit der Shift-Taste (= SHIFT+Fn), für F21-F30 in Verbindung mit CTRL (bzw. STRG) und für F31-F40 in Verbindung mit ALT. Alternativ können Sie die Funktionen auch durch die Eingabe ihres Namens, z. B. >F24<, in der Command-Line aufrufen.

- STRG+r (= RESHOW) stellt beispielsweise nach fehlerhaften Eingaben den letzten korrekten Bildschirm wieder her.
- CTRL+f Damit können Sie die Farbeinstellung des Editors nach persönlichem Geschmack verändern.
- CTRL+g Damit können Sie Form und Geschwindigkeit des Cursors einstellen.
- CTRL+l Damit können Sie Anzahl, Länge und Umbruch der Datensätze für die Bildschirmdarstellung im Editor bestimmen.

ERWEITERTE ANWEISUNGEN⁵

a,,, | und | oder |

Tausche in der ganzen Datei alle >und< gegen >oder< aus. Wenn Sie diese Anweisung mit RETURN losschicken, wird Ihr Text nach der Zeichenfolge >und< durchsucht und an der ersten gefundenen Belegstelle der Austauschvorgang am Bildschirm durch die Zustände >vorher< und >nachher< dargestellt. Auf die Frage des Programms

»SOLL AUSGETAUSCHT WERDEN?«

können Sie vier Antworten geben:

- j : ja (= tausche aus, suche *nicht* weiter nach anderen Textstellen);
 - RETURN: tausche aus, suche weiter nach anderen Textstellen, um dann erneut zurückzufragen;
 - n : nein (= tausche *nicht* aus, suche aber weiter nach anderen Textstellen);
 - w : weiter (= tausche ab hier alles weitere aus, *ohne* zurückzufragen).
- *Hinweis:* Sie können beliebig viele solcher Austausch-Zeichenfolgepaare zugleich angeben. Reicht die COMMAND-LINE dafür nicht aus, so löschen Sie den Bildschirm mit PLUS, ×⁶; danach können Sie die An-

⁵ Diese Anweisungen heißen so, weil sie durch *Bedingungen*, unter denen sie ausgeführt werden sollen, >erweitert< sind.

⁶ Mit dem Zeichen >×< ist das Malkreuz im NUM-Block im rechten Teil der Tastatur gemeint.

weisung an den Anfang des Bildschirms schreiben und mit `ENTER` abschicken.

- **Veto-Zeichenfolgen:** Um beispielsweise zu vermeiden, daß die Zeichenfolge `>und<` auch in größeren Wörtern wie `>Schund<` oder `>hundert<` ausgetauscht wird, können Sie sog. Veto-Zeichenfolgen angeben, in denen nicht ausgetauscht werden soll:

`a,,, | | oder | | {&a} und | und{&a} |`

Das gewählte Beispiel fordert von dem Programm, daß die Zeichenfolge `>und<` gegen `>oder<` ausgetauscht werden soll, jedoch *nicht*,⁷ wenn vor oder hinter `>und<` ein beliebiger Buchstabe steht. Der *Stellvertreter* für `>beliebiger Buchstabe<` ist `{&a}`, weitere Joker dieser Art sind:

`\0` : für `>beliebige Ziffer<`,
`\A` : für `>beliebiger Großbuchstabe<`,
`\a` : für `>beliebiger Kleinbuchstabe<`,
`?` : für `>beliebiges Zeichen<`.

- **Hinweis:** Sie können außerdem Zeichen- und Stringgruppen Ihrer Wahl (wichtig für Pattern-Matching) *selbst* definieren; beachten Sie hierzu die Hinweise der Grundlagenbeschreibung im Abschnitt »Zeichen- und Stringgruppen definieren/löschen/abfragen« auf S. 294ff des Handbuchs 2018.

Mit diesen Anweisungen durchforsten Sie Ihren *ganzen* Text. Wollen Sie das beschränken, geben Sie zwischen den ersten beiden Kommata nach dem `a` einen *Bereich* an – also diejenigen Dateisätze, in denen ausgetauscht werden soll. Wahlweise haben Sie auch die Möglichkeit, *innerhalb* der Sätze nur auf bestimmten Spaltenpositionen auszutauschen, z. B.:

`a,(2.1,2.100),1-9, | \Verfasser | Autor |`

Diese Anweisung veranlaßt das Programm, in den Sätzen 2.1 bis 2.100 nur dann die Zeichenfolge `>Verfasser<` gegen `>Autor<` auszutauschen, wenn sie ganz links im Satz, auf den Spalten 1 bis 9, vorkommt.

Analog können Sie auch bei anderen Anweisungen verfahren:

`1!,(+3,+7),,|++|`

Löscht im Bereich vom dritten bis siebten Satz alle diejenigen Sätze, in denen irgendwo die Zeichenfolge `>++<` vorkommt.

`zn,,,|!||!*|!?:|`

Zeigt nur diejenigen Sätze am Bildschirm, in denen ein `>!<` vorkommt, nicht jedoch, wenn es in den Kombinationen `>!*<` und `>!?:<` steht.

⁷ Der Operator `>nicht<` wird durch die Verdoppelung des Begrenzungszeichens `><` hinter `>oder<` gesetzt.

k,murks,,+10,, | Beethoven | Mozart | Haydn |

Kopiert aus der Datei MURKS alle diejenigen Sätze in die aktuelle Editor-Datei (dort hinter den zehnten Satz), in denen die Namen ›Beethoven‹, ›Mozart‹ und ›Haydn‹ vorkommen.

Anhand dieser Beispiele sollten Sie sich im Kapitel ›Editor-Anweisungen‹ der Grundlagenbeschreibung leicht weiter orientieren können. Ein Tip noch: Sehr anspruchsvolle Möglichkeiten der Editor-Nutzung eröffnen Ihnen die Hinweise zu ›Zahlenwertbedingungen‹, ›Häufigkeitsbedingungen‹, ›Umgebungsbedingungen‹ und ›Verweise‹ auf den Seiten 706 bis 711 der Grundlagenbeschreibung im Handbuch 2018.



