

Anwendungs-Programmsystem DAPROA

Format der Originaleingabedatei für den Verarbeitungsprogrammbaustein NORM

Version 5 .1

Stand: 31.10.2011

Ausgangspunkt:

Ausgangspunkt für die Arbeit mit DAPROA sind Datenbestände bzw. Originalmesswertdateien, die mittels Text-Editor, Entladen aus Office-Produkten oder Datenbanken oder durch andere Datenbereitstellungssoftware erzeugt wurden.

Der Baustein NORM ist das Eingangstor zu DAPROA. Er liest diese Originaleingabedatei ein und erstellt daraus die Standard- bzw. Basisdateien für die weitere Verarbeitung in DAPROA.

Falls das Projekt erstmals bearbeitet wird, trägt NORM das Projekt in die Projektliste prjlst_abl.txt ein und legt die Projektverzeichnisse in **..\daten** und **..\ergebnis** an.

Der Anwender muss also **nur die Originaleingabedatei** im richtigen Format bereitstellen und NORM das Verzeichnis, in welchem sie sich befindet, mitteilen.

Alle anderen Dateien, welche DAPROA braucht, und die in den Dokumenten beschrieben werden, werden von DAPROA selbst angelegt und verwaltet.

Format:

Originaleingabe-Messwert- bzw. -Daten-Datei: **projektdaten.txt**

- **"freiname".txt** Original-Datendatei im Textformat

Prinzipiell besteht diese Datei aus zwei vorangestellten **Informationssätzen** und der **Datenmatrix**.

Die **Datenmatrix** ist zeilenweise aufgebaut:

In den Zeilen stehen die Datensätze mit je einem Variablenwert, durch Trennzeichen voneinander getrennt.

In den Spalten befinden sich demzufolge die Variablen.

Die Anzahl der Variablen ist gleich der Anzahl der Trennzeichen plus 1

In der Datenmatrix dürfen zwischen den Trennzeichen nur Zahlenwerte stehen (auch in exponentieller Darstellung). Die Ausnahme sind ein Stern(*) oder eine leere Zeichenkette als Hinweis auf einen fehlenden Wert

1.Satz der Datenmatrix	1.Wert Variable 1	1.Wert Variable 2	...	1.Wert Variable N
2.Satz der Datenmatrix	2.Wert Variable 1	2.Wert Variable 2	...	2.Wert Variable N
...
m-ter Satz der Datenmatrix	m-ter Wert Variable 1	m-ter Wert Variable 2	...	m-ter Wert Variable N

Nachfolgend der Aufbau der Originaleingabedatei in der Reihenfolge der Datensätze

Inhalt	Bedeutung
Titel oder Informationszeile	 leitet eine Informationszeile mit maximal 79 Zeichen ein. Wird dieser Satz, gekennzeichnet mit , genutzt, muss er als erster Satz am Dateianfang stehen. Falls dieser Satz nicht vorhanden ist, setzt NORM Standardausdrücke für diesen Satzwert.
<N>alpha;beta;gamma;delta	<N> leitet den Satz mit den Namen der Variablen ein. Gleichzeitig legt er fest in welcher Reihenfolge die Variablen in der nachfolgenden Datenmatrix stehen und bestimmt die Anzahl der Variablen in der Datenmatrix. In den Datensätzen der Datenmatrix müssen genau so viele Variablenwerte stehen, wie Variablennamen eingetragen wurden. In beiden Satzarten wird der Satz nicht durch ein Trennzeichen beendet. Die Namenslänge beträgt maximal 15 Zeichen. Als Zeichen dürfen nur Buchstaben, Ziffern und der Bindestrich verwendet werden(siehe auch untere Anmerkung). Wird der Satz, gekennzeichnet mit <N>, genutzt, muss er direkt nach dem Satz mit dem und vor dem ersten Datensatz stehen. Falls dieser Satz nicht vorhanden ist, setzt NORM Standardausdrücke für die Variablennamen. Die Anzahl der Variablen wird dann aus dem ersten Datensatz der Datenmatrix ermittelt. Im Beispiel ist das Trennzeichen zwischen den Variablennamen ein Semikolon(;) . Damit hat die Datenmatrix vier Variable. Mögliche Trennzeichen sind : Komma , Semikolon ; Querstrich / Doppelkreuz #
26.1 ; -3.5 ; 20 ; 11.9	korrekter 1. Wertesatz in der Datenmatrix. das Trennzeichen muss genau das gleiche Trennzeichen wie im Satz <N> Variablennamen sein. Als Dezimalzeichen wurde der Punkt gewählt. Mögliche Dezimalzeichen sind Komma und Punkt: , . Wäre als Trennzeichen das Komma gewählt worden, muss das Dezimalzeichen ein Punkt sein.
1.33D02 ; -0.5E-01 ; 0. ; 1.2	korrekter 2. Datensatz, der erste und zweite Wert wurden in der Exponentialdarstellung geschrieben. In solchen Fällen muss der Punkt für alle Werte in der Datenmatrix als Dezimalzeichen dienen.
1.6 ; ; 3.4 ; -12.0	korrekter 3. Datensatz, der zweite Wert ist nicht vorhanden. Dieser Fehlwert wird durch die Leerstellen zwischen dem Trennzeichen angezeigt. Bei der Verarbeitung dieser Datei wird er durch den Wert -1.0 * 10⁺⁵⁰ ersetzt .
1.4D02 ; -1.E-01 ; 3.4 ; *	korrekter 4. Datensatz, der vierte Wert ist nicht vorhanden. Dieser Fehlwert wird durch einen Stern(*) angezeigt. Bei der Verarbeitung dieser Datei wird er durch den Wert -1.0 * 10⁺⁵⁰ ersetzt .
1.3 ; x ; 1,76 ; -123	falscher 5. Datensatz, im zweiten Wert steht kein Zahlenwert. Beim dritten Wert befindet sich ein Komma als Dezimalzeichen, obwohl ein Punkt vereinbart wurde. Jeder dieser Fehler, aber auch eine falsche Anzahl von Variablen, führt zum Abbruch der Verarbeitung von NORM. Der Fehler wird angezeigt.

Wenn der Baustein NORM zur Verarbeitung der Originaleingabedatei aufgerufen wird, sind die verwendeten Trennzeichen und Dezimalzeichen im Dialog einzugeben. Weicht in den zu verarbeitenden Sätzen eines dieser Zeichen von den festgelegten ab, wird die Verarbeitung abgebrochen und eine Meldung gegeben.

Beim Bilden von Namen durch den Anwender sind unbedingt folgende Regeln einzuhalten:

Die Variablennamen, Projektnamen und auch andere selbstbenannte Modell- und Dateinamen müssen aus dem Zeichenvorrat des ASCII-Zeichen-Codes (Dezimal) von 48-57 (Ziffern), 65-90 (Großbuchstaben) und 97-122 (Kleinbuchstaben) sowie 45 (Bindestrich) gebildet werden. Umlaute und ß sind nicht erlaubt.

Bei Nichtbeachtung dieser Forderung kommt es zu schweren Störungen bei der Dateiarbeit !

Insbesondere die Verwendung der Zeichen 95 , 46 , 44 , 59 , 43 , 42 , 32 (_ Unterstrich, . Punkt, , Komma, ; Semikolon, + plus, * mal, leer sowie Querstriche, Klammern und andere Sonderzeichen) führt zu Fehlern, da sie u.a. programmintern als Positionszeichen zum Aufbau der Dateinamen dienen oder auch in Steuerelementen benutzt werden.

Beispiel:

Der Anfang der Eingabedatei **ammon.txt** mit den Originaldaten hat folgendes Aussehen:

1.Satz mit Problembezeichnung:	Abhaengigkeit der Loeslichkeit eines Stoffes von Druck und Temperatur
2.Satz mit Variablennamen:	<N>Temp;Druck;Loeslkt
1.Datensatz:	-20. ;1 ; .615
2.Datensatz:	0. ;1 ; .438
	20. ;1 ; .325
	40. ;1 ; .228
	60. ;1 ; .14
	80. ;1 ; .062
	0. ;2 ; .566
	20. ;2 ; .418
	40. ;2 ; .314
n-ter Datensatz	60. ;2 ; .225

Fehlwerte werden durch einen * ;* ;
oder Leerzeichen ; ; zwischen den
Trennzeichen gekennzeichnet

Neben den Fehlerkennzeichen dürfen in den Datensätzen
nur numerische Zahlendarstellungen auftreten, z.B.: auch
1.0E-02 oder 0.123D02 (Punkt als Dezimaltrennzeichen)

Das Kennzeichen steht für eine nachfolgende Problembezeichnung.

Im Satz nach dem Kennzeichen <N> werden die Namen der Variablen/Datengrößen genannt, getrennt durch das Trennzeichen. Aus der Anzahl der Namen, falls Satz <N> vorhanden ist, oder dem ersten Datensatz wird die Anzahl der Variablenwerte in den nachfolgenden Datensätzen abgeleitet.

Die Anzahl der Namen und die Anzahl der Variablenwerte müssen übereinstimmen.

Wenn Sätze mit den Kennzeichen und <N> vorhanden sind, müssen sie in genau obiger Reihenfolge in der Datei stehen.

Wenn Sätze mit den Kennzeichen oder <N> nicht vorhanden sind, bildet NORM allgemeine Angaben zu deren Inhalten.
Die Zeichenlänge eines Namens oder Variablenwertes darf 15 Zeichen nicht überschreiten!

Diese Datei wird dem Baustein NORM bekannt gegeben und von NORM verarbeitet.

Es werden durch NORM die Unterverzeichnisse gebildet: `..\daten\ammon\` und `\ergebnis\ammon\`

Als Ergebnis der Verarbeitung mit NORM entstehen im Verzeichnis `..\daten\ammon\` folgende Projektbasisdateien bzw. –daten mit den auszugsweise dargestellten Inhalten

ammon_abd.txt

```
<norm      ><Problembezeichnung >L=0.361 -0.00802*T +0.11694*D +0.00005*TT -0.00333*DD -0.00066*TD
<norm      ><Eingabedatei       >c:\temp\ammon.txt
<norm      ><Trennzeichen           >;
<norm      ><Dezimalzeichen           >.
```

ammon_std.txt

```
L=0.361 -0.00802*T +0.11694*D +0.00005*TT -0.00333*DD -0.00066*TD
   3      44      0
Temp      -0.200000D+02 0.100000D+03 0.518182D+02 0.334992D+02      44
Druck      0.100000D+01 0.800000D+01 0.431818D+01 0.231047D+01      44
Loeslkt    0.620000D-01 0.935000D+00 0.386295D+00 0.211493D+00      44
 1.000000 0.250753 1.000000 -0.791975 0.305536 1.000000
```

ammon_mtx.txt

Anfang der Datei

```
1 -0.20000000D+02 0.10000000D+01 0.61500000D+00
2 0.00000000D+00 0.10000000D+01 0.43800000D+00
3 0.20000000D+02 0.10000000D+01 0.32500000D+00
4 0.40000000D+02 0.10000000D+01 0.22800000D+00
5 0.60000000D+02 0.10000000D+01 0.14000000D+00
6 0.80000000D+02 0.10000000D+01 0.62000000D-01
7 0.00000000D+00 0.20000000D+01 0.56600000D+00
8 0.20000000D+02 0.20000000D+01 0.41800000D+00
9 0.40000000D+02 0.20000000D+01 0.31400000D+00
10 0.60000000D+02 0.20000000D+01 0.22500000D+00
```

ammon_nqd.txt Anfang der Datei

```
L=0.361 -0.00802*T +0.11694*D +0.00005*TT -0.00333*DD -0.00066*TD
   3      44      0
Temp      Druck      Loeslkt
-0.20000000D+02  0.10000000D+01  0.61500000D+00
 0.00000000D+00  0.10000000D+01  0.43800000D+00
 0.20000000D+02  0.10000000D+01  0.32500000D+00
 0.40000000D+02  0.10000000D+01  0.22800000D+00
 0.60000000D+02  0.10000000D+01  0.14000000D+00
 0.80000000D+02  0.10000000D+01  0.62000000D-01
 0.00000000D+00  0.20000000D+01  0.56600000D+00
 0.20000000D+02  0.20000000D+01  0.41800000D+00
 0.40000000D+02  0.20000000D+01  0.31400000D+00
 0.60000000D+02  0.20000000D+01  0.22500000D+00
```

ammon_feh.txt (in diesem Beispiel keine fehlenden Werte)

Protokoll der fehlenden Werte: