

Erklärung zum Text

SETS:	fettgedruckter Text bezeichnet LINGO-Syntax (v.a. Befehle)
<i>[Ausdruck]</i>	kursiver Text bezeichnet Text, der innerhalb der LINGO-Syntax frei wählbar ist
variable	fetter und kursiver Text bezeichnet konkrete Syntax-Beispiele (Namen und Zahlen sind frei gewählt)

Grundstruktur eines LINGO-Modells:

MODEL:

SETS:

[Definitionen von Variablen mit Wertebereichen, so genannte "Sets"]

ENDSETS

[Zielfunktionsbedingung(en), Restriktionen und Wertebereichsfestlegungen]

DATA:

[Datenteil]

ENDDATA

END

Die hier genannten "Gerüstbefehle" sind die einzigen, die nicht mit Strichpunkt abgeschlossen werden. Allerdings steht nach **MODEL**, **SETS** und **DATA** ein Doppelpunkt!

MODEL ... END kann auch weggelassen werden.

Der Bereich **SETS: ... ENDSETS** ist nur erforderlich, wenn Sets zu definieren sind. "Normale" Variablen sind nicht zu definieren.

Der Bereich **DATA: ... ENDDATA** ist nur erforderlich, wenn bestimmte Konstanten vorgegeben werden sollen. Das können sowohl Set-Variablen als auch Einzelvariablen sein. Die Festlegung von Konstanten erfolgt (beispielhaft) folgendermaßen:

DATA:

x = 5;

y = 30 10 60 20;

z = 0.2 0.1 0.7 0.3 0.5 0.2;

ENDDATA

Dabei steht x hier stellvertretend für eine Einzelvariable, y für ein Set mit 4 Elementen und z für ein zweidimensionales Set mit 2·3 Elementen (2x3-Matrix).

Grundrechenarten:

+ - * / ^ (Potenzen)

LINGO rechnet nach üblicher Punkt-vor-Strich-Rechnung. Dabei gehen Potenzen vor Punkt und Punkt vor Strich. Die Reihenfolge kann durch gewöhnliche Klammern außer Kraft gesetzt werden.

Zielfunktionen:

max = [Zielfunktionsausdruck];

Maximierung einer Zielfunktion

min = [Zielfunktionsausdruck];

Minimierung einer Zielfunktion

Kommentar einfügen:

![Kommentartext];

Kommentare müssen durch Strichpunkt abgeschlossen werden. Innerhalb des Kommentartextes darf kein Strichpunkt stehen, da LINGO sonst den Kommentar für beendet hält.

vergleichende Operatoren:

= >= <= > <

Die vergleichenden Operatoren sind intuitiv einsetzbar. Allerdings muss beachtet werden, dass > und < wie >= bzw. <= behandelt werden!

wichtigste mathematische Funktionen:

@SUM([Ausdruck]);	liefert als Ergebnis die Summe des Ausdrucks
@MAX([Ausdruck]);	liefert als Ergebnis das Maximum des Ausdrucks
@MIN([Ausdruck]);	liefert als Ergebnis das Minimum des Ausdrucks
@ABS(x);	liefert den Betrag von x, also x
@LOG(x);	liefert den natürlichen Logarithmus, also $\ln x$
@EXP(x);	liefert Werte der Exponentialfunktion e^x

Wertebereiche für Variablen festlegen:

Standardmäßig sind für Variablen positive reelle Lösungen zugelassen. Soll dieser Wertebereich eingeschränkt werden, erfolgt dies über einen der folgenden Befehle:

@GIN(x);	ganzzahlige positive Werte
@BIN(x);	nur 0 und 1
@FREE(x);	positive und negative reelle Werte
@BND([Zahl Untergrenze], x, [Zahl Obergrenze]);	Werte innerhalb eines bestimmten Intervalls

Definition von Variablen mit mehreren Elementen (Sets und Matrizen):

- Setvariable s mit n Elementen definieren

```
SETS:  
    set1 /1..n/: s;  
ENDSETS
```

set1 ist ein frei wählbarer Variablenname. Es wird ein Set mit Elementen von 1 bis n definiert (analog zu Arrays in anderen Programmiersprachen). n ist eine natürliche Zahl, die konkret festgelegt werden muss.

- Matrixvariable m mit z Zeilen und s Spalten definieren

```
SETS:  
    zei /1..z/;  
    spa /1..s/;  
    mat (zei,spa): m;  
ENDSETS
```

Auch hier sind zei, spa und mat frei wählbare Namen. Für z und s sind konkrete Zahlen einzusetzen!

Variablen mit mehreren Elementen ansprechen

- Aufsummieren aller Elemente eines Sets

```
@SUM(set1(i): s(i));
```

- Aufsummieren aller Elemente einer Matrix

```
@SUM(mat(i,j): m(i,j));
```

Dieser Befehl funktioniert so nur, um alle Matrixelemente aufzusummieren.

- Setelemente einer Setvariable s der Reihe nach ansprechen

```
@FOR(set1(i): [mathematischer Ausdruck, der "s(i)" enthält, z.B. "s(i)<=10"]);
```

- Matrixelemente einer Matrixvariable m zeilenweise über alle Spalten ansprechen

```
@FOR(zei(i): @FOR(spa(j): [mathematischer Ausdruck, der "m(i,j)" enthält]));
```

Statt @FOR können die Elemente der Reihe nach auch über @SUM, @MAX oder @MIN angesprochen werden. Es können auch spaltenweise alle Zeilen angesprochen werden oder einzelne Spalten oder Zeilen festgehalten werden (z.B. mit "m(3,j)").