

# LISP

## Eine elementare Einführung in die Programmierung nichtnumerischer Aufgaben

*von*

Prof. Dr. Dieter Müller  
*Universität Hannover*



**Bibliographisches Institut Mannheim/Wien/Zürich**  
B. I.-Wissenschaftsverlag

## Inhaltsverzeichnis

<u>1</u>	<u>Das Gerüst der Sprache LISP</u>	13
1.1	Erstes Beispiel, Notation arithmetischer Ausdrücke	14
1.2	Darstellung von Formeln als Datenstrukturen	15
1.3	Operationen und Ablaufstrukturen	20
1.4	Das LISP-Programm für <b>diff</b>	22
1.5	LISP-Programme als Listenstrukturen	24
1.6	Speicherverwaltung und Garbage Collection	27
1.7	Die Interpreterfunktion EVAL	28
<u>2</u>	<u>LISP-Datenstrukturen</u>	30
2.1	Atome, symbolische Ausdrücke, Listen	30
2.2	Zahlen in LISP	36
<u>3</u>	<u>Elementare LISP-Funktionen, Formen, der Auswerteprozep (I), die Funktion EVAL</u>	38
3.1	Elementare LISP-Funktionen	38
3.2	Beispiele von Funktionsaufrufen	43
3.3	Formen	48
3.4	Der Auswerteprozep (I), Skizze von EVAL	49
	<u>Bindung von Namen an Objekte, P-Listen, die A-Liste, LAMBDA-Ausdrücke und Funktionsdefinitionen</u>	51
4.1	Permanente Bindung: die P-Liste eines Atoms	52
4.2	Die vorübergehende Bindung auf der A-Liste	54
4.3	LAMBDA-Ausdrücke	56
4.4	Die LAMBDA-Konversion	60
	<u>Die Konstruktion rekursiver LISP-Programme</u>	63
5.1	Grundregeln für den Aufbau rekursiver Definitionen, erste Beispiele	63
5.2	Weitere Beispiele, Effizienzbetrachtungen	66
5.3	Hilfsparameter und Hilfsfunktionen	71
5.4	Effizienzverbesserung durch anonyme LAMBDA-Ausdrücke	76
5.5	Die Schönheit der Rekursion, LINDENMAYER-Systeme	80
	<u>Iterative LISP-Programme, die Funktion PROG</u>	83

<u>Ein- und Ausgäbetfunktionen: READ, PRINT und TBRPRI</u>	88
7.1 Die Wirkung von READ, PRINT und TERPRI	88
7.2 Verwendung von PRINT bei der Fehlersuche	89
7.3 Ein Programm zur Eingabe von Fahrplandaten	91
<u>8 Ein Anwendungsbeispiel: Fahrplanauskunft</u>	97
8.1 Die Wegsuche	97
8.2 Die Zugsuche	104
<u>9 Erweiterung des Instrumentariums: FEXPR-Funktionen, funktionale Argumente, MAP-Funktionen</u>	115
9.1 Funktionen mit beliebig vielen, nicht notwendigerweise ausgewerteten Argumenten	115
9.2 Funktionale Argumente, MAP-Funktionen	118
9.3 Gebundene und freie Variable, das FUNARG-Problem	123
<u>10 Wertzuweisungen: Die gefährlichen Funktionen</u>	
RPLACA, RPLACD	131
10.1 RPLACA, RPLACD	132
10.2 Ein einfacher LISP-Editor	133
<u>11 Fehlersuche durch Tracing</u>	138
<u>12 Ein einfaches Spreadsheet-System</u>	142
12.1 Spreadsheet für einfache Tabellenberechnungen	142
12.2 Simulation mit dem Spreadsheet	146
<u>13 Mustererkennung (pattern matching)</u>	151
<u>14 Der Auswerteprozess (II): LISP-Interpreter in LISP</u>	156
14.1 Der klassische LISP-Interpreter	157
14.2 Einige Interpreter für LISP-ähnliche Sprachen	160
<u>15 LISP und der funktionale Programmierstil</u>	174
15.1 Das Curryfizieren von Funktionen	174
15.2 Der X-Kalkül, Auflösung einer Gleichung nach einer Funktion	177

<u>16 Bemerkungen zur Programmiermethodik in LISP</u> . . . . .	184
16.1 LISP und die Prinzipien der strukturierten Programmierung . . . . .	185
16.2 Die top-down-Entwicklung von Programmen . . . . .	188
16.3 Beweisen von Programmeigenschaften . . . . .	190
<u>Anhang: Die wichtigsten Systemfunktionen von LISP</u> . . . . .	194
Verwendete Zeichen und Schreibweisen . . . . .	198
Literaturhinweise . . . . .	200
<u>Sachregister</u> . . . . .	204