

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Warum LISP?	1
1.2	Zur Entwicklungsgeschichte von LISP	2
1.3	Was sind die Grundbausteine der Programmierung?	4
1.4	Grundbausteine der Programmierung in LISP	7
1.5	Der Inhalt dieses Lehrbuches	9
1.6	Literaturhinweise	10
1.7	Übungen	11
<b>2</b>	<b>Einfache Terme und der Umgang mit Objekten</b>	<b>12</b>
2.1	Einfache Terme	12
2.2	Bezugnahme auf Objekte	14
2.3	Umgang mit Objekten	14
2.4	Die externe Repräsentation von Objekten	15
2.5	Übungen	16
<b>3</b>	<b>Primitive Datenobjekte — zugeordnete Grundfunktionen und externe Repräsentation</b>	<b>17</b>
3.1	Zahlen	18
3.1.1	Funktionen, die für alle Typen von Zahlen verwendbar sind	18
3.1.2	Ganze Zahlen	19
3.1.3	Gleitkommazahlen	21
3.2	Zeichen	22
3.3	Zeichenketten	23
3.4	Symbole (Literal-Atome)	25
3.5	Funktionsobjekte	27
3.6	Paare (Cons-Objekte)	29
3.7	Weitere Datenstrukturen in verschiedenen LISP-Dialekten	36
3.8	Übungen	36
<b>4</b>	<b>Das LISP-System im einfachen Dialog</b>	<b>38</b>
4.1	Terme sind Listen	38
4.2	Der Dialog mit dem System	39
4.3	Die Umgebung verändert sich	41

4.4	Verschachtelung von Termen . . . . .	41
4.5	Auswertung von Termen . . . . .	42
4.6	Quotierung — Beabsichtigte Suspendierung der Auswertung . . . . .	45
4.7	Die Wahrheitswerte T und NIL . . . . .	47
4.8	Übungen . . . . .	48
<b>5</b>	<b>Funktionsdefinition als Abstraktion über Termen . .</b>	<b>51</b>
5.1	Was ist Abstraktion . . . . .	51
5.2	Funktionale Abstraktion in LISP . . . . .	53
5.3	Auswertung von Termen mit LAMBDA-Ausdrücken als Funktion . . . . .	57
5.4	Beziehungen zwischen Umgebungen — globale und lokale Effekte . . . . .	58
5.5	Bedingte Ausdrücke — Verwendung und Verknüpfung von Prädikaten . . . . .	61
5.6	Ein größeres Beispiel: Die Baukastenwelt („Blocks World“) . . . . .	65
5.7	Literaturhinweise . . . . .	70
5.8	Übungen . . . . .	71
<b>6</b>	<b>Komplexe Datenstrukturen und ihre Verarbeitung — Rekursion und Iteration . . . . .</b>	<b>72</b>
6.1	Zusammensetzen (Kombinieren) von Daten zu komplexeren Strukturen . . . . .	72
6.2	Datenabstraktion . . . . .	75
6.3	Lineare Datenstrukturen und lineare Rekursion . .	81
6.4	Hierarchische Datenstrukturen und Baum-Rekursion . . . . .	86
6.5	Arithmetische Ausdrücke als hierarchische Strukturen — Anwendungsbeispiel: Vereinfachung .	89
6.6	Rekursive Funktionen zur Manipulation von S-Ausdrücken . . . . .	92
6.6.1	Funktionen für allgemeine Listenstrukturen . . . .	93
6.6.2	Funktionen für Assoziationslisten . . . . .	94
6.6.3	Funktionen für echte Listen . . . . .	96
6.7	Literaturhinweise . . . . .	97
6.8	Übungen . . . . .	97
<b>7</b>	<b>Kontrollstrukturen, Spezialformen und Macros . . .</b>	<b>102</b>
7.1	Was sind Kontrollstrukturen? . . . . .	102
7.1.1	Kontrollstrukturen auf der Ebene der Anweisungen .	103
7.1.2	Kontrollstrukturen auf der Ebene der Programm- Einheiten . . . . .	104

7.1.3	Blöcke und Unterprogramme als Kontrollstrukturen . . . . .	104
7.1.4	Behandlung von Ausnahmefällen und nichtlokale Ausgänge aus Funktionen . . . . .	106
7.1.5	Coroutinen und parallele Prozesse . . . . .	110
7.1.6	Der Faden der Ariadne, oder: Kontrollstrukturen und Suchverfahren in Graphen . . . . .	112
7.2	Wozu werden Spezialformen benötigt? . . . . .	118
7.3	Die verallgemeinerte Variablenliste . . . . .	122
7.4	Definition von Spezialformen: FEXPRs, FLAMBDAs und NLAMBDAs . . . . .	123
7.5	Definition von Spezialformen: Macros und MLAMBDAs . . . . .	125
7.6	Memo-Funktionen: Ein Beispiel für Macros . . . . .	128
7.7	Sind FEXPRs wirklich erforderlich? . . . . .	130
7.8	Literaturhinweise . . . . .	131
7.9	Übungen . . . . .	132
<b>8</b>	<b>Ein- und Ausgabe . . . . .</b>	<b>133</b>
8.1	Dateien als Objekte . . . . .	133
8.2	Spezifikation von Dateien . . . . .	136
8.3	Allgemeines über die Ein-/Ausgabe . . . . .	137
8.4	Eingabe . . . . .	139
8.5	Lexikalische Verarbeitung und syntaktische Klassifikation von Zeichen . . . . .	141
8.6	Read-Macros . . . . .	144
8.7	Ausgabe . . . . .	151
8.8	Ströme und die Simulation von Ein-/Ausgabe-Vorgängen durch Pseudo-Dateien . . . . .	158
8.9	Direktzugriff auf Dateien . . . . .	160
8.10	Literaturhinweise . . . . .	161
8.11	Übungen . . . . .	162
<b>9</b>	<b>Funktionsobjekte . . . . .</b>	<b>163</b>
9.1	Funktionale: Funktionen, die Funktionen als Argumente haben . . . . .	163
9.2	Probleme mit Funktionsobjekten . . . . .	172
9.2.1	Erste Problembeschreibung . . . . .	172
9.2.2	Zweite Problembeschreibung. . . . .	177
9.2.3	Technische Gründe für die Bevorzugung der „dynamischen“ Semantik . . . . .	182
9.2.4	Historische Gründe für die Bevorzugung der „dynamischen“ Semantik . . . . .	182
9.2.5	Lösungen des Umgebungsproblems: FUNARGs, Closures und lexikalische Umgebungs konstruktion . . . . .	183

9.3	Generatoren und Ströme . . . . .	188
9.4	Objekt-orientierte Programmierung und Prozeßsimulation . . . . .	190
9.5	Simulation durch Funktionsobjekte mit zugeordneter Umgebung . . . . .	193
9.6	Das Hafenmodell: Ein konkretes Simulationsbeispiel . . . . .	198
9.7	Literaturhinweise . . . . .	202
9.8	Übungen . . . . .	203
<b>10</b>	<b>Generische Funktionen und datengesteuerte Programmierung . . . . .</b>	<b>204</b>
10.1	Gründe für eine weitere Abstraktionsstufe . . . . .	204
10.2	Generische Funktionen . . . . .	206
10.3	Realisierung generischer Funktionen . . . . .	208
10.4	Flavors: Ein neues Konzept für generische Funktionen und Nachrichtenaustausch . . . . .	215
10.4.1	Flavors als Mittel zur Realisierung abstrakter Datentypen . . . . .	216
10.4.2	Flavors als Mittel zur Realisierung generischer Funktionen . . . . .	218
10.4.3	Implementation von Flavors Methoden und Instanzen . . . . .	221
10.5	Grundstrukturen und generische Operationen in LISP-Systemen . . . . .	226
10.6	Aspekte der datengesteuerten Programmierung . . . . .	228
10.7	Literaturhinweise . . . . .	231
10.8	Übungen . . . . .	231
<b>11</b>	<b>Regel-orientierte Programmierung . . . . .</b>	<b>232</b>
11.1	Programmierstile und Informationsverarbeitungsmodelle . . . . .	232
11.2	Grundbegriffe des Mustervergleichs . . . . .	235
11.3	Mustervergleich mit strukturierten Daten . . . . .	241
11.4	Unifikation . . . . .	246
11.5	Einfache Produktionensysteme . . . . .	251
11.6	Verallgemeinerte Produktionensysteme . . . . .	270
11.7	Literaturhinweise . . . . .	280
11.8	Übungen . . . . .	281
<b>12</b>	<b>Verarbeitung von LISP in LISP . . . . .</b>	<b>282</b>
12.1	Interpretation von LISP . . . . .	282
12.2	Realisierung von Umgebungen — Bindungsstrategien . . . . .	287

12.2.1	Realisierung inkrementeller „deep-access“- Umgebungen durch Assoziationslisten . . . . .	288
12.2.2	Realisierung blockförmiger „deep-access“- Umgebungen durch Listenstrukturen . . . . .	289
12.2.3	Realisierung der „shallow-access“-Bindung . . . . .	290
12.3	LISP-Interpreter für verschiedene Bindungsstrategien . . . . .	293
12.3.1	Interpreter für einfache „deep-access“- Bindungsstrategie . . . . .	293
12.3.2	Interpreter für blockförmige „deep-access“- Bindungsstrategie . . . . .	296
12.3.3	Interpreter für „shallow-access“-Bindung . . . . .	296
12.4	Prinzipien der Compilation von LISP . . . . .	298
12.5	Elemente der Assemblersprache in LISP und ihre Verwendung bei der Compilation . . . . .	304
12.6	Literaturhinweise . . . . .	313
12.7	Übungen . . . . .	313

## **Anhänge**

13	Einige Bemerkungen über Programmiersysteme für LISP . . . . .	317
13.1	Was ist ein Programmiersystem . . . . .	317
13.2	Die hauptsächlichen Module eines Programmier- systems . . . . .	319
14	Funktionenverzeichnis . . . . .	327
15	Übersicht über die Abweichung der Funktions- definition in anderen LISP-Dialekten . . . . .	336
16	Literaturverzeichnis . . . . .	346
17	Namen- und Sachverzeichnis . . . . .	351