

Programmieren in COMMON LISP

von

Prof. Dr. Otto Mayer

Universität Kaiserslautern

Wissenschaftsverlag

Mannheim/Wien/Zürich

INHALTSVERZEICHNIS

| | | | |
|----------------|----------|--|-----------|
| KAPITEL | 0 | EINLEITUNG | 13 |
| | 0.1 | Entwicklung und Bedeutung von LISP | 13 |
| | 0.2 | Ziele und Voraussetzungen | 14 |
| | 0.3 | Aufbau und Inhalt | 14 |
| | | | |
| TEIL I | | Grundlagen | 19 |
| | | | |
| KAPITEL | 1 | LISTEN FÜR DATEN UND PROGRAMME | 21 |
| | 1.1 | Erster Versuch am LISP-System | 21 |
| | 1.2 | Atome und Listen | 24 |
| | 1.3 | Zur Darstellung von Listen im Speicher | 25 |
| | 1.4 | Zusammenfassung | 29 |
| | | | |
| KAPITEL | 2 | WERTZUWEISUNG UND WERTERMITTLUNG | 31 |
| | 2.1 | Wertzuweisung und Seiteneffekt | 31 |
| | 2.2 | Quotieren als Schutz gegen Ausweitung | 35 |
| | 2.3 | Zur Auswertung von LISP-Programmen | 36 |
| | 2.4 | Der Interpretierer als Systemfunktion | 38 |
| | 2.5 | Übersetzung von LISP | 41 |
| | 2.6 | Zusammenfassung | 42 |
| | | | |
| KAPITEL | 3 | AUFBAU UND ZERLEGUNG VON LISTEN | 43 |
| | 3.1 | Aufbau von Listen mit CONS | 43 |
| | 3.2 | Zerlegung von Listen mit CAR und CDR | 45 |
| | 3.3 | Mehr zu CONS, CAR und CDR | 49 |
| | 3.4 | Zusammenbau von Listen mit APPEND und LIST | 51 |
| | 3.5 | LIST», eine Modifikation von LIST | 53 |
| | 3.6 | Garbage-Collection | 54 |
| | 3.7 | Zusammenfassung | 55 |

| | | | |
|----------------|----------|---|------------|
| KAPITEL | 4 | PRÄDIKATE | 57 |
| | 4.1 | Drei grundlegende Prädikae: ATOM, CONSP und LJSTP | 57 |
| | 4.2 | Übereinstimmung von LISP-Strukturen | 61 |
| | 4.3 | Vergleich von Zahlen | 63 |
| | 4.4 | Mehr zum Testen auf Übereinstimmung | 66 |
| | 4.5 | Mitgliedschaft in einer Liste, die Funktion MEMBER | 67 |
| | 4.6 | Zusammenfassung | 69 |
| KAPITEL | 5 | BEDINGUNGEN | 71 |
| | 5.1 | Noch einige Prädikate | 71 |
| | 5.2 | Typprüfungen | 74 |
| | 5.3 | Die logischen Verknüpfungen AND, OR und NOT | 75 |
| | 5.4 | Bedingungen, die spezielle Form COND | 78 |
| | 5.5 | Weitere Funktionen zur Manipulation von Listen | 80 |
| | 5.6 | Zusammenfassung | 82 |
| KAPITEL | 6 | FUNKTIONEN | 83 |
| | 6.1 | Definition benannter Funktionen | 83 |
| | 6.2 | Definition unbenannter Funktionen, Lambda-Ausdrücke | 85 |
| | 6.3 | Beispiele zur Definition von Funktionen | 86 |
| | 6.4 | Argumentübergabe beim Funktionsaufruf | 89 |
| | 6.5 | Freie und gebundene Variable | 90 |
| | 6.6 | Zusammenfassung | 93 |
| KAPITEL | 7 | REKURSION | 95 |
| | 7.1 | Grundprinzipien zur Definition rekursiver Funktionen | 95 |
| | 7.2 | Einfache Beispiele für rekursive Funktionen | 97 |
| | 7.3 | Die Türme von Hanoi | 100 |
| | 7.4 | Zusammenfassung | 104 |
| KAPITEL | 8 | ARGUMENTE UND VARIABLE | 107 |
| | 8.1 | Bindung und Umgebung | 107 |
| | 8.2 | Globale Variable | 110 |
| | 8.3 | Einrichtung neuer Variablenbindungen mit LET-Ausdrücken | 113 |
| | 8.4 | Beispiele zur Skopierung | 115 |
| | 8.5 | Funktionale Argumente | 118 |
| | 8.6 | Funktionale Objekte | 121 |
| | 8.7 | Zusammenfassung | 126 |

| | | | |
|----------------|-----------|--|------------|
| KAPITEL | 9 | KONTROLLSTRUKTUREN | 127 |
| | 9.1 | PROG-Formen: PROG, PROG1 undPROGN | 128 |
| | 9.2 | Map-Funktionen | 129 |
| | 9.3 | Allgemeine Iteration | 131 |
| | 9.4 | Rekursion versus Iteration: ein Beispiel | 133 |
| | 9.5 | Zusammenfassung | 135 |
| KAPITEL | 10 | GRUNDLAGEN DER EIN- UND AUSGABE | 137 |
| | 10.1 | Ströme als Kommunikationswege | 137 |
| | 10.2 | Einlesen mit READ | 139 |
| | 10.3 | Ausgabe mit PRINT | 142 |
| | 10.4 | Exotische Symbolbezeichnungen | 145 |
| | 10.5 | Einfache Variationen | 146 |
| | 10.6 | Zusammenfassung | 148 |
| TEIL II | | WEITERER AUSBAU | 149 |
| KAPITEL | 11 | LISTEN | 151 |
| | 11.1 | Listen und Bäume | 151 |
| | 11.2 | Mengen | 153 |
| | 11.3 | Assoziationslisten | 154 |
| | 11.4 | Kopieren von Listen | 157 |
| | 11.5 | Zusammenfassung | 163 |
| KAPITEL | 12 | DESTRUKTIVE FUNKTIONEN | 165 |
| | 12.1 | RPLACA und RPLACD ändern CAR und CDR | 165 |
| | 12.2 | Zirkuläre Listen | 169 |
| | 12.3 | Wie arbeitet eigentlich APPEND? | 170 |
| | 12.4 | NCONC, die destruktive Variante von APPEND | 172 |
| | 12.5 | MAPCAN und MAPCON, zwei destruktive Map-Funktionen | 175 |
| | 12.6 | NREVERSE zur Umkehrung von Listen | 178 |
| | 12.7 | Zusammenfassung | 179 |

| | | | |
|----------------|-----------|---|------------|
| KAPITEL | 13 | FORMATIERTE AUSGABE | 181 |
| | 13.1 | Mehr zur PRINT, PRIN1 und PRINC | 181 |
| | 13.2 | Schönheit überzeugt leichter! | 184 |
| | 13.3 | Die Funktion FORMAT | 184 |
| | 13.4 | Weitere Steueranweisungen für FORMAT | 188 |
| | 13.5 | Fehlermeldungen | 193 |
| | 13.6 | Zusammenfassung | 196 |
| KAPITEL | 14 | PARAMETERSPEZIFIKATIONEN BEIM FUNKTIONS-KONZEPT | 197 |
| | 14.1 | Optionale Argumente | 197 |
| | 14.2 | Beliebig viele Argumente | 200 |
| | 14.3 | Schlüsselwort-Argumente, ein anderes Konzept für optionale Argumente | 202 |
| | 14.4 | Zusammenspiel verschiedener Parameterspezifikationen | 206 |
| | 14.5 | Zusammenfassung | 207 |
| KAPITEL | 15 | MAKROS | 209 |
| | 15.1 | Zur Definition von Makros | 209 |
| | 15.2 | Das Makro IF | 210 |
| | 15.3 | Makros für spezielle Formen | 214 |
| | 15.4 | Backquote | 216 |
| | 15.5 | Seiteneffekte bei Makros | 218 |
| | 15.6 | Zusammenfassung | 219 |
| KAPITEL | 16 | SYMBOLE | 221 |
| | 16.1 | Namen für Symbole | 221 |
| | 16.2 | Zugriff auf Symbole über Packages | 223 |
| | 16.3 | Zur Erzeugung von Symbolen | 225 |
| | 16.4 | Zur Struktur von Symbolen | 225 |
| | 16.5 | DieProperty-Lisle | 227 |
| | 16.6 | Symbole für Werte und Funktionsbeschreibungen | 230 |
| | 16.7 | Zusammenfassung | 232 |

| | | | |
|----------------|-----------|---|------------|
| KAPITEL | 17 | EIN LISP-INTERPRETER | 235 |
| | 17.1 | Ein Interpreter für LISP in LISP | 235 |
| | 17.2 | Die Funktion O-EVAL beschreibt den Interpreter | 236 |
| | 17.2.1 | Zur Auswertung von Symbolen | 237 |
| | 17.2.2 | Zur Auswertung von nichtleeren Listen | 238 |
| | 17.2.2.1 | Spezielle Formen und ihre Auswertung | 239 |
| | 17.2.2.2 | Primitive Operationen und ihre Auswertung | 240 |
| | 17.2.2.3 | Lambda-Ausdrücke und ihre Auswertung | 242 |
| | 17.2.2.4 | Frei definierte Funktionen und ihre Auswertung mit O-APPLY | 243 |
| | 17.2.2.5 | Vollständige Auswertung von nichtleeren Listen | 245 |
| | 17.3 | Zusammenfassung | 245 |
| KAPITEL | 18 | ARRAYS | 247 |
| | 18.1 | Einrichtung von Arrays | 247 |
| | 18.2 | Zugriff auf Array-Komponenten, das allgemeine Zuweisungsmakro SEIT | 250 |
| | 18.3 | Vektoren und Sequenzen | 253 |
| | 18.4 | Strings | 255 |
| | 18.5 | Vergleich von Strings | 256 |
| | 18.6 | Informationen über ARRAYS | 258 |
| | 18.7 | Zusammenfassung | 259 |
| KAPITEL | 19 | WEITERE KONTROLLSTRUKTUREN | 261 |
| | 19.1 | Ausgänge | 261 |
| | 19.1.1 | Statische Ausgänge aus Blöcken | 261 |
| | 19.1.2 | Dynamische Ausgänge | 263 |
| | 19.1.3 | UNWIND-PROTECT zur Sicherung von Seiteneffekten bei THROWS | 264 |
| | 19.2 | Einfache Wiederholungen | 266 |
| | 19.3 | Iteration und Endrekursion | 268 |
| | 19.4 | Zusammenfassung | 271 |

| | | | |
|-----------------------------|-----------|---|------------|
| KAPITEL | 20 | MUSTERVERGLEICH (Pattern-Matching) | 273 |
| | 20.1 | Einfacher Mustervergleich | 274 |
| | 20.2 | Patternvariable | 276 |
| | 20.3 | Eine Funktion MATCH | 277 |
| | 20.4 | Anwendung des Mustervergleichs | 279 |
| | 20.5 | Ein Dialog | 283 |
| | 20.6 | Zusammenfassung | 284 |
| | | | |
| KAPITEL | 21 | DATENTYPEN UND DATENSTRUKTUREN | 285 |
| | 21.1 | Datentypen in COMMON LISP | 285 |
| | 21.1.1 | Die wichtigsten Datentypen | 286 |
| | 21.1.2 | Typbezeichner | 288 |
| | 21.2 | Strukturierte Objekte | 289 |
| | 21.3 | Mehr zum DEFSTRUCT-Konzept | 292 |
| | 21.4 | Vielfach-Werte (multiple values) | 295 |
| | 21.5 | Zur Übernahme von Vielfach-Werten | 297 |
| | 21.6 | Zusammenfassung | 300 |
| | | | |
| KAPITEL | 22 | ZUR ENTWICKLUNG VON PROGRAMMEN | 303 |
| | 22.1 | Strukturiertes Programmieren | 303 |
| | 22.2 | Unterstützung durch Programmentwicklungssysteme | 305 |
| | 22.3 | Die Komponenten eines Programmentwicklungssystems | 307 |
| | 22.4 | DasTRACE-Paket | 316 |
| | 22.5 | Das STEP-Paket | 318 |
| | 22.6 | Zusammenfassung | 320 |
| | | | |
| LITERATURHINWEISE | | | 323 |
| | | | |
| STICHWORTVERZEICHNIS | | | 325 |