

Inhalt

| | |
|---|-----------|
| 1 Grundlagen der Programmierung in MUMPS | 1 |
| 1.1 Einführung | 1 |
| 1.1.1 Programmiersprachen in der historischen Entwicklung — ein Überblick | 1 |
| 1.1.2 Die Entwicklung von MUMPS | 3 |
| 1.1.3 Ein Kaleidoskop der Sprache — MUMPS im Überblick | 4 |
| 1.2 Benutzerbereiche und das Einloggen | 6 |
| 1.3 Einfache syntaktische Regeln | 8 |
| 1.4 Beispiele einfacher Sprachelemente | 10 |
| 1.4.1 Der WRITE-Befehl | 10 |
| 1.4.2 Variablen in MUMPS und der SET-Befehl | 12 |
| 1.4.3 Löschen von Variablen mit dem KILL-Befehl | 15 |
| 1.4.4 Wertzuweisung im Dialog mit dem READ-Befehl | 17 |
| 1.4.5 \$LENGTH als Beispiel einer einfachen Funktion | 20 |
| 1.5 Zahlen und numerische Operationen | 21 |
| 1.6 Zeichenvorrat und Zeichenketten | 26 |
| 1.6.1 Zeichensatz und Stringlitterale | 26 |
| 1.6.2 Der Datentyp in MUMPS und die numerische Interpretation | 28 |
| 1.7 Datum und Zeit — \$HOROLOG | 31 |
| 1.8 Die Syntax einer Befehlszeile | 33 |
| 2 Programme und Programmstrukturen | 35 |
| 2.1 Programme in MUMPS | 35 |
| 2.1.1 Aufbau eines Programms | 35 |
| 2.1.2 Aufruf eines Programms — der DO-Befehl | 37 |
| 2.2 Lokaler und globaler Aufruf | 39 |
| 2.3 Wertübergabe beim Unterprogrammaufruf | 43 |
| 2.4 Benutzerdefinierte Funktionen und spezielle Variablen | 46 |
| 2.5 Sichtbarkeit von lokalen Variablen — der NEW-Befehl | 49 |
| 2.6 Programmverzweigung mit GOTO | 52 |

| | | |
|----------|---|-----------|
| 2.7 | Zeichenketten als Programmzeilen — der XECUTE-Befehl . . | 54 |
| 2.8 | Über das Editieren von Programmen | 56 |
| 3 | Steuerung des Programmflusses | 59 |
| 3.1 | Vergleichsoperatoren und ihre Anwendung | 59 |
| 3.2 | Logische Operatoren und die logische Interpretation | 62 |
| 3.3 | Bedingte Programmausführung — der IF-Befehl | 65 |
| 3.4 | IF, ELSE und \$TEST | 68 |
| 3.5 | Bedingte Ausführung von Befehlen | 72 |
| 3.5.1 | Nachbedingung auf Befehle | 72 |
| 3.5.2 | Die Nachbedingung auf Argumente eines Befehls . . . | 73 |
| 3.6 | Schleifen — der FOR-Befehl | 76 |
| 3.7 | Rekursive Programmierung | 81 |
| 3.8 | Programmblöcke mit der Punktsyntax | 83 |
| 3.9 | Die Auswahlfunktion \$SELECT | 86 |
| 3.10 | Die Indirektion | 88 |
| 3.10.1 | Die Namensindirektion | 88 |
| 3.10.2 | Die Argumentindirektion | 90 |
| 3.11 | Fehlersuche mit BREAK | 92 |
| 4 | Kommunikation mit angeschlossenen Geräten | 97 |
| 4.1 | Bildschirmsteuerung im Rollmodus, \$X und \$Y | 97 |
| 4.2 | Zeichendarstellung | 100 |
| 4.2.1 | Der ASCII-Zeichensatz und Erweiterungen | 100 |
| 4.2.2 | Umwandlung eines Codes in ein Zeichen — \$CHAR . | 101 |
| 4.2.3 | Umwandlung eines Zeichens in seinen Code — \$ASCII | 102 |
| 4.2.4 | Die Sternsyntax | 103 |
| 4.3 | Prinzipien der Bildschirmsteuerung | 106 |
| 4.4 | Methodik der Bildschirmsteuerung | 109 |
| 4.4.1 | Bildschirmsteuerung in der Praxis | 109 |
| 4.4.2 | Prinzipien tabellengesteuerter Masken | 111 |
| 4.5 | Die Benutzung externer Geräte | 115 |
| 4.5.1 | Prinzipien der Kommunikation mit externen Geräten . | 115 |
| 4.5.2 | Die Reservierung von Geräten mit OPEN | 116 |
| 4.5.3 | Auswahl eines Geräts und Beenden der Kommunika- tion — USE, \$IO und CLOSE | 118 |

| | | |
|----------|---|------------|
| 4.6 | Der Sprachstandard und die Portabilität von Programmen . . | 120 |
| 5 | Datenhaltung und Datenmanagement | 123 |
| 5.1 | Indizierte Variablen | 123 |
| 5.1.1 | Zahlen als Indizes | 123 |
| 5.1.2 | Löschen und Wertübergabe von (Teil-) Bäumen | 126 |
| 5.1.3 | Zeichenketten als Indizes | 126 |
| 5.2 | Datenhaltung in MUMPS — Globale Variablen | 130 |
| 5.2.1 | Grundprinzipien globaler Variablen | 130 |
| 5.2.2 | Lesen, Schreiben und Löschen von globalen Variablen | 132 |
| 5.2.3 | Dienstprogramme für Globals | 134 |
| 5.3 | Sortierung und \$ORDER | 135 |
| 5.3.1 | Die Sortierreihenfolge | 135 |
| 5.3.2 | Das Navigieren in Datenstrukturen — \$ORDER . . . | 137 |
| 5.3.3 | Analyse der Baumstruktur — \$QUERY | 140 |
| 5.3.4 | Indirektion auf Indizes | 142 |
| 5.4 | Existenz von Datensätzen — \$DATA und \$GET | 145 |
| 5.5 | Der implizite Bezug auf einen Global | 150 |
| 5.5.1 | Die Naked Reference | 150 |
| 5.5.2 | Die Tücken des SET-Befehls | 152 |
| 5.6 | Design von Datenstrukturen — Überlegungen und Beispiele . | 154 |
| 5.6.1 | Einführung und Beschreibung des Modells | 154 |
| 5.6.2 | Abbildung in Globalstrukturen | 156 |
| 5.6.3 | Aspekte eindeutiger Personenidentifizierung | 159 |
| 5.6.4 | Die Einbeziehung leistungsbezogener Daten | 162 |
| 6 | Analyse und Synthese von Zeichenketten | 167 |
| 6.1 | Allgemeines Modell und Anforderungen | 167 |
| 6.2 | Allgemeine Verarbeitung | 169 |
| 6.2.1 | Länge und Extraktion — \$LENGTH und \$EXTRACT | 169 |
| 6.2.2 | Suchen von Teilzeichenketten — \$FIND | 170 |
| 6.2.3 | Ersetzen von Zeichen — \$TRANSLATE | 172 |
| 6.3 | Spezielle Feldverarbeitung | 175 |
| 6.3.1 | Anzahl und Extraktion von Feldern — \$LENGTH und \$PIECE | 175 |
| 6.3.2 | Zuweisung von Feldern — SET \$PIECE | 179 |

| | | |
|----------|--|------------|
| 6.4 | Datenprüfung mit Textoperatoren | 183 |
| 6.4.1 | Fehlerarten bei der Eingabe | 183 |
| 6.4.2 | Der Mustervergleich | 184 |
| 6.4.3 | Elemente eines Datumprüfprogramms | 188 |
| 6.4.4 | Der Folgt- und Enthält-Operator | 190 |
| 6.5 | Formatierung der Ausgabe | 193 |
| 6.5.1 | Rechtsbündige Ausgabe — \$JUSTIFY | 193 |
| 6.5.2 | Spezielle Formatierung mit \$FNUMBER | 195 |
| 6.6 | Ein einfacher Menütreiber mit \$TEXT | 197 |
| 7 | Mehrbenutzersysteme | 201 |
| 7.1 | Begriffsklärung | 201 |
| 7.2 | Sperren im Mehrbenutzerbetrieb — der LOCK-Befehl | 203 |
| 7.2.1 | Einführung in die Problematik | 203 |
| 7.2.2 | Der LOCK-Befehl | 204 |
| 7.2.3 | Zusätzliche Sperren | 206 |
| 7.3 | Hintergrundprozesse | 208 |
| 7.3.1 | Der JOB-Befehl | 208 |
| 7.3.2 | Wertübergabe an Hintergrundprozesse | 209 |
| 7.3.3 | Verzögerung der Ausführung mit HANG | 210 |
| A | Übersicht über die Sprache | 213 |
| A.1 | Darstellungsweise | 213 |
| A.2 | Struktur eines Programms | 214 |
| A.3 | Häufig benötigte nicht-terminalsymbole | 215 |
| A.3.1 | Namen | 215 |
| A.3.2 | Ausdrücke | 216 |
| A.3.3 | Zeilenreferenzen und Programmaufruf | 217 |
| A.3.4 | Ein / Ausgabe | 218 |
| A.4 | Übersicht über Befehle | 219 |
| A.4.1 | BREAK | 219 |
| A.4.2 | CLOSE | 219 |
| A.4.3 | DO | 220 |
| A.4.4 | ELSE | 221 |
| A.4.5 | FOR | 221 |
| A.4.6 | GOTO | 222 |

| | | |
|--------|---------------------------|-----|
| A.4.7 | HALT | 222 |
| A.4.8 | HANG | 223 |
| A.4.9 | IF | 223 |
| A.4.10 | JOB | 224 |
| A.4.11 | KILL | 224 |
| A.4.12 | LOCK | 225 |
| A.4.13 | NEW | 226 |
| A.4.14 | OPEN | 226 |
| A.4.15 | QUIT | 227 |
| A.4.16 | READ | 227 |
| A.4.17 | SET | 228 |
| A.4.18 | USE | 228 |
| A.4.19 | VIEW | 229 |
| A.4.20 | WRITE | 229 |
| A.4.21 | XECUTE | 229 |
| A.4.22 | Z-Befehle | 230 |
| A.5 | Übersicht über Funktionen | 230 |
| A.5.1 | \$ASCII | 230 |
| A.5.2 | \$CHAR | 230 |
| A.5.3 | \$DATA | 231 |
| A.5.4 | \$EXTRACT | 231 |
| A.5.5 | \$FIND | 232 |
| A.5.6 | \$FNUMBER | 232 |
| A.5.7 | \$GET | 233 |
| A.5.8 | \$JUSTIFY | 233 |
| A.5.9 | \$LENGTH | 234 |
| A.5.10 | \$NEXT | 234 |
| A.5.11 | \$ORDER | 234 |
| A.5.12 | \$PIECE | 235 |
| A.5.13 | \$QUERY | 235 |
| A.5.14 | \$RANDOM | 236 |
| A.5.15 | \$SELECT | 236 |
| A.5.16 | \$TEXT | 237 |
| A.5.17 | \$TRANSLATE | 237 |
| A.5.18 | \$VIEW | 238 |

| | |
|---|------------|
| A.5.19 \$Z-Funktionen | 238 |
| A.6 Übersicht über Systemvariablen | 238 |
| A.6.1 \$HOROLOG | 238 |
| A.6.2 \$IO | 239 |
| A.6.3 \$JOB | 239 |
| A.6.4 \$STORAGE | 239 |
| A.6.5 \$TEST | 239 |
| A.6.6 \$X | 240 |
| A.6.7 \$Y | 240 |
| A.6.8 \$Z-Variablen | 240 |
| A.7 Übersicht über Operatoren | 241 |
| A.7.1 Allgemeines | 241 |
| A.7.2 Übersicht über die einzelnen Operatoren | 242 |
| B Überblick über den Standard von 1989 | 247 |
| C Verzeichnis der ASCII-Zeichen | 251 |
| Literaturverzeichnis | 253 |
| Sachverzeichnis | 256 |