

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>v</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Eigenschaften funktionaler Sprachen . . . . .	2
1.2 Überblick über Miranda . . . . .	6
1.3 Warum nicht LISP oder Scheme? . . . . .	12
1.4 Literaturhinweise . . . . .	13
<b>2 Vordefinierte Typen</b>	<b>15</b>
2.1 Der Typ num . . . . .	15
2.2 Der Typ bool . . . . .	16
2.3 Der Typ char . . . . .	18
2.4 Tupeltypen . . . . .	18
2.5 Listentypen . . . . .	19
2.6 Funktionale Typen . . . . .	23
2.7 Miranda-System . . . . .	28
2.8 Syntax . . . . .	33
<b>3 Definitionen</b>	<b>39</b>
3.1 Skalare Definitionen . . . . .	40
3.1.1 Die Offside-Regel . . . . .	40
3.1.2 Bewachte Ausdrücke . . . . .	41
3.1.3 Pattern-Matching . . . . .	44
3.1.4 DIY-Infix-Operatoren . . . . .	49
3.2 Konforme Definitionen . . . . .	51
3.3 Lokale Definitionen . . . . .	52
3.4 Syntax . . . . .	55
<b>4 Typsystem</b>	<b>63</b>
4.1 Parametrisierter Typpolymorphismus . . . . .	67

4.2	Typinferenz . . . . .	70
4.3	Wertespezifikationen . . . . .	79
4.4	Typsynonyme . . . . .	80
4.5	Die generische Funktion <code>show</code> . . . . .	82
4.6	Literaturhinweise . . . . .	83
4.7	Syntax . . . . .	84
<b>5</b>	<b>Funktionen höherer Ordnung</b>	<b>89</b>
5.1	Partiell parametrisierte Funktionen . . . . .	91
5.2	Operator-Sections . . . . .	92
5.3	Generische Funktionen . . . . .	93
5.4	Funktionen als Datenstrukturen . . . . .	98
5.5	Implementierung von Rekursionsmustern . . . . .	100
5.5.1	<code>map</code> und <code>map2</code> . . . . .	100
5.5.2	<code>filter</code> , <code>dropwhile</code> und <code>takewhile</code> . . . . .	105
5.5.3	<code>foldr</code> und <code>foldl</code> . . . . .	107
5.5.4	Kontrollstrukturen . . . . .	113
5.6	Literaturhinweise . . . . .	115
5.7	Syntax . . . . .	116
<b>6</b>	<b>List-Comprehensions</b>	<b>123</b>
6.1	Einfache List-Comprehensions . . . . .	123
6.2	Iterative Generatoren . . . . .	130
6.3	Diagonalisierende List-Comprehensions . . . . .	132
6.4	Syntax . . . . .	133
<b>7</b>	<b>Benutzerdefinierte Typen</b>	<b>139</b>
7.1	Algebraische Datentypen . . . . .	139
7.1.1	Aufzählungstypen . . . . .	140
7.1.2	Disjunkte Vereinigungen . . . . .	143
7.1.3	Rekursive Datentypen . . . . .	147
7.1.4	DIY-Infix-Operatoren . . . . .	163
7.2	Abstrakte Datentypen . . . . .	164
7.3	Platzhaltertypen . . . . .	173

7.4	Literaturhinweise . . . . .	174
7.5	Syntax . . . . .	175
<b>8</b>	<b>Lazy Evaluation</b>	<b>183</b>
8.1	Eager Evaluation versus Lazy Evaluation . . . . .	183
8.2	Nichtstrikte Funktionen . . . . .	188
8.3	Backtracking-Probleme . . . . .	189
8.3.1	Das $n$ -Damen-Problem . . . . .	189
8.3.2	Das Springer-Problem . . . . .	196
8.4	Kombinator-Parsing . . . . .	198
8.5	Von Prolog zu Miranda . . . . .	210
8.6	Unendliche Datenstrukturen . . . . .	218
8.6.1	Simulation von synchronen Schaltwerken . . . . .	222
8.6.2	Netzwerke kommunizierender Prozesse . . . . .	227
8.6.3	Primzahlen . . . . .	231
8.6.4	Memotabellen . . . . .	232
8.7	Programmierung mit Unbekannten . . . . .	238
8.8	Steuerung der Auswertung . . . . .	241
8.9	Literaturhinweise . . . . .	243
<b>9</b>	<b>Programmierung im Großen</b>	<b>253</b>
9.1	<u>%include</u> und <u>%export</u> -Direktive . . . . .	254
9.1.1	<u>%include</u> -Direktive . . . . .	254
9.1.2	<u>%export</u> -Direktive . . . . .	256
9.2	Parametrisierte Module . . . . .	260
9.2.1	Verwaltung von Binärbäumen . . . . .	262
9.2.2	Prioritäts-Parser . . . . .	265
9.3	Software-Engineering . . . . .	271
9.4	Programmierrichtlinien für Miranda . . . . .	280
9.5	Literaturhinweise . . . . .	282
9.6	Syntax . . . . .	283
<b>10</b>	<b>Interaktive Programme</b>	<b>285</b>
10.1	Eingabe . . . . .	287

10.2	Interpretierte Eingabe . . . . .	288
10.3	Ausgabe . . . . .	290
10.4	Einbindung in UNIX . . . . .	292
10.5	Fortsetzungsbasierte Ein- und Ausgabe . . . . .	293
10.6	Literaturhinweise . . . . .	303
<b>A</b>	<b>Kleine Projekte</b>	<b>309</b>
A.1	Taschenrechner . . . . .	309
A.2	Datenbank . . . . .	312
A.3	KWIC-Index . . . . .	315
A.4	Textformatierung . . . . .	318
<b>B</b>	<b>Blaise-Compiler</b>	<b>323</b>
B.1	Ziel des Projektes . . . . .	323
B.2	Projektbeschreibung . . . . .	323
B.2.1	Definition von Blaise . . . . .	323
B.2.2	Definition der Zielmaschine . . . . .	327
B.2.3	Programmierungsumgebung . . . . .	329
B.3	Projektorganisation . . . . .	330
B.3.1	Gruppe 1: Lexikalische Analyse und Symboltabelle . . . . .	331
B.3.2	Gruppe 2: Syntaktische Analyse . . . . .	332
B.3.3	Gruppe 3: Codeerzeugung . . . . .	332
B.3.4	Gruppe 4: Interpreter und Programmierungsumgebung . . . . .	333
	<b>Literatur</b>	<b>335</b>