

# Inhaltsverzeichnis

<b>1</b>	<b>Was ist Informatik?</b>	<b>5</b>
<b>2</b>	<b>Elemente des Programmierens</b>	<b>9</b>
2.1	Handwerkszeug für das Programmieren . . . . .	9
2.2	Bausteine für Programme . . . . .	11
2.3	Sorten und Verträge . . . . .	15
2.4	Testfälle . . . . .	17
2.5	Probleme und Teilprobleme . . . . .	18
2.6	Das Substitutionsmodell . . . . .	20
2.7	Lexikalische Bindung . . . . .	22
<b>3</b>	<b>Fallunterscheidungen und Verzweigungen</b>	<b>27</b>
3.1	Fallunterscheidungen und boolesche Ausdrücke . . . . .	27
3.2	Binäre Verzweigungen . . . . .	30
3.3	Weitere boolesche Operatoren . . . . .	31
<b>4</b>	<b>Zusammengesetzte und gemischte Daten</b>	<b>37</b>
4.1	Schokokekse im Computer . . . . .	37
4.2	Record-Definitionen . . . . .	40
4.3	Prozeduren, die Records konsumieren . . . . .	42
4.4	Prozeduren, die Records herstellen . . . . .	44
4.5	Ein erweitertes Beispiel für zusammengesetzte Daten . . . . .	45
4.6	Gemischte Daten . . . . .	49
<b>5</b>	<b>Induktive Definitionen</b>	<b>55</b>
5.1	Natürliche Zahlen . . . . .	55
5.2	Endliche Folgen . . . . .	58
5.3	Terme . . . . .	59
5.4	Darstellung von Termen . . . . .	61
5.5	Strukturelle Induktion . . . . .	62
5.6	Algebren . . . . .	63
<b>6</b>	<b>Rekursion</b>	<b>67</b>
6.1	Funktionen auf endlichen Folgen . . . . .	67
6.2	Listen in Scheme . . . . .	69
6.3	Rekursion über Zahlen . . . . .	73

6.4	Rekursion ohne Ende . . . . .	77
6.5	Induktionsbeweise über rekursive Funktionen . . . . .	78
6.6	Rekursive Scheme-Programme beweisen . . . . .	79
6.7	Endrekursion und Iteration . . . . .	81
6.8	Invarianten . . . . .	84
<b>7</b>	<b>Praktische Programme mit Listen</b>	<b>87</b>
7.1	Eingebaute Listen . . . . .	87
7.2	Mehr Prozeduren auf Listen . . . . .	87
7.3	Die Türme von Hanoi . . . . .	90
7.4	Zwischenergebnisse benennen . . . . .	97
7.5	Audio-Kassetten optimal bespielen . . . . .	99
<b>8</b>	<b>Higher-Order-Programmierung</b>	<b>107</b>
8.1	Higher-Order-Prozeduren auf Listen . . . . .	107
8.2	Listen zusammenfalten . . . . .	111
8.3	Anonyme Prozeduren . . . . .	113
8.4	Prozedurfabriken . . . . .	115
8.5	Der Schönfinkel-Isomorphismus . . . . .	116
<b>9</b>	<b>Zeitabhängige Modelle</b>	<b>121</b>
9.1	Das Teachpack <code>image.ss</code> . . . . .	121
9.2	Modelle und Ansichten . . . . .	124
9.3	Bewegung und Zustand . . . . .	126
9.4	Andere Welten . . . . .	128
<b>10</b>	<b>Abstrakte Datentypen</b>	<b>133</b>
10.1	ADTs, Signaturen und Datentypen . . . . .	133
10.2	Zähler . . . . .	135
10.3	Gleichungsdefinierte Datentypen . . . . .	137
10.4	Konstruktoren und wohlgeformte ADTs . . . . .	140
10.5	Listen . . . . .	142
10.6	ADTs und Implementierung . . . . .	143
10.7	Suchen in endlichen Mengen . . . . .	147
10.8	Parametrisierte Implementierung . . . . .	150
<b>11</b>	<b>Binäre Bäume</b>	<b>153</b>
11.1	Binärbäume . . . . .	153
11.2	Suchbäume . . . . .	158
11.3	Huffman-Bäume . . . . .	163
<b>12</b>	<b>Zuweisungen und Zustand</b>	<b>175</b>
12.1	Zustandsvariablen . . . . .	175
12.2	Zustand kapseln . . . . .	178
12.3	Erweitertes Beispiel . . . . .	181

12.4	Zuweisungen und das Substitutionsmodell . . . . .	186
12.5	Das Umgebungsmodell für die Programmauswertung . . . . .	189
12.6	Gekapselter Zustand mit Closures . . . . .	194
12.7	Sharing und Identität . . . . .	197
12.8	Zeiger . . . . .	199
<b>13</b>	<b>Objektorientiertes Programmieren</b>	<b>203</b>
13.1	Message-Passing-Style . . . . .	203
13.2	OOP = MPS + Zustand + self + Vererbung . . . . .	205
13.3	Vererbung und self . . . . .	212
13.4	Mehrfachvererbung . . . . .	216
13.5	Abstraktion über Klassen . . . . .	219
<b>14</b>	<b>Logische Kalküle</b>	<b>225</b>
14.1	Wahrheit und Beweisbarkeit . . . . .	225
14.2	Ein Kalkül für die Aussagenlogik . . . . .	226
14.3	Modelle für die Aussagenlogik . . . . .	228
14.4	Korrektheit, Konsistenz und Vollständigkeit . . . . .	229
14.5	Der Reduktionskalkül $RC_1$ . . . . .	230
<b>15</b>	<b>Der <math>\lambda</math>-Kalkül</b>	<b>237</b>
15.1	Sprache und Reduktionssemantik . . . . .	237
15.2	Normalformen . . . . .	242
15.3	Der $\lambda$ -Kalkül als Programmiersprache . . . . .	243
15.4	Auswertungsstrategien . . . . .	248
15.5	Die Auswertungsstrategie von Scheme . . . . .	250
15.6	Übungsaufgaben . . . . .	250
<b>16</b>	<b>Interpretation von Scheme</b>	<b>253</b>
16.1	Quote und Symbole . . . . .	253
16.2	Mini-Scheme: eine Untermenge von Scheme . . . . .	257
16.3	Repräsentation von Werten . . . . .	265
16.4	Repräsentation von Umgebungen und Frames . . . . .	266
16.5	Auswertung und Anwendung . . . . .	273
16.6	Programme ausführen . . . . .	280
<b>A</b>	<b>Geschichte der Informatik</b>	<b>287</b>
A.1	Automatisierung des Denkens . . . . .	287
A.2	Programmierbare Maschinen . . . . .	288
A.3	Programme, Berechnungen und Algorithmen . . . . .	290
<b>B</b>	<b>Mathematische Grundlagen</b>	<b>293</b>
B.1	Aussagenlogik . . . . .	293
B.2	Mengen . . . . .	294
B.3	Prädikatenlogik . . . . .	297

B.4	Multimengen . . . . .	298
B.5	Relationen und Abbildungen . . . . .	298
B.6	Ordnungen . . . . .	300
<b>C</b>	<b>Mantras zur Programmierung</b>	<b>303</b>
<b>D</b>	<b>Konstruktionsanleitungen</b>	<b>305</b>