

## Inhalt

<b>1 Einleitung</b>	<b>1</b>
<b>2 Einführung</b>	<b>5</b>
2.1 Was ist Programmierung	5
2.2 Eine systemorientierte Sichtweise der Programmierung	8
2.2.1 Analyse der Modellbildung	8
2.2.1.1 Aspekte von Phänomenen	9
2.2.1.2 Aspekte von Begriffen	9
2.2.1.3 Aspekte von Abstraktion	10
2.2.2 Resümee	12
2.2.3 Analyse der Modelle	12
2.3 Andere Sichtweisen von Programmierung	14
2.4 Maschinen, Verarbeitungsmodelle, Programmiersprachen, Semantik	15
2.5 Klassifizierung und Einordnung von Verarbeitungsmodellen	17
2.6 Zur Favorisierung der objektorientierten Programmierung	20
2.7 Was ist diskrete Simulation	21
<b>3 Sprachkonzepte</b>	<b>25</b>
3.1 Sprachkonzepte sequentieller Programmierung	26
3.1.1 Das Wert- und Objektkonzept	26
3.1.1.1 Werte als Sprachelemente	27
3.1.1.2 Objekte als Sprachelemente	27
3.1.1.3 Objektarten	28
3.1.1.4 Objekte und die algorithmische Basis	30
3.1.1.5 Zur Bedeutung des Objektkonzepts für den Entwurf für Programmiersprachen	30
3.1.1.6 Zum Begriff der Konstanten und Variablen	31
3.1.2 Das Typkonzept	32
3.1.2.1 Ein allgemeiner Typbegriff	32
3.1.2.2 Zum Konzept der Datentypen	33
3.1.2.3 Zum Begriff der Datenstruktur	35
3.1.2.4 Zur Bedeutung des Datentypkonzepts	36
3.1.3 Kontrollstrukturen und Anweisungen	38
3.1.4 Das Blockkonzept	40
3.1.4.1 Der Block als Sprachkonstrukt	40
3.1.4.2 Zur Bedeutung des Blockkonzepts	42
3.1.5 Das Prozedur/Funktionskonzept	44
3.1.5.1 Prozeduren/Funktionen als Sprachkonstrukte	44

5.5 Beta.....	144
5.5.1 Entstehung und Einordnung.....	144
5.5.2 Programmierung in Beta.....	144
5.5.3 Pattern als Abstraktionsmechanismus .....	145
5.5.3.1 Objektdeskriptoren und Pattern .....	146
5.5.3.2 Statische Deklarationen .....	147
5.5.3.3 Objektattribute.....	147
5.5.3.4 Patternattribute.....	148
5.5.3.5 Parameter.....	148
5.5.3.6 Auswertungen und Parameterübergabe .....	148
5.5.3.7 Kontrollstrukturen.....	149
5.5.4 Generierungsweisen.....	150
5.5.5 Datenabstraktion und algorithmische Abstraktion in Beta.....	151
5.5.5.1 Datenobjekte.....	153
5.5.5.2 Prozeduren und Funktionen.....	153
5.5.6 Blockstrukturen in Beta.....	154
5.5.7 Vererbung in Beta.....	154
5.5.7.1 Vererbung von Attributen.....	155
5.5.7.2 Vererbung von Aktionen.....	155
5.5.7.3 Typüberprüfung und Qualifizierung.....	157
5.5.7.4 Virtuelle Pattern.....	158
5.5.7.4.1 Einfache Spezialisierung.....	158
5.5.7.4.2 Abkürzungsmechanismus zur Verwendung virtueller Pattern.....	159
5.5.7.4.3 Polymorphismen: Typ- und Prozedurparameter .....	159
5.5.8 Nichtsequentialität in Beta .....	164
5.5.8.1 Arten von Nichtsequentialität .....	165
5.5.8.2 Kommunikation .....	166
5.5.8.3 Alternation.....	168
5.5.8.4 Nebenläufigkeit.....	170
5.5.8.5 Verbundsysteme.....	170
5.5.8.5.1 Ausführung globaler Objekte .....	171
5.5.8.5.2 Kommunikation mit internen Systemen .....	172
5.5.8.5.3 Kommunikation unter Benutzung globaler Objekte.....	173
5.5.9 Modularisierung in Beta.....	173
5.6 Warum kein GPSS .....	175
<b>6 Möglichkeiten diskreter Simulation.....</b>	<b>177</b>
6.1 Diskrete Simulation in Simula-67.....	178
6.1.1 Allgemeines zur Simulation in Simula-67 .....	178
6.1.2 Konzeption des Simula-67-Simulators .....	179

6.2 Diskrete Simulation in Modula-2 .....	180
6.2.1 Allgemeines zur Simulation in Modula-2.....	180
6.2.2 Konzeption eines konkreten Simulators in Modula-2.....	181
6.3 Diskrete Simulation in Pascal .....	183
6.4 Diskrete Simulation in Smalltalk-80.....	184
6.4.1 Allgemeines zur Simulation in Smalltalk-80.....	184
6.4.2 Konzeption des Smalltalk-80-Simulators.....	185
6.5 Diskrete Simulation in Beta .....	188
<b>7 Bewertung der Sprachen.....</b>	<b>191</b>
7.1 Methodik der Bewertung .....	193
7.1.1 Dimensionen der Bewertung.....	193
7.1.2 Bewertungskriterien.....	196
7.1.3 Zur Dimension der pragmatischen und formalen Vorgaben.....	199
7.2 Vergleichende Bewertung .....	201
7.2.1 Pragmatische Bewertung aus der Sicht des Programmierers .....	201
7.2.1.1 Konzeptioneller Abstand .....	201
7.2.1.2 Strukturierungsmöglichkeiten .....	202
7.2.1.3 Benutzungssicherheit der Sprache.....	203
7.2.1.4 Modularisierbarkeit.....	204
7.2.1.5 Erweiterbarkeit und Flexibilität.....	205
7.2.1.6 Ausdrucksmittel für Nichtsequentialität.....	206
7.2.1.7 Simulationsspezifische Datenstrukturen .....	207
7.2.1.8 Objekte als Simulationseinheiten .....	208
7.2.1.9 Dynamische Speicherverwaltung .....	208
7.2.1.10 Polymorphismen .....	209
7.2.1.11 Konkrete Simulationsumgebung.....	210
7.2.1.12 Unterstützung der Validierbarkeit .....	211
7.2.1.13 Effizienz .....	212
7.2.1.14 Softwaretechnische Kriterien .....	212
7.2.1.15 Anforderungen an die Programmierungsumgebung.....	212
7.2.2 Pragmatische Bewertung aus der Sicht des Wissenschaftlers .....	213
7.2.2.1 Implementierbarkeit .....	213
7.2.2.2 Semantische Fundierung .....	213
7.2.2.3 Methodenunterstützung.....	214
7.2.2.4 Konsistenz, Gültigkeit und Angemessenheit der Konzepte .....	215

<b>8 Schlußbemerkungen zu den Sprachen .....</b>	<b>217</b>
8.1 Schlußbemerkungen zu Simula-67 .....	217
8.2 Schlußbemerkungen zu Modula-2 .....	218
8.3 Schlußbemerkungen zu Smalltalk-80 .....	219
8.4 Schlußbemerkungen zu Beta .....	221
 <b>9 Resümee und Ausblick .....</b>	 <b>223</b>
 <b>Anhänge</b>	
<b>A Konkrete Simulatoren .....</b>	<b>227</b>
A1 Ein Simulator in Simula-67 .....	228
A1.1 Standarddatenstrukturen für die Simulation .....	228
A1.1.1 Listen und Schlangen: die Klasse <i>SIMSET</i> .....	228
A1.1.2 Die Klasse <i>LINKAGE</i> .....	229
A1.1.3 Die Klasse <i>LINK</i> .....	229
A1.1.4 Die Klasse <i>HEAD</i> .....	230
A1.2 Ein Prozeßkonzept: die Klasse <i>SIMULATION</i> .....	230
A1.2.1 Struktur der Klasse <i>PROCESS</i> .....	231
A1.2.2 Zugriffsprozeduren und Simulationsanweisungen der Klasse <i>SIMULATION</i> .....	232
A1.2.3 Zusammenfassung der Simulation in Simula-67 .....	233
A2 Ein Simulator in Modula-2 .....	235
A2.1 Systemgliederung (Modularisierung) .....	235
A2.2 Modul <i>Simulation</i> .....	235
A2.2.1 Definitionen des Moduls <i>Simulation</i> .....	236
A2.3 Modul <i>ProcessQueues</i> .....	239
A2.3.1 Definitionen des Moduls <i>ProcessQueues</i> .....	239
A2.4 Modul <i>StorageManager</i> .....	241
A2.4.1 Definitionen des Moduls <i>StorageManager</i> .....	242
A2.5 Modul <i>Random</i> .....	243
A2.5.1 Definitionen des Moduls <i>Random</i> .....	243
A2.6 Modul <i>SimIO</i> und <i>Errors</i> .....	244
A2.7 Zusammenfassung der Abweichungen des Simulationskonzepts gegenüber Simula-67 .....	245
A3 Ein Simulator in Smalltalk-80 .....	247
A3.1 Die Klasse <i>SimulationObject</i> .....	247
A3.2 Die Klasse <i>Simulation</i> .....	250
A3.3 Weitere Simulationsunterstützung .....	253

<b>B Beispielimplementierungen einer Autowaschanlage</b> .....	<b>255</b>
B1 Simulation einer Autowaschanlage in Simula-67.....	256
B1.1 Die Implementierung: Die Klasse <i>WashingCars</i> .....	256
B2 Simulation einer Autowaschanlage in Modula-2 .....	258
B2.1 Dokumentation der Implementierung .....	258
B2.1.1 Globale Variablen und Prozeduren.....	258
B2.1.2 Die Koroutinen als aktive Simulationseinheiten .....	259
B2.1.3 Der Hauptprozeß.....	259
B2.2 Die Implementierung: Das Programmodul <i>CarWash</i> .....	260
B2.3 Ablaufprotokoll.....	264
B3 Simulation einer Autowaschanlage in Smalltalk-80 .....	265
B3.1 Die Simulationsklasse <i>CarWash</i> .....	265
B3.2 Die Simulationsobjektklasse <i>Washer</i> .....	266
B3.3 Die Simulationsobjektklasse <i>Wash</i> .....	267
B3.4 Die Simulationsobjektklasse <i>WashAndWax</i> .....	268
<b>C Begriffsklärungen</b> .....	<b>269</b>
C1 Arbeit und Arbeitsprozesse .....	270
C2 Abstrakte und konkrete Datentypen.....	272
C3 Polymorphe und monomorphe Datentypen.....	274
C4 Innere Festigkeit von Modulen.....	278
<b>Literaturverzeichnis</b> .....	<b>279</b>
<b>Indexverzeichnis</b> .....	<b>289</b>