

Inhaltsverzeichnis

1. Objektorientierung: Ein Einstieg	1
1.1 Objektorientierung: Konzepte und Stärken	1
1.1.1 Gedankliche Konzepte der Objektorientierung	2
1.1.2 Objektorientierung als Antwort auf softwaretechnische Anforderungen	4
1.2 Paradigmen der Programmierung	8
1.2.1 Prozedurale Programmierung	10
1.2.2 Deklarative Programmierung	12
1.2.3 Objektorientierte Programmierung: Das Grundmodell	15
1.2.4 Bemerkungen zum Paradigmenvergleich	21
1.3 Programmiersprachlicher Hintergrund	21
1.3.1 Grundlegende Sprachmittel am Beispiel von Java	22
Objekte und Werte: Eine begriffliche Abgrenzung	22
Objektreferenzen, Werte, Typen und Variablen	23
Anweisungen, Blöcke und deren Ausführung	28
1.3.2 Objektorientierte Programmierung mit prozeduralen Sprachen	35
Typsystem und Typdeklarationen von C	36
Objekte, Methoden, Konstruktoren	37
Spezialisierung und Vererbung	40
Subtyping und dynamisches Binden	41
Programmvergleich mit einer Java-Fassung	42
1.3.3 Objektorientierte Sprachen im Überblick	44
1.4 Aufbau und thematische Einordnung des Buches	46
2. Objekte, Klassen, Kapselung	49
2.1 Objekte und Klassen	49
2.1.1 Beschreibung von Objekten	50
2.1.2 Klassen beschreiben Objekte	51
2.1.3 Benutzen und Entwerfen von Klassen	59
2.1.4 Spracherweiterungen	62
Initialisierung, Felder und Überladen	63
Klassenmethoden und Klassenattribute	68
Zusammenwirken der Spracherweiterungen	70

2.1.5	Rekursive Klassendeklaration	73
2.1.6	Typkonzept und Parametrisierung von Klassen	76
2.1.7	Klassen als Objekte	78
2.2	Kapselung und Strukturierung von Klassen	85
2.2.1	Kapselung und Schnittstellenbildung: Erste Schritte	85
2.2.2	Strukturieren von Klassen	87
	Innere Klassen	87
	Modularisierung von Programmen: Pakete	94
2.2.3	Beziehungen zwischen Klassen	101
3.	Vererbung und Subtyping	103
3.1	Klassifizieren von Objekten	103
3.2	Subtyping und Schnittstellen	111
3.2.1	Subtyping und Realisierung von Klassifikationen	111
	Deklaration von Schnittstellentypen und Subtyping	113
	Klassifikation und Subtyping	117
	Subtyping und dynamische Methodenauswahl	123
3.2.2	Subtyping genauer betrachtet	124
	Subtyping bei vordefinierten Typen und Feldtypen	124
	Was es heißt, ein Subtyp zu sein	127
	Unterschiedliche Arten von Polymorphie	134
3.2.3	Programmieren mit Schnittstellen	135
	Schnittstellen zur Realisierung von Methodenparametern ..	136
	Beobachter und lokale Klassen	138
3.3	Vererbung	143
3.3.1	Vererbung: Das Sprachkonzept und seine Anwendung	143
	Vererbung von Programmteilen	143
	Erweitern und Anpassen von Ererbtem	144
	Spezialisieren mit Vererbung	147
3.3.2	Vererbung + Subtyping = Subclassing	153
3.3.3	Vererbung und Kapselung	157
	Kapselungskonstrukte im Zusammenhang mit Vererbung ..	158
	Zusammenspiel von Vererbung und Kapselung	160
	Realisierung gekapselter Objektgeflechte	162
3.4	Objektorientierte Programmierung und Wiederverwendung	171
4.	Bausteine für objektorientierte Programme	173
4.1	Bausteine und Bibliotheken	173
4.1.1	Bausteine in der Programmierung	173
4.1.2	Überblick über die Java-Bibliothek	176
4.2	Ausnahmebehandlung mit Bausteinen	179
4.2.1	Eine Hierarchie von einfachen Bausteinen	179
4.2.2	Zusammenspiel von Sprache und Bibliothek	181

- 4.3 Stromklassen: Bausteine zur Ein- und Ausgabe 183
 - 4.3.1 Ströme: Eine Einführung 183
 - 4.3.2 Ein Baukasten mit Stromklassen 187
 - Javas Stromklassen: Eine Übersicht 187
 - Ströme von Objekten 191

- 5. Objektorientierte Programmgerüste 197**
 - 5.1 Programmgerüste: Eine kurze Einführung 197
 - 5.2 Ein Gerüst für Bedienoberflächen: Das AWT 200
 - 5.2.1 Aufgaben und Aufbau graphischer Bedienoberflächen 201
 - 5.2.2 Die Struktur des Abstract Window Toolkit 202
 - Das abstrakte GraBo-Modell des AWT 203
 - Komponenten 204
 - Darstellung 207
 - Ereignissteuerung 208
 - Programmtechnische Realisierung des AWT im Überblick . 213
 - 5.2.3 Praktische Einführung in das AWT 214
 - Initialisieren und Anzeigen von Hauptfenstern 214
 - Behandeln von Ereignissen 216
 - Elementare Komponenten 218
 - Komponentendarstellung selbst bestimmen 223
 - Layout-Manager: Anordnen von Komponenten 225
 - Erweitern des AWT 231
 - Rückblick auf die Einführung ins AWT 234
 - 5.3 Anwendung von Programmgerüsten 235
 - 5.3.1 Programmgerüste und Software-Architekturen 235
 - 5.3.2 Entwicklung graphischer Bedienoberflächen 238
 - Anforderungen 239
 - Entwicklung von Anwendungsschnittstelle und Dialog 241
 - Entwicklung der Darstellung 242
 - Realisierung der Steuerung 247
 - Zusammenfassende Bemerkungen 247

- 6. Parallelität in objektorientierten Programmen 249**
 - 6.1 Parallelität und Objektorientierung 249
 - 6.1.1 Allgemeine Aspekte von Parallelität 250
 - 6.1.2 Parallelität in objektorientierten Sprachen 252
 - 6.2 Lokale Parallelität in Java-Programmen 253
 - 6.2.1 Java-Threads 253
 - Programmtechnische Realisierung von Threads in Java 254
 - Benutzung von Threads 257
 - 6.2.2 Synchronisation 265
 - Synchronisation: Problemquellen 266
 - Ein objektorientiertes Monitorkonzept 268

Synchronisation mit Monitoren	272
Das Speichermodell paralleler Java-Programme	279
6.2.3 Sprachliche Umsetzung von lokaler Parallelität	281
7. Programmierung verteilter Objekte	283
7.1 Verteilte objektorientierte Systeme	283
7.1.1 Grundlegende Aspekte verteilter Systeme	283
7.1.2 Programmierung verteilter objektorientierter Systeme	286
7.2 Kommunikation über Sockets	289
7.2.1 Sockets: Allgemeine Eigenschaften	289
7.2.2 Realisierung eines einfachen Servers	290
7.2.3 Realisierung eines einfachen Clients	293
7.2.4 Client und Server im Internet	295
Dienste im Internet	296
Zugriff auf einen http-Server	297
Netzsurfer im Internet	298
7.2.5 Server mit mehreren Ausführungssträngen	301
7.3 Kommunikation verteilter Objekte über entfernten	
Methodenaufruf	302
7.3.1 Problematik entfernter Methodenaufrufe	303
Behandlung verteilter Objekte	303
Simulation entfernter Methodenaufrufe über Sockets	306
7.3.2 Realisierung von entfernten Methodenaufrufen in Java	310
Der Stub-Skeleton-Mechanismus	310
Entfernter Methodenaufruf in Java	311
Parameterübergabe bei entferntem Methodenaufruf	316
8. Zusammenfassung, Varianten, Ausblick	325
8.1 Objektorientierte Konzepte zusammengefasst	325
8.2 Varianten objektorientierter Sprachen	328
8.2.1 Objektorientierte Erweiterung prozeduraler Sprachen	328
8.2.2 Originär objektorientierte Sprachen	331
Typisierte objektorientierte Sprachen	332
Untypisierte objektorientierte Sprachen	333
8.3 Zukünftige Entwicklungslinien	335
Literaturverzeichnis	339
Stichwortverzeichnis	343