

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	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
1.3.1.1 Objekte und Werte: Eine begriffliche Abgrenzung	22
1.3.1.2 Objektreferenzen, Werte, Typen und Variablen	23
1.3.1.3 Anweisungen, Blöcke und deren Ausführung	28
1.3.2 Objektorientierte Programmierung mit prozeduralen Sprachen	35
1.3.2.1 Typsystem und Typdeklarationen von C	36
1.3.2.2 Objekte, Methoden, Konstruktoren	37
1.3.2.3 Spezialisierung und Vererbung	40
1.3.2.4 Subtyping und dynamisches Binden	41
1.3.2.5 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	63
2.1.4.1 Initialisierung, Felder und Überladen	63
2.1.4.2 Klassenmethoden und Klassenattribute	69
2.1.4.3 Zusammenwirken der Spracherweiterungen	70
2.1.5 Rekursive Klassendeklaration	74

2.1.6 Typkonzept und Parametrisierung von Klassen	77
2.1.7 Klassen als Objekte	80
2.2 Kapselung und Strukturierung von Klassen	86
2.2.1 Kapselung und Schnittstellenbildung: Erste Schritte	86
2.2.2 Strukturieren von Klassen	88
2.2.2.1 Innere Klassen	88
2.2.2.2 Modularisierung von Programmen: Pakete	95
2.2.3 Beziehungen zwischen Klassen	102
3. Vererbung und Subtyping	105
3.1 Klassifizieren von Objekten	105
3.2 Subtyping und Schnittstellen	113
3.2.1 Subtyping und Realisierung von Klassifikationen	113
3.2.1.1 Deklaration von Schnittstellentypen und Subtyping	115
3.2.1.2 Klassifikation und Subtyping	119
3.2.1.3 Subtyping und dynamische Methodenauswahl	125
3.2.2 Subtyping genauer betrachtet	126
3.2.2.1 Subtyping bei vordefinierten Typen und Feldtypen	126
3.2.2.2 Was es heißt, ein Subtyp zu sein	130
3.2.2.3 Unterschiedliche Arten von Polymorphie	136
3.2.3 Programmieren mit Schnittstellen	138
3.2.3.1 Schnittstellen zur Realisierung von Methodenparametern	138
3.2.3.2 Beobachter und lokale Klassen	140
3.3 Vererbung	145
3.3.1 Vererbung: Das Sprachkonzept und seine Anwendung	145
3.3.1.1 Vererbung von Programmteilen	145
3.3.1.2 Erweitern und Anpassen von Erbtem	147
3.3.1.3 Spezialisieren mit Vererbung	149
3.3.2 Vererbung + Subtyping = Subclassing	155
3.3.3 Vererbung und Kapselung	160
3.3.3.1 Kapselungskonstrukte im Zusammenhang mit Vererbung	160
3.3.3.2 Zusammenspiel von Vererbung und Kapselung	163
3.3.3.3 Realisierung gekapselter Objektgeflechte	164
3.4 Objektorientierte Programmierung und Wiederverwendung	174
4. Bausteine für objektorientierte Programme	175
4.1 Bausteine und Bibliotheken	175
4.1.1 Bausteine in der Programmierung	175
4.1.2 Überblick über die Java-Bibliothek	178
4.2 Ausnahmebehandlung mit Bausteinen	181
4.2.1 Eine Hierarchie von einfachen Bausteinen	181
4.2.2 Zusammenspiel von Sprache und Bibliothek	183

4.3	Stromklassen: Bausteine zur Ein- und Ausgabe	185
4.3.1	Ströme: Eine Einführung	185
4.3.2	Ein Baukasten mit Stromklassen	189
4.3.2.1	Javas Stromklassen: Eine Übersicht	189
4.3.2.2	Ströme von Objekten	193
5.	Objektorientierte Programmgerüste	199
5.1	Programmgerüste: Eine kurze Einführung	199
5.2	Ein Gerüst für Bedienoberflächen: Das AWT	202
5.2.1	Aufgaben und Aufbau graphischer Bedienoberflächen	203
5.2.2	Die Struktur des Abstract Window Toolkit	204
5.2.2.1	Das abstrakte GraBo-Modell des AWT	205
5.2.2.2	Komponenten	206
5.2.2.3	Darstellung	209
5.2.2.4	Ereignissesteuerung	210
5.2.2.5	Programmtechnische Realisierung des AWT im Überblick	215
5.2.3	Praktische Einführung in das AWT	216
5.2.3.1	Initialisieren und Anzeigen von Hauptfenstern	216
5.2.3.2	Behandeln von Ereignissen	218
5.2.3.3	Elementare Komponenten	220
5.2.3.4	Komponentendarstellung selbst bestimmen	225
5.2.3.5	Layout-Manager: Anordnen von Komponenten	227
5.2.3.6	Erweitern des AWT	233
5.2.3.7	Rückblick auf die Einführung ins AWT	236
5.3	Anwendung von Programmgerüsten	237
5.3.1	Programmgerüste und Software-Architekturen	237
5.3.2	Entwicklung graphischer Bedienoberflächen	240
5.3.2.1	Anforderungen	241
5.3.2.2	Entwicklung von Anwendungsschnittstelle und Dialogführung	243
5.3.2.3	Entwicklung der Darstellung	244
5.3.2.4	Realisierung der Steuerung	249
5.3.2.5	Zusammenfassende Bemerkungen	249
6.	Parallelität in objektorientierten Programmen	251
6.1	Parallelität und Objektorientierung	251
6.1.1	Allgemeine Aspekte von Parallelität	252
6.1.2	Parallelität in objektorientierten Sprachen	254
6.2	Lokale Parallelität in Java-Programmen	255
6.2.1	Java-Threads	255
6.2.1.1	Programmtechnische Realisierung von Threads in Java	256
6.2.1.2	Benutzung von Threads	259

6.2.2 Synchronisation	267
6.2.2.1 Synchronisation: Problemquellen	268
6.2.2.2 Ein objektorientiertes Monitorkonzept	270
6.2.2.3 Synchronisation mit Monitoren	274
6.2.2.4 Das Speichermodell paralleler Java-Programme ..	281
6.2.3 Sprachliche Umsetzung von lokaler Parallelität	283
7. Programmierung verteilter Objekte	285
7.1 Verteilte objektorientierte Systeme	285
7.1.1 Grundlegende Aspekte verteilter Systeme	285
7.1.2 Programmierung verteilter objektorientierter Systeme ..	288
7.2 Kommunikation über Sockets	291
7.2.1 Sockets: Allgemeine Eigenschaften	291
7.2.2 Realisierung eines einfachen Servers	292
7.2.3 Realisierung eines einfachen Clients	295
7.2.4 Client und Server im Internet	297
7.2.4.1 Dienste im Internet	298
7.2.4.2 Zugriff auf einen http-Server	299
7.2.4.3 Netzsurfer im Internet	300
7.2.5 Server mit mehreren Ausführungssträngen	303
7.3 Kommunikation über entfernten Methodenaufruf	304
7.3.1 Problematik entfernter Methodenaufrufe	305
7.3.1.1 Behandlung verteilter Objekte	305
7.3.1.2 Simulation entfernter Methodenaufrufe über Sockets	308
7.3.2 Realisierung von entfernten Methodenaufrufen in Java ..	312
7.3.2.1 Der Stub-Skeleton-Mechanismus	312
7.3.2.2 Entfernter Methodenaufruf in Java	313
7.3.2.3 Parameterübergabe bei entferntem Methodenaufruf	318
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
8.2.2.1 Typisierte objektorientierte Sprachen	331
8.2.2.2 Untypisierte objektorientierte Sprachen	333
8.3 Zukünftige Entwicklungslinien	335
Literaturverzeichnis	339
Stichwortverzeichnis	343