

# Inhaltsverzeichnis

**Vorwort.....3**

**Benutzung des Buches.....4**

**Wege durch das Buch.....5**

**1 Einführung.....11**

1.1 Geschichte.....11

1.2 Aufbau.....12

1.2.1 Compiler.....12

1.2.2 Interpreter.....12

1.2.3 Java / Compiler und Interpreter.....13

1.3 Abgrenzung.....13

1.3.1 JavaScript.....13

1.3.2 Java-Applet.....13

1.3.3 Java-Applikation.....13

1.4 Sprachmerkmale.....14

1.5 Tools / Entwicklungsumgebung.....15

1.5.1 Tools.....15

1.5.2 Entwicklungsumgebungen.....16

1.5.3 Umgebung (Paket \_umgebung).....17

**2 Basiswissen.....19**

2.1 Erste Beispielprogramme.....19

2.1.1 Hallo Welt.....19

2.1.2 Namenskonventionen.....20

2.1.3 Kommentare.....20

2.1.4 Bildschirmausgabe / Tastatureingabe.....21

2.2 Primitive Datentypen.....22

2.2.1 Eigenschaften primitiver Datentypen.....23

2.2.2 Operatoren.....25

2.2.3 Typkonvertierung.....26

2.2.4 Sichtbarkeit lokaler Variablen.....27

2.2.5 Konstanten.....27

2.3 Kontrollstrukturen.....28

2.3.1 Bedingungen.....28

2.3.2 Verzweigung.....30

2.3.3 Schleifen.....31

2.4 Funktionen / Methoden.....33

2.4.1 Funktions- / Methodendefinition.....33

2.4.2 Aufruf einer Funktion / Methode.....34

2.4.3 Call by Value.....34

2.4.4 Beispielprogramm Überladen.....34

2.4.5 EXKURS: Struktogramme.....35

2.5 Referentielle Datentypen.....36

2.5.1 Eigenschaften referentieller Datentypen.....36

2.5.2 Array.....37

2.5.3 EXKURS: Iterationsschleifen.....40

2.5.4 Beispielprogramme Array.....41

2.5.5 String.....43

2.5.6 Nützliche Stringmethoden.....45

2.5.7 Konvertierung String – char – int.....46

2.5.8 Beispielprogramme String.....47

2.5.9 Die Klasse StringBuilder.....47

2.5.10 Call-by-Reference.....48

**3 Erweiterte Themen.....49**

3.1 EXKURS: Ausliefern von Java- Programmen.....49

3.2 EXKURS: Übergabeparameter der main()-Funktion auswerten.....49

3.3 EXKURS: Methoden mit variabler Parameterliste.....51

3.4 EXKURS: Die Klasse Math.....52

3.5 EXKURS: Rekursion / Iteration.....53

3.6 EXKURS: Sortieralgorithmus.....54

3.6.1 BubbleSort-Algorithmus.....54

3.6.2 QuickSort-Algorithmus.....55

**4 OO-Grundlagen.....56**

4.1 Objekte / Klasse.....56

4.1.1 Objektvariable.....56

4.1.2 Objektmethode.....56

4.1.3 Klasse.....56

4.1.4 Objekt.....57

4.1.5 Ausgabe.....58

4.1.6 Beispielprogramm.....58

4.2 Konstruktor / Destruktor.....59

4.2.1 Konstruktor.....59

4.2.2 this-Referenz / this()-Konstruktoraufruf.....60

4.2.3 Destruktor.....61

4.2.4 UML-Klassendiagramm.....61

4.2.5 Beispielprogramm.....62

4.3 Sichtbarkeiten und Kapselung.....63

4.3.1 Sichtbarkeiten (Teil 1).....63

4.3.2 Kapselung.....64

4.4 Das Schlüsselwort static.....66

4.4.1 Beispiel Objektzähler.....66

4.4.2 Beispielprogramm.....67

4.5 EXKURS: Besondere Datentypen.....68

4.5.1 Wrapper-Klassen.....68

4.5.2 Aufzählungstyp.....69

4.6 EXKURS: Aufbau von Java-Programmen...70

4.6.1 Module.....71

## Inhaltsverzeichnis

<b>5</b>	<b>Beziehungen.....73</b>	<b>7.4</b>	<b>Zusammenfassung UML- Umsetzungshinweise..... 119</b>
5.1	Aufrufbeziehung..... 73	7.4.1	Assoziation..... 119
5.1.1	Aufrufbeziehungen im UML-Klassendiagramm..... 74	7.4.2	Assoziation und Aggregation..... 119
5.1.2	Aufrufbeziehungen im UML-Sequenzdiagramm... 74	7.4.3	Komposition..... 121
5.2	Ist-Teil-Beziehung Aggregation / Komposition ..... 75	7.4.4	Generalisierung / Spezialisierung Vererbung..... 121
5.2.1	Aggregation..... 75	7.4.5	Interface..... 122
5.2.2	Aggregation in UML..... 76	7.4.6	Adapterklasse..... 123
5.2.3	Komposition..... 76	7.4.7	Fazit..... 123
5.2.4	Komposition im UML..... 76	<b>8</b>	<b>Exceptions / Dateizugriff..... 124</b>
5.2.5	Innere Klassen (mit static)..... 77	8.1	Exception..... 124
5.2.6	Innere Klassen (ohne static)..... 78	8.1.1	Prinzipieller Aufbau..... 125
5.2.7	Lokale Klasse / Anonyme Klasse..... 78	8.1.2	Vorgegebene Exception-Klasse..... 126
5.2.8	Aggregation mit beliebig vielen Teilen..... 79	8.1.3	Eigene Exception-Klasse..... 127
5.3	Generalisierung / Spezialisierung..... 80	8.1.4	Mehrere Exceptions..... 128
5.3.1	Generalisierung / Spezialisierung in UML..... 80	8.1.5	Catch or Throw -Regel..... 129
5.3.2	Sichtbarkeiten (Teil 2) protected..... 81	8.1.6	Bedeutung der finally-Anweisung..... 130
5.3.3	Überlagern von Methoden..... 82	8.2	Dateizugriff..... 131
5.3.4	super-Referenz..... 83	8.2.1	I/O-Exception..... 131
5.3.5	Das Schlüsselwort final..... 84	8.2.2	Dateinamen und Pfadangaben..... 131
5.3.6	Vererben verboten..... 84	8.2.3	Das Interface Path..... 132
5.3.7	super()-Konstruktoraufruf..... 85	8.2.4	Datei-Operationen (Klasse Files)..... 133
5.3.8	Automatisierte Konstruktorenverkettung..... 87	8.2.5	Parameterkontrolle (Klasse Files)..... 134
5.3.9	Polymorphismus..... 89	8.2.6	Dateizugriff (komplette Datei)..... 135
5.4	G/S vs. Aggregation..... 91	8.2.7	Text-Dateizugriff (Teil einer Datei)..... 136
5.5	Beziehungsmerkmal..... 93	8.2.8	Datenströme..... 137
<b>6</b>	<b>Abstr. Klasse / Interfaces.....95</b>	8.2.9	Verketten von Datenströmen..... 137
6.1	Abstrakte Klasse..... 95	8.2.10	Ein- / Ausgabe ganzer Objekte..... 138
6.1.1	Abstrakte Klassen / - Methoden in UML..... 97	8.2.11	Wahlfreier Dateizugriff..... 139
6.2	Interface (einfach)..... 97	8.2.12	Verzeichnis-Operationen (Klasse Files)..... 140
6.2.1	Interfaces in UML..... 99	8.2.13	Dateiattribute..... 142
6.2.2	Abstrakte Klassen vs. Interfaces..... 99	<b>9</b>	<b>Thread/Synchronisation..... 145</b>
6.2.3	Anwendungsbeispiel Interface..... 100	9.1	Threads erzeugen und steuern..... 145
6.3	Adapterklasse..... 102	9.1.1	Ein erstes Beispiel..... 145
6.4	Interfaces (erweitert)..... 103	9.1.2	Methoden der Klasse Thread..... 146
<b>7</b>	<b>Generics / Collections..... 104</b>	9.1.3	Konstruktoren der Klasse Thread..... 147
7.1	Generics..... 104	9.1.4	Beenden eines Threads..... 148
7.1.1	Generische Klasse (verkettete Liste)..... 106	9.2	Threads synchronisieren..... 149
7.1.2	Generische Methoden (Vergleich) Erweiterte Typbedingung..... 108	9.2.1	Beispiel gemeinsame Klassenvariable..... 149
7.2	Collections..... 111	9.2.2	Beispiel gemeinsame Objektvariable..... 150
7.2.1	Collection-API-Aufbau..... 111	9.2.3	Das Badezimmerproblem..... 151
7.2.2	Datenstrukturen..... 112	9.2.4	synchronized..... 152
7.2.3	Implementierungseigenschaften..... 113	9.2.5	Schlüsselproblem..... 155
7.2.4	Verwenden einer Datenstruktur..... 114	9.2.6	Warteschlange..... 157
7.2.5	Integrieren einer Datenstruktur..... 115	9.2.7	Scheduling..... 157
7.2.6	Einen eigenen Iterator verwenden..... 117	9.2.8	Alternative Synchronisation..... 157
7.3	Ist-Teil-Beziehung dynamisch..... 118	9.3	Thread als Dämon..... 157
		<b>10</b>	<b>Funktionale Programmierung..... 158</b>
		10.1	Lambda-Ausdrücke..... 158
		10.1.1	Erstes Beispiel..... 158
		10.1.2	Hintergrund zum Lambda-Ausdruck..... 159
		10.1.3	Lambda-Notation..... 160

## Inhaltsverzeichnis

10.1.4	Methoden- und Konstruktorreferenzen.....	160	13.2.1	Datenbank einrichten.....	209
10.1.5	Einsatzmöglichkeiten.....	161	13.2.2	JDBC einrichten.....	210
10.2	Ströme.....	162	13.3	JDBC-Beispielprogramm.....	210
10.2.1	Datenstruktur in Strom umwandeln.....	162	13.3.1	Tabellendaten nur anzeigen.....	211
10.2.2	Arbeitsstationen.....	163	13.3.2	Tabellendaten auch ändern.....	212
10.2.3	Beispiele.....	164	13.3.3	Datensätze erstellen/löschen.....	212
<b>11</b>	<b>AWT.....</b>	<b>166</b>	13.4	JDBC-Zugriffsoptimierung.....	213
11.1	Erstes Fenster.....	166	<b>14</b>	<b>UML-Diagramme.....</b>	<b>214</b>
11.1.1	Farbschema.....	167	14.1	Diagramme in UML.....	214
11.1.2	Zeichnen.....	169	14.2	Strukturdiagramme.....	215
11.1.3	Text.....	170	14.2.1	Klassendiagramm.....	215
11.1.4	Rahmen.....	171	14.2.2	Objektdiagramm.....	215
11.2	Event-Handling.....	172	14.2.3	Paketdiagramm.....	216
11.2.1	Beispiel Wetterdienst / Wetterwarnung.....	172	14.2.4	Kompositionsstrukturdiagramm.....	216
11.2.2	Beispiel Mouse-Event.....	174	14.2.5	Komponentendiagramm.....	218
11.2.3	Event-Übersicht.....	176	14.2.6	Verteilungsdiagramm.....	218
11.2.4	Fenster schließen, aber richtig.....	177	14.3	Verhaltensdiagramme.....	219
11.2.5	Umsetzungsalternativen.....	178	14.3.1	Anwendungsfalldiagramm.....	219
11.2.6	Bedeutung der paint()-Methode.....	181	14.3.2	Aktivitätsdiagramm.....	220
11.3	Komponenten.....	184	14.3.3	Zustandsdiagramm (Zustandsautomat).....	221
11.3.1	Komponente Button.....	184	14.4	Interaktionsdiagramme.....	222
11.3.2	Weitere Komponenten.....	185	14.4.1	Sequenzdiagramm.....	222
11.3.3	Und noch mehr Komponenten.....	185	14.4.2	Kommunikationsdiagramm.....	223
11.4	LayoutManager.....	186	14.4.3	Timing-Diagramm (Zeitdiagramm).....	223
11.4.1	LayoutManager (isoliert).....	186	14.4.4	Interaktionsübersichtsdiagramm.....	224
11.4.2	Kombinieren von Layoutmanagern.....	188	14.5	Fazit zur UML.....	224
<b>12</b>	<b>Swing.....</b>	<b>190</b>	14.6	Entity-Relationship-Modell.....	224
12.1	AWT auf Swing übertragen.....	190	<b>15</b>	<b>Softwareentwurf Grundlagen.....</b>	<b>225</b>
12.2	AWT/Swing Unterschiede.....	191	15.1	Gründe und Fakten.....	225
12.2.1	Prinzipieller Fensteraufbau.....	191	15.1.1	Erfolgsquote.....	225
12.2.2	Nicht von JFrame ableiten.....	192	15.1.2	Fehleinschätzungen.....	225
12.2.3	Zusätzliche und erweiterte Hilfsmethoden.....	193	15.1.3	Teamgrößen.....	226
12.2.4	Threadsicherheit.....	193	15.1.4	Fehler vermeiden.....	226
12.2.5	Look-And-Feel.....	194	15.1.5	Qualitätsmerkmale.....	226
12.3	MVC: Interner Aufbau.....	196	15.1.6	Grundbegriffe OOA / OOD / OOP.....	227
12.3.1	Komponenten des MVC-Musters.....	196	15.2	Softwareentwicklungsmodelle.....	227
12.3.2	MVC-Beispielprogramm.....	196	15.2.1	Wasserfallmodell.....	228
12.4	Dialogfenster.....	199	15.2.2	V-Modell [XT] (eXtreme Tailoring).....	228
12.4.1	Fensterarten.....	199	15.2.3	eXtreme Programming (XP).....	229
12.4.2	Einfache Dialoge.....	199	15.2.4	RUP (Rational Unified Process).....	230
12.4.3	Komplexe Dialoge.....	200	15.2.5	Vergleich: V-Modell, XP, RUP.....	231
12.5	Komplexe Komponenten.....	202	15.3	Konkrete Arbeitsschritte.....	231
12.5.1	Komponentenübersicht.....	202	15.3.1	OOA-Arbeitsschritte.....	232
12.5.2	Beispiel JTable (und JScrollPane).....	203	15.3.2	OOD-Arbeitsschritte.....	232
<b>13</b>	<b>JDBC.....</b>	<b>208</b>	15.3.3	OOP-Arbeitsschritte.....	232
13.1	XAMPP/LAMP/WAMP.....	208	<b>16</b>	<b>OOA-Analyse.....</b>	<b>233</b>
13.2	Vorarbeiten.....	209	16.1	Lokale Bücherei (OOA).....	233

## Inhaltsverzeichnis

16.1.1	OOA-1 Systemidee und Zielsetzung.....	233	21.1.4	Die Datei activity_main.xml.....	263
16.1.2	OOA-2 Anwendungsfälle und Akteure.....	233	21.1.5	Die Datei R.java.....	264
16.1.3	OOA-3 Anwendungsfälle Essenz.....	234	21.1.6	Die Datei MainActivity.java.....	265
16.1.4	OOA-4 Funktionsklassen identifizieren.....	234	21.1.7	Die Datei AndroidMainManifest.xml.....	265
16.1.5	OOA-5 Aktivitätsdiagramm.....	237	21.1.8	Ausblick.....	266
16.1.6	OOA-6 Anwendungsfälle Details.....	237	21.2	Hintergrundinformationen.....	266
16.1.7	OOA-7 Merkmale und Methoden.....	238	21.2.1	Activity.....	266
16.1.8	Relationale Datenbank.....	239	21.2.2	Intent.....	266
16.1.9	OOA-8 Systemschnittstellen.....	240	21.2.3	Zustände und Lebenszyklus.....	267
16.2	OOA-Ergebnisse.....	242	21.2.4	Architektur von Android.....	268
<b>17</b>	<b>Entwurfsmuster.....</b>	<b>243</b>	<b>22</b>	<b>Aufgaben und Übungen.....</b>	<b>269</b>
17.1	Grundlagen.....	243	22.1	Einführung.....	270
17.1.1	Was ist ein Muster?.....	243	22.1.1	Weiterentwicklungen.....	270
17.1.2	Gang of Four (GoF) / Geschichte.....	243	22.1.2	Linker / Binder.....	270
17.1.3	Eigenschaften.....	243	22.1.3	Script-Sprache.....	270
17.1.4	Klassifizierung.....	243	22.1.4	Sprachvergleich Java, C, C++, C#.....	270
17.2	Entwurfsmuster-Beispiele.....	244	22.1.5	Grafische _umgebung.....	270
17.2.1	(E5) Einzelstück (Singleton).....	244	22.2	Basiswissen.....	271
17.2.2	(S4) Dekorierer (Decorator).....	245	22.2.1	Hallo Welt.....	271
17.2.3	(V4) Beobachter (Observer).....	246	22.2.2	Namenskonvention.....	271
17.2.4	(V6) Iterator (Cursor).....	246	22.2.3	Kommentar.....	271
17.3	Bedeutung / Fazit.....	247	22.2.4	Bildschirmausgabe/Tastatureingabe.....	272
17.3.1	Entwurfsmusterübersicht.....	247	22.2.5	Primitive Datentypen Operatoren.....	272
<b>18</b>	<b>OOD-Design.....</b>	<b>249</b>	22.2.6	Konvertierung (Teil 1) : Typcast.....	272
18.1.1	OOD-1 Programmarchitektur.....	249	22.2.7	Lebensdauer / Konstanten.....	272
18.1.2	OOD-2 Anwendungsfälle überprüfen, Benutzeroberfläche bestimmen.....	250	22.2.8	Analyse der Umgebung.....	273
18.1.3	OOD-3 Entwurfsmuster.....	252	22.2.9	Zahlenspiel.....	273
18.1.4	OOD-4 Klassendiagramm um Steuerk omponenten und Schnittstellen erweitern.....	252	22.2.10	Skatclub.....	274
18.1.5	OOD-5 Zusammenspiel zwischen Steuerklassen und Funktionsklassen überprüfen.....	254	22.2.11	Tagesausflug.....	275
<b>19</b>	<b>OOP-Programmierung.....</b>	<b>255</b>	22.2.12	Buchstabiertafeln.....	276
19.1.1	OOP-1 Sprachspezifische Eigenarten und Entwurfsmuster berücksichtigen.....	255	22.2.13	Zylinderberechnung (Teil 1).....	277
<b>20</b>	<b>XML-Template.....</b>	<b>258</b>	22.2.14	Ein-mal-Eins-Tabelle.....	278
20.1	Was ist XML?.....	258	22.2.15	Kreissegment.....	278
20.2	XML-Dokument.....	258	22.2.16	Farbpalette.....	279
20.3	XML-Schema.....	259	22.2.17	Konvertierung (Teil 2) : Überladen.....	280
20.4	XML-Verarbeitung in Java.....	260	22.2.18	Primzahlen.....	280
<b>21</b>	<b>Android-App-Entwicklung.....</b>	<b>261</b>	22.2.19	Würfelstatistik.....	281
21.1	Erste Beispielapp (KW-PS).....	262	22.2.20	Zahnputzroboter.....	282
21.1.1	Entwicklungsumgebung.....	262	22.2.21	Array-Notation.....	282
21.1.2	Verzeichnisstruktur.....	263	22.2.22	Nadelbaum.....	283
21.1.3	Die Datei string.xml.....	263	22.2.23	Pascalsches Dreieck (Teil 1).....	284
			22.2.24	Passwort-Test.....	285
			22.2.25	String-Statistik.....	285
			22.2.26	ISBN-Test.....	286
			22.2.27	Hexadezimalzahlen (Teil 1).....	287
			22.2.28	Schachspiel.....	288
			22.2.29	Tic-Tac-Toe.....	289
			22.2.30	Lotto.....	290
			22.3	Erweiterte Themen.....	291
			22.3.1	Fertiges Projekt.....	291
			22.3.2	Parameteranalyse.....	291
			22.3.3	Dateipfad.....	292
			22.3.4	Konvertierung (Teil 3) : Parameterliste.....	292
			22.3.5	Funktionsgraph.....	293
			22.3.6	Schleife mal rekursiv.....	294
			22.3.7	Pascalsches Dreieck (Teil 2) : rekursiv.....	294
			22.3.8	Fibonacci-folge.....	295
			22.3.9	Weizenkornlegende.....	295
			22.3.10	Vollständige Induktion.....	295

## Inhaltsverzeichnis

22.3.11	Kürzester Weg.....	296	22.15	Softwareentwurf Grundlagen.....	370
22.3.12	Türme von Hanoi (Teil 1).....	296	22.15.1	Gründe und Fakten.....	370
22.3.13	BubbleSort verbessern.....	297	22.15.2	Softwareentwicklungsmodelle.....	370
22.3.14	Geschwindigkeitstest (Teil 1).....	297	22.16	OOA-Analyse.....	372
<b>22.4</b>	<b>OOP-Grundlagen.....</b>	<b>298</b>	22.16.1	Grundlagen.....	372
22.4.1	Objekte / Klassen.....	300	22.16.2	Anwendungsfälle kontrollieren.....	372
22.4.2	Konstruktoren / Destruktoren.....	305	22.16.3	Funktionsklassen bestimmen.....	373
22.4.3	Sichtbarkeiten / Kapselung.....	306	22.16.4	Bücherei komplettieren (Teil1).....	373
22.4.4	Das Schlüsselwort static.....	311	22.16.5	Projekte analysieren.....	373
22.4.5	Wrapper-Klasse / Aufzählungstypen.....	313	22.17	Entwurfsmuster.....	374
22.4.6	Aufbau von Javaprogrammen.....	314	22.17.1	Grundlagen.....	374
22.5	Beziehungen.....	315	22.17.2	Klassifizierung.....	374
22.5.1	Aufrufbeziehung.....	315	22.17.3	Prophet.....	374
22.5.2	Ist-Teil-Beziehung.....	317	22.17.4	Parallelen zu Java-Sprachelementen.....	375
22.5.3	Generalisierung / Spezialisierung.....	321	22.17.5	Weitere Entwurfsmuster.....	375
22.6	Abstr. Klassen / Interfaces.....	328	22.18	OOD-Design.....	375
22.6.1	Abstrakte Klassen.....	328	22.18.1	Programmarchitektur.....	375
22.6.2	Interface.....	329	22.18.2	Softwareprojektanforderungen.....	376
22.7	Generics / Collections.....	335	22.18.3	Bücherei komplettieren (Teil 2).....	376
22.7.1	Generics.....	335	22.18.4	Projekte designen.....	376
22.7.2	Collections.....	336	22.19	OOP-Programmierung.....	376
22.7.3	Ist-Teil-Beziehung dynamisch.....	336	22.19.1	Sprachspezifische Eigenarten.....	376
22.7.4	Zusammenfassung UML-Umsetzung.....	340	22.19.2	Bücherei komplettieren (Teil 3).....	376
22.8	Exceptions / Dateizugriff.....	344	22.19.3	Projekte programmieren.....	377
22.8.1	Exceptions.....	344	22.20	XML-Template.....	377
22.8.2	Dateizugriff.....	346	22.20.1	Grundlagen.....	377
22.9	Thread / Synchronisation.....	348	22.20.2	XML-Schemadatei.....	377
22.9.1	Threads erzeugen und steuern.....	348	22.20.3	XML-Verarbeitung in Java.....	377
22.9.2	Threads synchronisieren.....	350	22.21	Android-App-Entwicklung.....	378
22.10	Funktionale Programmierung.....	354	22.21.1	Entwicklungsumgebung installieren.....	378
22.10.1	Lambda-Ausdrücke.....	354	22.21.2	Beispiel nachvollziehen.....	378
22.10.2	Ströme.....	355	22.21.3	Weitere Projekte.....	378
22.11	AWT.....	356	22.21.4	Hintergrundinformationen.....	378
22.11.1	Erstes Fenster.....	356	<b>Downloadbereich.....</b>	<b>391</b>	
22.11.2	Event-Handling.....	358			
22.11.3	Komponenten / Layoutmanager.....	360			
22.12	Swing.....	362			
22.12.1	Swing Grundlagen.....	362			
22.12.2	Umrechnen (Teil 2).....	362			
22.12.3	Zylinderberechnung (Teil 3).....	362			
22.12.4	Würfel (Teil 6).....	363			
22.12.5	Kryptographie (Teil 2).....	363			
22.12.6	Geburtstagsliste (Teil 1).....	363			
22.13	JDBC.....	364			
22.13.1	JDBC Grundlagen.....	364			
22.13.2	XAMPP.....	364			
22.13.3	Geburtstagsliste (Teil 2).....	364			
22.13.4	Vokabeltrainer.....	364			
22.14	UML-Diagramme.....	366			
22.14.1	Grundlagen.....	366			
22.14.2	Anwendungsfalldiagramm.....	367			
22.14.3	Klassendiagramm.....	367			
22.14.4	Aktivitätsdiagramm.....	368			
22.14.5	Zustandsdiagramm.....	369			
22.14.6	Sequenzdiagramme.....	369			