

Inhalt

1	Softwarearchitektur als Herausforderung	1
1.1	Was ist Softwarearchitektur?	3
1.2	Quasar: Qualitätssoftwarearchitektur	4
1.3	Der Fahrplan	5
1.3.1	Klassen und Schnittstellen (Kapitel 2)	6
1.3.2	Komponenten und Schnittstellen (Kapitel 3)	6
1.3.3	Softwarekategorien – wie findet man Komponenten? (Kapitel 4)	6
1.3.4	Fehler und Ausnahmen – Rechte und Pflichten (Kapitel 5) . . .	6
1.3.5	Wie spezifiziert man Schnittstellen? (Kapitel 6)	7
1.3.6	Softwarearchitekturen (Kapitel 7)	7
1.3.7	Anwendungskern und Anwendungskomponenten (Kapitel 8) .	8
1.3.8	Pools, Persistenz und Transaktionen (Kapitel 9)	8
1.3.9	GUI-Architektur (Kapitel 10)	8
1.4	Quasar – Hilfe oder Korsett?	8
1.5	Warum brauchen wir Quasar?	10
1.5.1	Argumente gegen Quasar	10
1.5.2	Quasar und Muster	11
1.5.3	Quasar und Wissenschaft	11
1.5.4	Quasar und UML	12
1.6	Was sagen andere?	12
2	Klassen und Schnittstellen	15
2.1	Schnittstellen	15
2.2	Listen und Iteratoren	17

2.3	Vom Nutzen des Iterators	18
2.4	Das Kleingedruckte: Was kann alles schief gehen?	19
2.5	Konfiguration und Implementierung von Klassen	21
2.6	Schnittstellen und Muster	22
2.7	Eigene Listen implementieren	24
2.8	Schnittstellen und Test	25
2.9	Weitere Java-Behälter	28
2.10	Arbeit sparen durch Schnittstellen	30
2.11	Prinzip der minimalen Annahme	31
2.12	Rückruffschnittstellen (callbacks)	31
2.12.1	Beispiel: Fahrstuhl mit virtueller Zeit	32
2.12.2	Java-Schnittstellen und funktionale Programmierung	33
2.13	Pakete in Java	34
2.13.1	Wie baut man Java-Pakete?	37
2.14	Schnittstellen in anderen Sprachen	37
2.14.1	Schnittstellen in C++	37
2.14.2	Schnittstellen in C	38
2.14.3	Schnittstellen in typfreien Sprachen: Beispiel Python	39
2.15	Zusammenfassung	40
3	Komponenten und Schnittstellen	41
3.1	Komponenten	42
3.1.1	Drei Definitionen, sechs Merkmale	42
3.2	Schnittstellen	44
3.3	Mehr Beispiele für Komponenten	46
3.4	Semantik und Verfeinerung von Schnittstellen	47
3.5	Beispiel Berechtigung	50
3.6	Komponenten und Objektorientierung	52
3.7	Rollen im Entwicklungsprozess	56
3.8	Wie findet man Komponenten: Beispiel E-Mail	57
3.9	Konfiguration: Komponenten verbinden	59
3.10	Kompositionsmanager: Konfiguration und Komposition	61

3.11	Symmetrische Komposition	64
3.12	Konfiguration mit XML	65
3.13	Wer definiert Schnittstellen?	66
3.13.1	Standardschnittstellen	67
3.13.2	Angebotene Schnittstelle	68
3.13.3	Angeforderte Schnittstellen und Adapter	68
3.14	Wie beschreibt man Komponenten?	70
3.15	Zusammenfassung	71
4	Softwarekategorien – wie findet man Komponenten?	73
4.1	Beispiel: Schafkopfen mit dem Computer	74
4.2	Softwarekategorien und Komplexität	77
4.3	Was gehört zu 0-Software?	79
4.4	Softwarekategorien und Variabilität	79
4.5	Kommunikation zwischen Komponenten verschiedener Kategorien .	80
4.5.1	Kommunikation mit neutralen Schnittstellen und R-Software	80
4.5.2	Beispiel 1: Vektoren und Matrizen	81
4.5.3	Beispiel 2: Anwendung und Persistenz	84
4.5.4	R-Software und Sichtbarkeit	87
4.6	A- und T-Software	88
4.6.1	A-Software	88
4.6.2	T-Software	88
4.6.3	Kombination von A und T	90
4.7	A und T – Revolution oder Selbstverständlichkeit?	90
4.8	Variabilitätsanalyse mit SAAM	92
4.8.1	Zurück zum Beispiel: Schafkopfen mit dem Computer	94
4.8.2	SAAM und Softwarekategorien	95
4.9	Fünf Regeln zu Softwarekategorien	95
5	Fehler und Ausnahmen – Rechte und Pflichten	97
5.1	Ausnahmen und Programmiersprachen	98
5.2	Ausnahmen und Softwarearchitektur	100
5.3	Optionen der Ausnahmebehandlung	103

5.4	Architektur der Ausnahmebehandlung	104
5.4.1	Sicherheitsfassade	105
5.4.2	Diagnose und Reparatur	107
5.4.3	Experten für Diagnose und Reparatur	108
5.4.4	Komposition als Risikogemeinschaft	109
5.4.5	Beispiel Anwendungsserver	111
5.5	Java-Ausnahmen und Performance	112
5.6	Java-Ausnahmen: Geprüft oder ungeprüft?	113
5.7	Vor- und Nachbedingungen	116
5.8	Neun Regeln zu Fehlern und Ausnahmen	118
6	Spezifikation von Schnittstellen	119
6.1	Die Schnittstelle als virtuelle Komponente	120
6.1.1	Zustand, Konsistenz, Konstanz	120
6.1.2	Ausnahmen	121
6.1.3	ACID und RR	121
6.1.4	Zusammenfassung: Annahmen bei der Spezifikation	123
6.2	Elemente der Spezifikation	123
6.2.1	Zustandsmodell	124
6.2.2	Invarianten	124
6.2.3	Vor- und Nachbedingungen	125
6.2.4	Testfälle	125
6.2.5	Weitere Angaben	125
6.3	Vorgehen bei der Spezifikation	126
6.4	Erweiterung und Verfeinerung von Schnittstellen	126
6.4.1	Erweiterung von Schnittstellen	127
6.4.2	Verfeinerung von Schnittstellen	127
6.5	Quasar Specification Language (QSL): Design by Contract	128
6.5.1	OCL	128
6.5.2	Parnas-Tabellen	128
6.5.3	Eiffel	129
6.5.4	JML	129
6.5.5	QSL	129
6.5.6	Definition der QSL-Syntax mit Java	131

6.6	Beispiel: Einfaches Berechtigungssystem	135
6.7	ACID und RRR bei einem Prozess	139
6.7.1	Seiteneffekte und RRR	139
6.7.2	Rückruf und ACID	140
6.7.3	Konsistenz und Konstanz	141
6.8	Zusammenfassung	144
7	Softwarearchitekturen	145
7.1	Übersicht	145
7.2	Architektur eines Informationssystems	149
7.3	A-Architektur	151
7.4	A und T: Zwei Welten	153
7.5	TI-Architektur	154
7.6	T-Architektur und Standard-T-Architektur	156
7.7	Rolle der Architektur im Entwicklungsprozess	158
7.8	Alternative Ansätze	160
7.8.1	Die Architektursichten von Hofmeister, Nord, Soni	160
7.8.2	MDA und MDC	161
7.9	Quasar-Konformität	163
8	Anwendungskomponenten und Anwendungskern	165
8.1	A-Komponenten und Anwendungskern	166
8.2	Welche Schnittstellen importiert der Anwendungskern?	167
8.3	Robert: Rosenheimer Stadtbücherei	169
8.4	Außensicht von A-Komponenten: Gebühren und Ausleihe	171
8.4.1	Gebühren	171
8.4.2	Ausleihe	172
8.5	Entitätstypen und Datentypen	176
8.6	Implementierung von Datentypen	179
8.6.1	Beispiele: ISBN, Versichertenart	180
8.6.2	Fehler, die man vermeiden sollte	181
8.6.3	Spezielle Transformationen	182

8.7	Enge und lose Koppelung	183
8.7.1	Objektorientierte und dienstorientierte Schnittstellen	184
8.7.2	Assoziationen	186
8.7.3	Datenhoheit	187
8.7.4	Schnittstellen im Entwicklungsprozess	188
8.8	Zusammenfassung	190
9	Pools, Persistenz und Transaktionen	191
9.1	Standardprobleme	192
9.2	Zugriffsschichten im Überblick	194
9.2.1	Außensicht	194
9.2.2	Innensicht	197
9.3	Transaktionen aus Anwendungssicht	200
9.4	Pools	203
9.4.1	Pool-Schnittstellen	204
9.4.2	SimplePool: Ein einfacher transaktionsfähiger Pool	205
9.4.3	SimplePool: Was kann er nicht?	206
9.4.4	Spätes Lesen: Virtuelle Objekte, virtuelle Attribute	207
9.4.5	Objektidentität und Beziehungen	210
9.4.6	Sperren und Transaktionen	210
9.4.7	Mapping	210
9.5	MappingManager	211
9.5.1	Außensicht	211
9.5.2	Abbildungsmodell	212
9.5.3	Innensicht	213
9.6	Wer verantwortet die Konsistenz der Daten?	213
9.7	Gespeicherte Prozeduren (Stored Procedures)	214
9.8	Mehrere Benutzer: Sperren und Sichtbarkeit	215
9.8.1	Sperren und Sperrmodi	216
9.8.2	Isolationsstufen	217
9.9	Transaktionen (Vertiefung)	218
9.9.1	Geschachtelte Transaktionen	218
9.9.2	Parallele Transaktionen	220
9.9.3	Verteilte Transaktionen	221
9.9.4	Weiche Transaktionen	222

9.10	QuasarPersistence	223
	9.10.1 Was leistet die Anwendung?	224
	9.10.2 Pool	224
	9.10.3 Abfragen	224
	9.10.4 Transaktionen, Sperren	225
	9.10.5 OR-Abbildung	225
	9.10.6 Eigenschaften von QuasarPersistence im Überblick	226
9.11	JDO (Java Data Objects)	227
	9.11.1 Was leistet die Anwendung?	228
	9.11.2 PersistenceManager	230
	9.11.3 Abfragen	231
	9.11.4 Transaktionen, Sperren	232
	9.11.5 OR-Abbildung	232
9.12	Neun Regeln zu Pools, Persistenz und Transaktionen	233
10	Architektur grafischer Bedienoberflächen	235
10.1	Begriffe	236
10.2	Prinzip lokaler Bedienoberflächen	238
10.3	Standardarchitektur einer lokalen GUI-Engine	241
	10.3.1 Model-View-Controller (MVC)	243
10.4	Web-Oberflächen und lokale Oberflächen	243
10.5	Wertschöpfungskette grafischer Bedienoberflächen	245
	10.5.1 GUI-Frontend und Präsentation	246
	10.5.2 Präsentation und Dialogkern	246
	10.5.3 Dialogkern und Anwendungskern	247
10.6	Dialogkern	247
10.7	Präsentation	248
10.8	Spezielle Themen	250
	10.8.1 Prüfung von Benutzereingaben	250
	10.8.2 Behandlung von Fehlern und Ausnahmen	251
	10.8.3 Komposition von Dialogen	252
10.9	Zusammenfassung	253

A	Autorisierung und QSL	255
A.1	Autorisierung: Übersicht	255
A.2	Autorisierung: Außensicht	256
A.3	Der algorithmische Kern	259
A.4	Beispiel Robert	259
A.5	QSL-Spezifikation	262
	A.5.1 Operative Schnittstelle	262
	A.5.2 Administration	263
	A.5.3 Instrumentierung	267
	Nachwort	269
	Literatur	271
	Index	275