
Inhaltsverzeichnis

1	Grundlagen des Software-Entwurfs	1
1.1	Entwurfsprinzipien	1
1.2	Heuristiken des objektorientierten Entwurfs	8
1.3	Grundprinzipien der Dokumentation	13
2	Grundkonstrukte der Objektorientierung in Java, C# und C++	19
2.1	Vererbung	19
2.2	Abstrakte Klassen	19
2.3	Beispiel: Ein Modell von Fahrzeugen	20
3	Erzeugungsmuster	25
3.1	Abstract Factory (Abstrakte Fabrik)	25
3.2	Builder (Erbauer)	29
3.3	Factory Method (Fabrik-Methode)	34
3.4	Singleton	38
3.5	Object Pool	43
4	Verhaltensmuster	51
4.1	Command	51
4.2	Command Processor	54
4.3	Iterator	56
4.4	Visitor (Besucher)	60
4.5	Strategy	66

4.6	Template Method (Schablonenmethode)	68
4.7	Observer	70
5	Strukturmuster	77
5.1	Adapter	77
5.2	Bridge	79
5.3	Decorator (Dekorierer)	83
5.4	Fassade	87
5.5	Proxy (Stellvertreter)	89
5.6	Model View Controller (MVC)	92
5.7	Flyweight	96
5.8	Composite (Kompositum)	102
6	Verteilung	105
6.1	Combined Method	105
6.2	Data Transfer Object (DTO, Transferobjekt)	109
6.3	Transfer Object Assembler	114
6.4	Active Object	117
6.5	Master-Slave	122
7	Integration	127
7.1	Wrapper	127
7.2	Gateway	130
7.3	PlugIn	132
7.4	Mapper	136
7.5	Dependency Injection	138
8	Persistenz	143
8.1	O/R-Mapping	143
8.2	Identity Map	152
8.3	Lazy Load (Verzögertes Laden)	155
8.4	Coarse-Grained Lock (Grobkörnige Sperre)	158
8.5	Optimistic Offline Lock (Optimistisches Sperren)	161
8.6	Pessimistic Offline Lock (Pessimistisches Sperren)	165
9	Datenbankschlüssel	171
9.1	Identity Field (Schlüsselklasse)	174
9.2	Sequenzblock	177

9.3	UUID (Universally Unique Identifier, Global eindeutiger Schlüssel)	180
10	Sonstige Patterns	183
10.1	Money (Währung)	183
10.2	Null-Objekt	186
10.3	Registry	189
10.4	Rohbau (Building Shell)	191
10.5	Service Stub	193
10.6	Value Object (Wertobjekt)	195
10.7	Schablonendokumentation	197
11	Patterns – Wie geht es weiter?	205
11.1	Patterns erleichtern Wissenstransfer	205
	Literatur	213
	Sachverzeichnis	217
	Kolophon	223