

Inhaltsverzeichnis

	Vorwort	17
	Geleitwort von Grady Booch	21
	Einleitung	23
I	Einführung	25
I.1	Was ist ein Design Pattern?	27
I.2	Design Patterns im Smalltalk MVC	30
I.3	Beschreibung der Design Patterns	32
I.4	Der Design-Patterns-Katalog	34
I.5	Aufbau des Katalogs	38
I.6	Die Anwendung von Design Patterns zur Behebung von Designproblemen	41
I.6.1	Passende Objekte finden	41
I.6.2	Objektgranularität bestimmen	42
I.6.3	Objektschnittstellen spezifizieren	42
I.6.4	Objektimplementierungen spezifizieren	44
I.6.5	Wiederverwendungsmechanismen einsetzen	49
I.6.6	Strukturen der Laufzeit und beim Kompilieren abstimmen	54
I.6.7	Designänderungen berücksichtigen	56
I.7	Auswahl eines Design Patterns	63
I.8	Anwendung eines Design Patterns	65
2	Fallstudie: Erstellung eines Texteditors	67
2.1	Designprobleme	67
2.2	Dokumentstruktur	69
2.2.1	Rekursive Komposition	71
2.2.2	Glyphen	72
2.2.3	Das Design Pattern <i>Composite</i> (<i>Kompositum</i>)	75
2.3	Formatierung	76
2.3.1	Kapselung des Formatierungsalgorithmus	76
2.3.2	Die Unterklassen <i>Compositor</i> und <i>Composition</i>	77
2.3.3	Das Design Pattern <i>Strategy</i> (<i>Strategie</i>)	79

2.4	Ausgestaltung der Benutzeroberfläche	80
2.4.1	Durchsichtige Umhüllung (Transparent Enclosure)	80
2.4.2	Die Unterklasse MonoGlyph	82
2.4.3	Das Design Pattern <i>Decorator</i> (<i>Dekorierer</i>)	84
2.5	Unterstützung mehrerer Look-and-Feel-Standards	85
2.5.1	Abstrahierung der Objekterzeugung	85
2.5.2	Factories und Produktklassen	86
2.5.3	Das Design Pattern <i>Abstract Factory</i> (<i>Abstrakte Fabrik</i>)	89
2.6	Unterstützung mehrerer Fenstersysteme	90
2.6.1	Eignung des Design Patterns <i>Abstract Factory</i> (<i>Abstrakte Fabrik</i>)	90
2.6.2	Kapselung von Implementierungsabhängigkeiten	91
2.6.3	Die Klassenhierarchien Window und WindowImp	93
2.6.4	Das Design Pattern <i>Bridge</i> (<i>Brücke</i>)	97
2.7	Userseitige Operationen	98
2.7.1	Kapselung eines Requests	99
2.7.2	Die Command-Klasse und ihre Unterklassen	100
2.7.3	Die Funktion Undo (Rückgängig)	101
2.7.4	Befehlshistorie	102
2.7.5	Das Design Pattern <i>Command</i> (<i>Befehl</i>)	104
2.8	Rechtschreibprüfung und Silbentrennung	105
2.8.1	Zugriff auf verteilte Informationen	105
2.8.2	Kapselung von Zugriff und Traversierung	106
2.8.3	Die Iterator-Klasse und ihre Unterklassen	108
2.8.4	Das Design Pattern <i>Iterator</i> (<i>Iterator</i>)	111
2.8.5	Traversierung kontra Traversierungsaktionen	111
2.8.6	Kapselung der Analyse	112
2.8.7	Die Visitor-Klasse und ihre Unterklassen	117
2.8.8	Das Design Pattern <i>Visitor</i> (<i>Besucher</i>)	118
2.9	Zusammenfassung	119

Design-Patterns-Katalog

121

3	Erzeugungsmuster (Creational Patterns)	123
3.1	Abstract Factory (Abstrakte Fabrik)	128
3.2	Builder (Erbauer)	139
3.3	Factory Method (Fabrikmethode)	149
3.4	Prototype (Prototyp)	161

3.5	Singleton (Singleton)	172
3.6	Weitere Erläuterungen zu den Erzeugungsmustern	181
4	Strukturmuster (Structural Patterns)	185
4.1	Adapter (Adapter)	187
4.2	Bridge (Brücke)	200
4.3	Composite (Kompositum)	212
4.4	Decorator (Dekorierer)	226
4.5	Facade (Fassade)	237
4.6	Flyweight (Fliegengewicht)	247
4.7	Proxy (Proxy)	262
4.8	Weitere Erläuterungen zu den Strukturmustern	275
4.8.1	<i>Adapter (Adapter, siehe Abschnitt 4.1) kontra Bridge (Brücke, siehe Abschnitt 4.2)</i>	275
4.8.2	<i>Composite (Kompositum, siehe Abschnitt 4.3) kontra Decorator (Dekorierer, siehe Abschnitt 4.4) kontra Proxy (Proxy, siehe Abschnitt 4.7)</i>	276
5	Verhaltensmuster (Behavioral Patterns)	279
5.1	Chain of Responsibility (Zuständigkeitskette)	280
5.2	Command (Befehl)	292
5.3	Interpreter (Interpreter)	304
5.4	Iterator (Iterator)	320
5.5	Mediator (Vermittler)	338
5.6	Memento (Memento)	349
5.7	Observer (Beobachter)	360
5.8	State (Zustand)	372
5.9	Strategy (Strategie)	383
5.10	Template Method (Schablonenmethode)	394
5.11	Visitor (Besucher)	400
5.12	Weitere Erläuterungen zu den Verhaltensmustern	417
5.12.1	Variieren der Kapselung	417
5.12.2	Objekte als Argumente	418
5.12.3	Kommunikation: Kapseln oder Verteilen?	418
5.12.4	Absender und Empfänger entkoppeln	419
5.12.5	Zusammenfassung	422

6	Schlusswort der Autoren	425
6.1	Was kann man von Design Patterns erwarten?	425
6.1.1	Ein gemeinsames Designvokabular	426
6.1.2	Eine Dokumentations- und Lernhilfe	426
6.1.3	Eine Ergänzung zu existierenden Methoden	427
6.1.4	Zielsetzungen für Refactorings	428
6.2	Eine kleine Kataloggeschichte	430
6.3	Die Pattern-Gemeinde	431
6.3.1	Christopher Alexanders »Muster-Sprache«	432
6.3.2	Patterns in Softwaresystemen	433
6.4	Eine Einladung	434
6.5	Ein abschließender Gedanke	435
A	Glossar	437
B	Notationshinweise	443
B.1	Klassendiagramme	444
B.2	Objektdiagramme	447
B.3	Interaktionsdiagramme	447
C	Fundamentale Klassen	449
C.1	Die Klasse List	449
C.2	Iterator	452
C.3	ListIterator	452
C.4	Point	453
C.5	Rect	454
D	Quellenverzeichnis	455
	Stichwortverzeichnis	461