

Gesamtinhaltsverzeichnis

1	Die Java Enterprise Edition.....	1-3
1.1	Dienste	1-3
1.1.1	Transaktionen	1-4
1.1.2	Authentifizierung	1-5
1.1.3	Autorisierung	1-6
1.1.4	Ausfallsicherheit.....	1-6
1.1.5	Zugriff auf persistente Daten.....	1-8
1.1.6	Gesamtbild.....	1-9
1.2	JEE Komponentenmodelle	1-10
1.2.1	Der Servlet Container	1-10
1.2.2	Der Enterprise JavaBean Container	1-11
1.2.3	Stateless SessionBeans	1-14
1.2.4	Stateful SessionBeans.....	1-15
1.2.5	MessageDrivenBeans.....	1-16
1.2.6	EntityBeans.....	1-17
1.2.7	Die Java Connector Architecture	1-18
1.3	Das Basis-Design mit JEE-Komponenten	1-20
2	Die Java Enterprise Architekturen.....	2-3
2.1	Architekturen im Überblick.....	2-3
2.1.1	Monolithische Architekturen	2-3
2.1.2	Client-Server Architektur	2-4
2.1.3	3-Schichtige Architektur	2-4
2.1.4	n-schichtige e-Business Plattformen.....	2-4
2.1.5	Cluster	2-6
2.2	Moderne IT-Architekturen	2-7
2.2.1	Architektur-Vorgaben	2-7
2.2.2	Die Rolle des Applikationsservers	2-10
2.2.3	Kommunikationsprotokolle	2-12
2.3	Java Applikationsserver und die Java Enterprise Edition	2-15
2.3.1	Implementierungen	2-15
2.3.2	Interne Dienste	2-16
2.3.3	Inbound Connectors.....	2-17
2.3.4	JEE Komponenten	2-18

2.3.5	Outbund Connectors.....	2-19
2.4	Programmiermodell	2-20
2.4.1	Allgemeines	2-20
2.4.2	Stateless SessionBeans	2-20
2.4.3	Context and Dependency Injection	2-21
2.4.4	Java Persistence API.....	2-22
2.4.5	JavaServer Faces	2-23
2.4.6	Jax-WS	2-23
2.4.7	Jax-RS	2-26
2.5	Patterns und Idiome	2-28
2.5.1	Was ist mit den J2EE-Pattern?	2-28
2.5.2	GoF Patterns	2-29
2.5.3	Fowler-Patterns	2-29
3	JEE und die Business Schicht.....	3-3
3.1	Services.....	3-3
3.1.1	Definition.....	3-3
3.1.2	Service-Beschreibung und Java	3-3
3.1.3	Service-Realisierung in Java	3-4
3.2	Context and Dependency Injection.....	3-6
3.3	Dekoration	3-9
3.3.1	Business Object.....	3-9
3.3.2	Container	3-10
3.3.3	Aspektorientierte Programmierung	3-11
3.3.4	Das JEE Pattern Layer Supertype	3-11
3.4	Realisierung	3-12
3.4.1	Stateless SessionBeans	3-12
3.4.2	Erweiterte Paketstruktur.....	3-13
3.5	Zusammenfassung	3-14
3.6	Beispiel: Eine SessionBean für den KeyGenerator	3-14
4	Verteilte Anwendungen	4-3
4.1	RMI/IIOP.....	4-3
4.1.1	Allgemeines	4-3
4.1.2	Exkurs: Session Facade und Business Delegate der J2EE	4-4
4.1.3	EJB 3	4-12
4.1.4	Remote Facade, spezielle Business Delegates und Transfer Objects	4-12
4.1.5	Erweiterungen.....	4-17
4.2	Die Message Facade.....	4-35

4.2.1	Vorteile des JMS-Protokolls	4-35
4.2.2	Layer Supertype und Basisklasse für den Business Delegate	4-36
4.2.3	Der Service to Worker.....	4-42
4.2.4	Message Facade und Klassendiagramm.....	4-51
4.2.5	Dokumenten-orientierte Services.....	4-51
4.2.6	Dokumenten-orientierte Remote Facade	4-52
4.3	Die Http Facade.....	4-53
4.4	Die REST Facade.....	4-56
4.5	Die SOAP Facade	4-58
5	Typische Design-Beispiele aus dem JEE Umfeld	5-3
5.1	Verteilte Anwendungen	5-3
5.1.1	RMI/IIOP	5-3
5.1.2	Die Message Facade	5-15
5.1.3	Die Http Facade	5-19
5.1.4	Die SOAP Facade.....	5-20
5.2	Zugriff auf Entities.....	5-21
5.2.1	Lokale Services und Entities.....	5-21
5.2.2	Transaktionsskripte.....	5-25
5.2.3	Fast Lane Reader	5-26
5.2.4	EntityBeans der JEE 1.4	5-26
5.2.5	Java Persistence API.....	5-28
6	Standard Design Pattern	6-5
6.1	Übersicht	6-5
6.1.1	Typische technologische Beschränkungen.....	6-5
6.1.2	Einfluss auf das Design der Anwendung	6-6
6.1.3	Übersicht.....	6-6
6.2	Die GoF-Pattern	6-8
6.2.1	Erzeugungsmuster.....	6-8
6.2.2	Singleton.....	6-9
6.2.3	Factory	6-11
6.2.4	Prototype	6-14
6.2.5	Strukturmuster	6-15
6.2.6	Composite.....	6-15
6.2.7	Hüllklassen	6-16
6.2.8	Flyweight.....	6-19
6.2.9	Facade.....	6-20
6.2.10	Verhaltensmuster.....	6-21

6.2.11	Command	6-22
6.2.12	Chain of Responsibility	6-23
6.2.13	Template Method	6-24
6.2.14	Mediator	6-25
6.2.15	Observer	6-26
6.2.16	State	6-27
6.2.17	Strategy	6-28
6.2.18	Visitor	6-29
6.2.19	Interpreter, Iterator und Memento	6-29
6.3	Patterns für verteilte Anwendungen	6-30
6.3.1	Gateway	6-30
6.3.2	Mapper	6-31
6.3.3	Layer Supertype	6-31
6.3.4	Registry	6-32
6.3.5	Value Object	6-32
6.3.6	Special Case	6-34
6.3.7	Plug In	6-34
6.3.8	Service Stub	6-35
6.3.9	Record Set	6-35
6.3.10	Model View Controller	6-36
6.3.11	Page Controller	6-36
6.3.12	Front Controller	6-37
6.3.13	Template View	6-37
6.3.14	Transform View	6-38
6.3.15	Two step View	6-38
6.3.16	Application Controller	6-38
6.3.17	Remote Facade	6-39
6.3.18	Data Transfer Object	6-39
6.3.19	Client Session State	6-40
6.3.20	Server Session State	6-40
6.3.21	Database Session State	6-40
6.3.22	Optimistic Offline Lock	6-41
6.3.23	Pessimistic offline lock	6-42
6.3.24	Coarse Grained Lock	6-42
6.3.25	Implicit Lock	6-42
6.3.26	Transaktionsskript	6-43
6.3.27	Domain Model	6-43
6.3.28	Service Layer	6-43
6.3.29	Table Modul	6-44
6.3.30	Table Data Gateway	6-44

6.3.31	Row Data Gateway	6-44
6.3.32	Active Record	6-45
6.3.33	Data Mapper	6-45
6.3.34	Unit of Work	6-45
6.3.35	Identity Map	6-46
6.3.36	Lazy Load	6-46
6.3.37	Identity Field	6-46
6.3.38	Foreign Key Mapping	6-47
6.3.39	Association Table Mapping	6-47
6.3.40	Dependent Mapping	6-48
6.3.41	Embedded Value	6-48
6.3.42	Serialized Large Object	6-49
6.3.43	Single Table Inheritance	6-49
6.3.44	Class Table Inheritance	6-50
6.3.45	Inheritance Mappers	6-51
6.3.46	Concrete Table Inheritance	6-52
6.3.47	Metadata Mapping	6-52
6.3.48	Query Objekt	6-53
6.3.49	Repository	6-53
6.4	Model Driven Architecture	6-54
6.5	Das Basis-Design mit JEE-Komponenten	6-56
6.6	JEE Patterns	6-57
6.6.1	Service Locator	6-57
6.6.2	Layer Supertype für JEE	6-57
6.6.3	Business Delegate und Session Facade	6-58
6.6.4	Value List Handler	6-59
6.6.5	Value Object und Transfer Object	6-60
6.6.6	Value Object Assembler, Data Transfer Object Factory ..	6-60
6.6.7	Data Access Object	6-62
6.6.8	Die Message Facade	6-64
6.6.9	Der Service Activator	6-64
6.6.10	Aggregate Entity	6-65
6.6.11	Der Front Controller	6-65
6.6.12	Service to Worker	6-65
6.6.13	Weitere Patterns	6-65
6.7	Basis-Design unter Verwendung von JEE Patterns	6-66

