Vorlesung

# Middleware

Winter 2004/2005

MW

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

# Inhalt


A Organisatorisches . . . . . A.2
A.1 Dozent . . . . . A.2
A.2 Übungsbetreuung . . . . . A.2
A.3 Inhalt . . . . . A.3
A.4 Vorlesung . . . . . A.4
A.5 Übungen . . . . . A.6
A.6 Prüfungen, Scheine . . . . . A.7

B Überblick . . . . . B.1
B.1 Objektorientierte Programmierung . . . . . B.1
B.2 Verteilte Systeme, Verteilte Objekte und CORBA . . . . . B.2
B.3 Objektorientierte Komponentenmodelle . . . . . B.3
B.4 Verteilte Anwendungen, Web-Anwendungen . . . . . B.3
B.5 Peer-to-Peer-Architekturen . . . . . B.4
B.6 WebServices . . . . . B.4
B.7 Verteilte Betriebssysteme . . . . . B.4

C Objektorientierte Programmierung . . . . . C.1
C.1 Überblick . . . . . C.1
C.2 Literatur . . . . . C.2



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.




C.3 Motivation für das objektorientierte Paradigma . . . . . C.3
1 Ziele . . . . . C.3
C.4 Software-Design Methoden . . . . . C.4
1 Einordnung nach Booch (aus [Boo94]) . . . . . C.4
2 Klassen von Programmiersprachen . . . . . C.4
3 Top-down Structured Design (Composite Design)C.5
4 Objektorientiertes Design . . . . . C.7
C.5 Objektorientierte Programmierung . . . . . C.10
1 Definition (Grady Booch) . . . . . C.10
2 Grundbegriffe . . . . . C.11
3 Objekte & Methoden . . . . . C.12
4 Klassen . . . . . C.13
5 Vererbung . . . . . C.14
6 Vererbung in C++ . . . . . C.19
7 Dynamisches Binden . . . . . C.20
8 Statisches Binden . . . . . C.22
C.6 Fundamentale Konzepte des objektorientierten Paradigmas . . . . . C.23
1 Abstraktion . . . . . C.24
2 Kapselung . . . . . C.25
3 Modularisierung . . . . . C.27
4 Hierarchie . . . . . C.28
5 Typisierung . . . . . C.29
6 Typhierarchie . . . . . C.30
7 Polymorphismus . . . . . C.31
8 Polymorphismus in C++ . . . . . C.32
9 Typen und C++: Abstrakte Klassen . . . . . C.35
10 Typen und Java: Interfaces . . . . . C.36



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

11 Generizität (Genericity) . . . C.38
12 Generizität und C++: Templates . . . C.39
13 Generizität und Java (ab Java 5) . . . C.40
14 Nebenläufigkeit (Concurrency) . . . C.41
15 Nebenläufigkeit und Java . . . C.42
16 Persistenz . . . C.45
C.7 Objektorientierte Software-Entwicklung . . . C.47
1 Objektorientiertes Software-Engineering . . . C.48
2 Warum objektorientiertes Software-Engineering? . . . C.49
3 Phasen der Software-Entwicklung . . . C.52
C.8 Objektorientierte Analyse . . . C.56
1 Der Prozess . . . C.57
2 OOA — Anforderungsanalyse . . . C.58
3 OOA — Beispiele einer Anforderungsanlyse . . . C.59
4 OOA — Use Cases . . . C.60
5 OOA — Objekte finden . . . C.62
6 OOA — Objekte organisieren . . . C.63
7 OOA — Beschreibe Interaktionen . . . C.71
8 OOA — Struktur verfeinern . . . C.73
9 OOA - OOD? . . . C.74
C.9 Objektorientiertes Design . . . C.75
1 Phasen . . . C.76
C.10 OOA / OOD - Zusammenfassung . . . C.77
C.11 Entwurfsmuster (Design Patterns) . . . C.78
1 Beispiel: Smalltalk's Model/View/Controller . . . C.79
2 Elemente eines Entwurfsmusters . . . C.83
3 Design Pattern Space . . . C.84




Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.




D Verteilte Objekte und CORBA . . . D.1
D.1 Überblick . . . D.1
D.2 Verteilte Systeme . . . D.2
D.3 Kommunikationsmodelle . . . D.3
1 Protokollschichten nach dem ISO OSI Referenzmodell . . . D.3
2 Klassifikation . . . D.4
3 Synchrones Request-Reply Modell . . . D.5
4 Asynchrones Request-Reply Modell . . . D.6
5 Zuverlässigkeit . . . D.7
6 Remote Procedure Calls . . . D.10
7 Name-Server und Binden . . . D.15
D.4 OO Verteilte Anwendungen . . . D.16
D.5 OOP und Verteilung . . . D.17
1 Klassifikation von Interaktionsformen . . . D.17
2 explizite, orthogonale Interaktion . . . D.18
3 implizite, nicht-orthogonale Interaktion . . . D.19
4 uniforme / nicht-uniforme Interaktion . . . D.20
5 transparente / nicht-transparente Verteilung . . . D.21
6 Herausforderungen . . . D.22
D.6 Java RMI . . . D.23
1 Entfernte Objekte finden . . . D.24
2 Methodenaufruf . . . D.25
3 Parameterübergabe . . . D.26
4 Resumee . . . D.26
D.7 Middleware für verteiltes Programmieren . . . D.27




Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

D.8 CORBA — Architektur . . . . . D.28
1 Überblick . . . . . D.28
2 Literatur, URLs . . . . . D.29
3 Motivation . . . . . D.30
4 Entwurfsziele . . . . . D.31
5 OMA – Object Management Architecture . . . . . D.33
6 CORBA-Implementierungen . . . . . D.34
D.9 CORBA-Anwendungsobjekte . . . . . D.35
1 Verteilte Objekte . . . . . D.35
2 Interface Definition Language (IDL) . . . . . D.37
3 Objekte Erzeugen und Binden . . . . . D.41
D.10 Object Request Broker – ORB . . . . . D.42
1 Architektur . . . . . D.43
2 Statische Stubs . . . . . D.44
3 Interface Repository . . . . . D.46
4 Dynamic Invocation Interface (DII) . . . . . D.50
5 Dynamic Skeleton Interface (DSI) . . . . . D.52
6 Object Adaptor . . . . . D.53
7 Implementation Repository . . . . . D.54
8 Inter-ORB-Kommunikation . . . . . D.55
D.11 Portable Object Adaptor (POA) . . . . . D.56
1 Ziele . . . . . D.56
2 Terminologie . . . . . D.57
3 Erzeugung eines CORBA-Objekts . . . . . D.58
4 Alternativen zur Aktivierung von CORBA-Objekten . . . . . D.63
5 Deaktivierung und Aktivierung . . . . . D.64
6 POA-Strategien (Policies) . . . . . D.65



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.



D.12 CORBA Services . . . . . D.66
1 Naming Service . . . . . D.67
2 Life Cycle Service . . . . . D.70
3 Object Transaction Service (OTS) . . . . . D.74
4 CORBA Services - Zusammenfassung . . . . . D.78

E Java & Komponentenmodelle & Jini . . . . . E.1
E.1 Überblick . . . . . E.1
E.2 Literatur . . . . . E.2
E.3 Komponentenmodelle . . . . . E.3
1 Software Komponenten . . . . . E.3
2 Softwarekomponenten (2) . . . . . E.4
3 Philosopie . . . . . E.5
4 Beispiel für ein Builder Tool . . . . . E.6
5 Beispiel für eine Anwendung . . . . . E.7
6 Komponentenmodell . . . . . E.8
E.4 Java & Softwarekomponenten . . . . . E.9
1 Java — Ziele . . . . . E.9
2 Java — wesentliche Eigenschaften für Komponenten . . . . . E.10
3 Java-Komponentenmodelle . . . . . E.10
E.5 JavaBeans . . . . . E.11
1 Definition . . . . . E.11
2 Architektur . . . . . E.12
3 Beispiele . . . . . E.13
4 Properties . . . . . E.14
5 Events . . . . . E.17



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

6 Adapter . . . . . E.19
7 Introspection . . . . . E.20
8 JavaBeans — Summary . . . . . E.21
E.6 Jini . . . . . E.22
1 Überblick . . . . . E.22
2 wesentliche Konzepte . . . . . E.23
3 Jini Services . . . . . E.24
4 Djinn . . . . . E.25
5 Discovery . . . . . E.26
6 Leasing . . . . . E.27
7 Ereignisse (Events) . . . . . E.28
8 Sicherheit . . . . . E.28
E.7 Realistische Anwendungen für Komponenten-Software? . . . . . E.29
E.8 Software-Komponenten für realistische Anwendungen? . . . . . E.30
E.9 OSGi — Open Services Gateway Initiative . . . . . E.31
1 Überblick . . . . . E.31
2 Einsatzbeispiele . . . . . E.32
3 Architektur . . . . . E.33
4 Services . . . . . E.34
E.10 Web-Services . . . . . E.35
1 EJB-Architektur (Details später in Kapitel EJB) . . . . . E.35
2 Java Server Pages . . . . . E.36
3 JSP Beispiel . . . . . E.37
4 JSP Ziele . . . . . E.37
5 Servlets . . . . . E.38




Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.




F Microsoft & Verteilte Objekte . . . . . F.1
F.1 Überblick . . . . . F.1
F.2 Literatur . . . . . F.2
F.3 Terminology . . . . . F.3
1 OLE – Object Linking and Embedding . . . . . F.3
2 COM – Component Object Model . . . . . F.4
3 DCOM – Distributed COM . . . . . F.5
4 ActiveX . . . . . F.5
5 COM+ . . . . . F.5
F.4 COM Architecture . . . . . F.6
1 IDL – Interface Definition Language . . . . . F.6
2 Object Model . . . . . F.7
3 Process of Creation and Binding . . . . . F.9
4 Monikers . . . . . F.11
F.5 Comparison to CORBA . . . . . F.12
F.6 .NET-Architektur . . . . . F.15
1 Motivation & Ziele (1) . . . . . F.15
2 .NET-Framework — Überblick . . . . . F.18
3 Sprachunterstützung . . . . . F.19
4 Common Language Infrastructur — CLI . . . . . F.20
5 Common Language Runtime . . . . . F.21
6 Assemblies . . . . . F.22
7 Virtual Execution System — VES . . . . . F.23
8 Unmanaged / Unsafe Code . . . . . F.24
9 Application Domains . . . . . F.25
10 Programmierung verteilter Systeme . . . . . F.27




Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

G Enterprise Java Beans . . . . . G.1
G.1 J2EE . . . . . G.1
G.2 Motivation . . . . . G.2
G.3 Architektur . . . . . G.4
1 EJB-Architektur . . . . . G.8
2 Rollen von EJB . . . . . G.10
3 Unterschied zu klassischer Middleware . . . . . G.14
G.4 EJB-Grundlagen . . . . . G.20
1 Bean-Klassen . . . . . G.21
2 Interaktion mit Beans . . . . . G.22
3 Erzeugung von Beans . . . . . G.27
4 Verwaltung des Lebenszyklus . . . . . G.29
5 Bean-Interaktion mit dem Container . . . . . G.31
6 Beispiel . . . . . G.32
7 Deployment . . . . . G.36
8 Interaktion mit der Bean . . . . . G.37
G.5 Beispiel: Session-Beans . . . . . G.38
1 Zustandslose Session-Bean . . . . . G.39
2 Zustandsbehaftete Session-Bean . . . . . G.43
G.6 Einordnung . . . . . G.48


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.



H Webservices & Verteilte Anwendungen . . . . . H.1
H.1 Überblick . . . . . H.1
H.2 Problematik . . . . . H.2
1 Standards für Anwendungsinteraktion . . . . . H.2
2 Enterprise Application Integration — EAI . . . . . H.3
3 Business-to-Business Integration — B2BI . . . . . H.6
H.3 Web Services . . . . . H.7
1 Web Service Techniken . . . . . H.8
2 Politik . . . . . H.10
H.4 Web Services — Grundlagen und Standards . . . . . H.11
1 SGML - Standard für Auszeichnungssprachen . . . . . H.11
2 XML - Extensible Markup Language . . . . . H.12
3 XML-Resumee . . . . . H.24
4 Simple Object Access Protocol — SOAP . . . . . H.25
5 Web Services Description Language — WSDL . . . . . H.28
6 Universal Description, Discovery and Integration — UDDI . . . . . H.37
H.5 Web-Services in Java . . . . . H.39
H.6 Vergleich mit objektbasierter Middleware . . . . . H.41
1 Minimalanforderungen . . . . . H.42
H.7 weitere Standards und Entwicklungen . . . . . H.44


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

I P2P-Systeme . . . . . I.1
I.1 Überblick . . . . . I.1
I.2 Einführung . . . . . I.2
1 Ausgangssituation . . . . . I.2
2 Klassifikation . . . . . I.8
3 Literatur . . . . . I.9
I.3 Napster . . . . . I.10
1 Protokoll - Verbindungsaufbau . . . . . I.11
2 Protokoll - Dateisuche . . . . . I.12
3 Protokoll - Dateitransfer . . . . . I.13
4 Zusammenfassung . . . . . I.15
I.4 Gnutella . . . . . I.16
1 Grundlage . . . . . I.17
2 Verbindungsaufbau . . . . . I.18
3 Nachrichten - Erhaltung der Netzwerkstruktur . . . . . I.19
4 Dateitransfer . . . . . I.21
5 Routing - Erhaltung der Netzwerkstruktur: . . . . . I.22
I.5 Verteilte Hash-Tabellen . . . . . I.26
1 Übersicht . . . . . I.26
2 Prinzipielle Arbeitsweise . . . . . I.27
3 Eigenschaften von verteilten Hash-Tabellen . . . . . I.34
I.6 JXTA . . . . . I.35
1 Überblick . . . . . I.35
2 Überblick . . . . . I.36
3 Überblick . . . . . I.37
4 Überblick . . . . . I.38
5 Literatur . . . . . I.39



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.



6 Grundkonzepte . . . . . I.42
7 Protokolle . . . . . I.50
8 Endpoint Routing Protocol (ERP) . . . . . I.51
9 Peer Resolver Protocol (PRP) . . . . . I.53
10 Peer Rendezvous Protocol (RVP) . . . . . I.55
11 Peer Discovery Protocol (PDP) . . . . . I.59
12 Peer Information Protocol (PIP) . . . . . I.60
13 Pipe Binding Protocol (PIB) . . . . . I.61
14 Protokollstack . . . . . I.62
15 Bewertung des Protokollstack . . . . . I.63
16 Rendezvous Netzwerk der Referenzimplementierung . . . . . I.64
17 Zusammenfassung . . . . . I.66

J Design Patterns für nebenläufige und verteilte Objekte . . . . . J.1
J.1 Überblick . . . . . J.1
J.2 Literatur . . . . . J.1
J.3 Motivation . . . . . J.2
J.4 Standardprobleme . . . . . J.3
J.5 Design Patterns . . . . . J.4
1 Service Access and Configuration Patterns . . . . . J.4
2 Event Handling Patterns . . . . . J.6
3 Synchronization Patterns . . . . . J.7
4 Concurrency Patterns . . . . . J.8



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

K Betriebssystemarchitekturen . . . . . K.1
K.1 Entwicklung der Betriebssystemstrukturierung . . . . . K.1
1 Literatur . . . . . K.2
K.2 Monolithische Betriebssystemkerne . . . . . K.3
1 Vorteile . . . . . K.3
2 Nachteile . . . . . K.4
3 Gesamtablauf . . . . . K.5
K.3 Minimalkerne . . . . . K.6
1 Vorteile . . . . . K.7
2 Nachteile . . . . . K.8
3 Gesamtablauf . . . . . K.9
K.4 Objektbasierte, offene Betriebssysteme . . . . . K.10
K.5 Hardware- vs. Softwareschutzkonzepte . . . . . K.12
1 Hardware-Schutz . . . . . K.12
2 Software-Schutzkonzepte . . . . . K.13
K.6 Objektorientierte BS: JX . . . . . K.14
1 Grundkonzepte . . . . . K.14
2 Architektur . . . . . K.15
3 Inter-Domain-Kommunikation — Portale . . . . . K.19
4 Optimierungen: Fast Portals . . . . . K.24
5 Optimierungen: Memory Objekte . . . . . K.25
6 Gerätetreiber . . . . . K.30
7 Betriebssystembaukasten: Spezialsystem . . . . . K.32
8 Erweiterungsmöglichkeiten . . . . . K.36


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.