

Vorlesung

Middleware

Winter 2003/2004



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.4
- C Objektorientierte Programmierung C.1
 - C.1 Überblick C.1
 - C.2 Literatur C.2
 - 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
 - 11 Generizität (Genericity) C.38
 - 12 Generizität und C++: Templates C.39
 - 13 Nebenläufigkeit (Concurrency) C.40
 - 14 Nebenläufigkeit und Java C.41
 - 15 Persistenz C.43



	Inhalt
C.7	Objektorientierte Software-Entwicklung C.45
1	Objektorientiertes Software-Engineering C.46
2	Warum objektorientiertes Software-Engineering? C.47
3	Phasen der Software-Entwicklung C.50
C.8	Objektorientierte Analyse C.54
1	Der Prozess C.55
2	OOA — Anforderungsanalyse C.56
3	OOA — Beispiele einer Anforderungsanalyse C.57
4	OOA — Use Cases C.58
5	OOA — Objekte finden C.60
6	OOA — Objekte organisieren C.61
7	OOA — Beschreibe Interaktionen C.69
8	OOA — Struktur verfeinern C.71
9	OOA - OOD? C.72
C.9	Objektorientiertes Design C.73
1	Phasen C.74
C.10	OOA / OOD - Zusammenfassung C.75
C.11	Entwurfsmuster (Design Patterns) C.76
1	Beispiel: Smalltalk's Model/View/Controller C.77
2	Elemente eines Entwurfsmusters C.81
3	Design Pattern Space C.82
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



	Inhalt
D.9	CORBA-Anwendungsobjekte D.41
1	Verteilte Objekte D.41
2	Interface Definition Language (IDL) D.43
3	Objekte Erzeugen und Binden D.47
D.10	Object Request Broker – ORB D.48
1	Architektur D.49
2	Statische Stubs D.50
3	Interface Repository D.52
4	Dynamic Invocation Interface (DII) D.56
5	Dynamic Skeleton Interface (DSI) D.58
6	Object Adaptor D.59
7	Implementation Repository D.60
8	Inter-ORB-Kommunikation D.61
D.11	Portable Object Adaptor (POA) D.62
1	Ziele D.62
2	Terminologie D.63
3	Erzeugung eines CORBA-Objekts D.64
4	Alternativen zur Aktivierung von CORBA-Objekten D.69
5	Deaktivierung und Aktivierung D.70
6	POA-Strategien (Policies) D.71
D.12	CORBA Services D.72
1	Naming Service D.73
2	Life Cycle Service D.76
3	Object Transaction Service (OTS) D.80
4	CORBA Services - Zusammenfassung D.84



	Inhalt
2	Klassifikation D.4
3	Rendezvous-Modell D.6
4	Synchrones Request-Reply Modell D.7
5	Asynchrones Request-Reply Modell D.8
6	Zuverlässigkeit D.9
7	Remote Procedure Calls D.16
8	Name-Server und Binden D.21
D.4	OO Verteilte Anwendungen D.22
D.5	OOP und Verteilung D.23
1	Klassifikation von Interaktionsformen D.23
2	explizite, orthogonale Interaktion D.24
3	implizite, nicht-orthogonale Interaktion D.25
4	uniforme / nicht-uniforme Interaktion D.26
5	transparente / nicht-transparente Verteilung D.27
6	Herausforderungen D.28
D.6	Java RMI D.29
1	Entfernte Objekte finden D.30
2	Methodenaufruf D.31
3	Parameterübergabe D.32
4	Resume D.32
D.7	Middleware für verteiltes Programmieren D.33
D.8	CORBA — Architektur D.34
1	Überblick D.34
2	Literatur, URLs D.35
3	Motivation D.36
4	Entwurfsziele D.37
5	OMA – Object Management Architecture D.39
6	CORBA-Implementierungen D.40



	Inhalt
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	Philosophie 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
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



	Inhalt
4	Djinn E.25
5	Discovery E.26
6	Leasing E.27
7	Ereignisse (Events) E.28
8	Sicherheit E.28
E.7	Enterprise Java Beans E.29
1	Architektur E.29
2	Java Server Pages E.30
3	JSP Beispiel E.31
4	JSP Ziele E.31
5	Servlets E.32
6	EJB Container E.33
7	EJB Bean-Typen E.34
8	Elemente einer EJB-Bean E.35
9	EJB Contracts: Client-View-Contract E.36
10	EJB Contracts: Component Contract E.37
E.8	Realistische Anwendungen für Komponenten-Software? E.38
E.9	Software-Komponenten für realistische Anwendungen? E.39

	Inhalt
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



	Inhalt
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 Infrastructure — 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

	Inhalt
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-Resume 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

