Objektorientierte Konzepte in der Betriebsprogrammierung

Vorlesungsmanuskript

SS 1997

Jürgen Kleinöder

Friedrich-Alexander-Universität Erlangen-Nürnberg
Institut für Mathematische Maschinen und Datenverarbeitung IV
Lehrstuhl für Betriebssysteme

Alle Unterlagen der Vorlesung und Übungen zu “Objektorientierte Konzepte in der Betriebsprogrammierung” sind ausschließlich zur Verwendung im Rahmen der Lehre an der Friedrich-Alexander-Universität Erlangen-Nürnberg bestimmt.

Einsatz, sowie Reproduktion jeder Art der Unterlagen oder Teilen davon, insbesondere zu Lehrzwecken, außerhalb der Universität Erlangen-Nürnberg ist nicht gestattet!

© 1993, 1994, 1995, 1996, 1997 Universität Erlangen-Nürnberg, IMMD IV
© 1993, 1994, 1995, 1997 Dr. Jürgen Kleinöder - Univ. Erlangen
© 1997 Dr. Franz J. Hauck - Univ. Erlangen
© 1993, 1994, 1995 Thomas Riechmann, Stephan Reitzner - Univ. Erlangen
© 1997 Michael Golm, Uwe Rastofer - Univ. Erlangen
© 1994, 1995 Sun Microsystems

Inhalt

Objektorientierte Programmierung 0.2
Verteiltes Programmieren mit CORBA 0.3
Betriebssystemarchitekturen 0.4
Java — OO Sprache + virtuelle Maschine für Objekte 0.4
Objektorientierte Betriebssystemarchitekturen - Konzepte und Forschungsprojekte 0.5

Objektorientierte Programmierung 1.1
Überblick 1.1
Literatur 1.1
Motivation für das objektorientierte Paradigma 1.2
Ziele 1.2
Software-Design Methoden 1.3
Einordnung nach Booch (aus [Boo94]) 1.3
Top-down structured design (composite design) 1.4
Data-driven design 1.5
Object-oriented design 1.5
Entstehung des Objektmodells 1.8
Generationen von Programmiersprachen ([Boo94]) 1.8
Objektorientierte Programmierung 1.9
Definition (Grady Booch [Boo94]) 1.9
Objekte & Methoden 1.10
Klassen 1.11
Objekte und Klassen in C++ 1.12
Methoden in C++ 1.13
Instantiierung in C++ 1.14
Konstruktoren 1.16
Objekte und Klassen in Java 1.17
Vererbung 1.18
Vererbung (2) 1.19
Vererbung in C++ 1.20
Basiskonzepte des OO Paradigmas 1.23
Abstraktion 1.24
Kapselung 1.25
Abstrakter Datentyp 1.26
Modularisierung 1.29
Hierarchie 1.30

© Universität Erlangen Nürnberg, IMMD IV, 1997, *Jürgen Kleinöder*

Inhalt

Typisierung 1.31
Typhierarchie 1.32
Polymorphismus 1.33
Polymorphismus in C++ 1.34
Dynamisches Binden in C++ 1.38
Dynamisches Binden in Java 1.40
Abstrakte Klassen in C++ 1.41
Abstrakte Klassen in Java 1.42
Interfaces in Java 1.43
Generizität (Genericity) 1.45
Generizität — Templates in C++ 1.46
Nebenläufigkeit (Concurrency) 1.48
Nebenläufigkeit in Java 1.49
Persistenz 1.52
OOA & OOD 1.55
Literatur 1.55
UML-Notation(1): Klassendiagramme 1.57
UML-Notation(2): Kollaborationsdiagramme 1.58
Design Patterns 1.59

Verteiltes Programmieren mit CORBA 2.1
OOP und Verteilung 2.2
Middleware für verteiltes Programmieren 2.4
Motivation für CORBA 2.5
Überblick über die Architektur 2.6
Anwendungsobjekte 2.8
Object Request Broker – ORB 2.9
IDL Schnittstellen 2.10
IDL und das Binden 2.11
Services 2.12
Common Facilities 2.13
CORBA Implementierung 2.14
Der ORB und seine zentrale Komponenten 2.15
Aufgaben 2.16
Statische Stubs 2.17
Stuberzeugung 2.18
Interface Repository 2.19

© Universität Erlangen Nürnberg, IMMD IV, 1997, *Jürgen Kleinöder*

Inhalt

DII – Dynamic Invocation Interface 2.23
DSI – Dynamic Skeleton Interface 2.25
Object Adaptor 2.26
BOA – Basic Object Adaptor 2.27
Implementation Repository 2.30
Inter-ORB Kommunikation 2.31
CORBA Services 2.32
Namensdienst 2.32
Life Cycle Service 2.35

Betriebssystemarchitekturen 3.1
Entwicklung der Betriebssystemstrukturierung 3.1
Literatur 3.1
Monolithische Betriebssystemkerne 3.2
Vorteile 3.2
Nachteile 3.3
Minimalkerne 3.5
Vorteile 3.5
Nachteile 3.6
Objektbasierte, offene Betriebssysteme 3.8
Anforderungen an zukünftige Betriebssysteme 3.10

Spring 4.1
Literatur 4.1
Überblick 4.2
Motivation 4.3
Ziele 4.4
Anwendungssicht 4.5
Systemarchitektur 4.6
Programmierung & Schnittstellen 4.7
Namensraum-Konzepte 4.9
Schutzmechanismen 4.12
Kern-Architektur 4.14
Objektinteraktion über Adreßraumgrenzen 4.15
Threads 4.17
Ortstransparente Methodenaufrufe 4.18

© Universität Erlangen Nürnberg, IMMD IV, 1997, *Jürgen Kleinöder*

Inhalt

Doors 4.19

Java 5.1
Überblick 5.1
Eigenschaften von Java 5.2
Aus dem Hochglanzprospekt … 5.2
Anfänge des Java-Projekts 5.3
Design-Ziele 5.4
Java-Architektur 5.5
Java — Sprache 5.6
Java Environment 5.7
Java Compiler 5.8
Java Plattformen 5.10
Klassenbibliotheken = Java Packages 5.10
Java Core API 5.11
Java Standard Extension API 5.12
Java Virtual Machine 5.13
Basissysteme für die Java-Plattform 5.16
WWW-Browser 5.16
Java Interpreter als Programm auf einem herkömmlichen BS 5.16
JavaOS 5.16
JavaChips 5.17
Java Plattform — Überblick 5.18
JavaOS Architektur 5.19
Dynamisches Laden und Binden 5.20
Sicherheitsaspekte 5.21
Unterstützung von Nebenläufigkeit 5.23
Java im verteilten System 5.24

Objektorientierte BS — Beispiel Choices 6.1
Überblick 6.1
Literatur 6.1
Frameworks 6.2
Hierarchische Frameworks 6.5
Framework-Dokumentation 6.6

© Universität Erlangen Nürnberg, IMMD IV, 1997, *Jürgen Kleinöder*

Inhalt

Choices-Überblick 6.10
Choices-Framework 6.11
- Komponenten 6.11
- Beziehungen 6.11
- Verfeinerung (Beispiel) 6.12
- Kontrollfluß (Beispiel) 6.13

Prozeßverwaltung 6.14
- Komponenten 6.14
- Klassenhierarchien 6.15
- Klassenbeziehungen 6.16
- Prozeßumschaltung 6.17

Speicherverwaltung 6.19
- Komponenten 6.19
- Klassenhierarchien 6.20
- Klassenbeziehungen 6.21
- Kontrollfluß (Beispiel page-fault) 6.22
- MemoryObjects 6.23

System-Schnittstelle 6.25
- Proxy-Objekte 6.26
- Beispiel 6.27
- Erweiterungen für C++ 6.28

© Universität Erlangen Nürnberg, IMMD IV, 1997, *Jürgen Kleinöder*