Object-Oriented Concepts in Distributed Systems

Contents

A Organizational Topics . . . A.1
A.1 Lecture . . . A.1
A.2 Exercise classes . . . A.5

B Overview . . . B.1
B.1 Object-Oriented Programming . . . B.1
B.2 Distributed Systems . . . B.2
B.3 Programming Distributed Systems with CORBA . . . B.3
B.4 Java — OO Language + Virtual Machine for Objects . . . B.4
B.5 Frameworks . . . B.5
B.6 Research on Distributed Object-Oriented Systems . . . B.5

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

C Object-oriented Programming . . . C.1
C.1 Overview . . . C.1
C.2 References . . . C.2
C.3 Motivation for the OO Paradigm . . . C.4
1 Goals . . . C.4
C.4 Software-Design Methods . . . C.6
1 Classification [Boo94] . . . C.6
2 Top-Down Structured Design (Composite Design) . . . C.7
3 Data-driven Design . . . C.9
4 Object-oriented Design . . . C.10
C.5 The Evolution of the Object Model . . . C.13
1 Generations of Programming Languages . . . C.13
C.6 Object-oriented Programming . . . C.15
1 Definition (Grady Booch) . . . C.15
2 Basic Terms . . . C.16
3 Objects & Methods . . . C.17
4 Classes . . . C.18
5 Objects and Classes in C++ . . . C.19
6 Methods in C++ . . . C.20
7 Instantiation in C++ . . . C.21
8 Objects and Classes in Java . . . C.24
9 Inheritance . . . C.25
10 Inheritance in C++ . . . C.31
11 Dynamic Binding . . . C.34

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

C.7 Basic Concepts of the OO Paradigm . . . C.36
1 Abstraction . . . C.37
2 Encapsulation . . . C.38
3 Abstract Data Type . . . C.39
4 Modularity . . . C.42
5 Hierarchy . . . C.43
6 Typing . . . C.44
7 Type Hierarchy . . . C.45
8 Polymorphism . . . C.46
9 Polymorphism in C++ . . . C.47
10 Types & C++ — Abstract Classes . . . C.53
11 Types & Java — Abstract Classes . . . C.54
12 Types & Java — Interfaces . . . C.55
13 Genericity . . . C.57
14 Genericity & C++: Templates . . . C.58
15 Concurrency . . . C.60
16 Concurrency & Java . . . C.61
17 Persistence . . . C.63
C.8 OOA & OOD . . . C.66
1 Overview . . . C.66
2 OOAD Methods . . . C.67
3 OOA — Process . . . C.68
4 OOD . . . C.69
5 UML Notation(1): Class Diagrams . . . C.70
6 UML-Notation(2): Collaboration Diagrams . . . C.71
7 Design Patterns . . . C.72

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

D Distributed Systems . . . D.1
D.1 Definition and Motivation . . . D.1
1 Motivation . . . D.2
2 Advantages . . . D.4
3 Disadvantages . . . D.6
D.2 Taxonomy . . . D.8
1 Multiprocessors . . . D.10
2 Multicomputers . . . D.12
3 Network Operating Systems . . . D.14
4 True Distributed Systems . . . D.16
D.3 Communication Models . . . D.17
1 Protocol layers according to the ISO OSI reference model . . . D.17
2 Classification . . . D.20
3 Rendezvous Model . . . D.22
4 Synchronous Request-Reply Model . . . D.23
5 Asynchronous Request-Reply Model . . . D.24
6 Reliability . . . D.25
7 Remote Procedure Calls . . . D.32
8 Name Server and Binding . . . D.37
9 Group Communication . . . D.38

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

D.4 Selected Problems of Distributed Systems . . . D.43
1 Logical Clocks . . . D.45
2 Clock Synchronization . . . D.51
3 Vector Time . . . D.52
4 Mutual Exclusion . . . D.54
5 Election Algorithms . . . D.60
6 Deadlock Detection . . . D.60
7 Distributed Garbage Collection . . . D.61
8 Echo Algorithms . . . D.61
D.5 Object-Based Distributed Systems . . . D.62
1 Whole Object Approach . . . D.63
2 Fragmented Object Approach . . . D.66

E Distributed Objects in CORBA . . . E.1
E.1 Middleware Approach . . . E.2
E.2 CORBA Overview . . . E.3
1 Design Goals . . . E.3
2 Architectural Overview . . . E.5
3 CORBA Implementations . . . E.7
E.3 Application Objects . . . E.8
1 Distributed Object . . . E.8
2 Distributed Object (2) . . . E.9
3 Interface Definition Language (IDL) . . . E.10
4 Process of Creation and Binding . . . E.14

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

E.4 Object Request Broker – ORB . . . E.15
1 Architecture . . . E.16
2 Static Stubs . . . E.17
3 Interface Repository . . . E.19
4 Dynamic Invocation Interface (DII) . . . E.23
5 Dynamic Skeleton Interface (DSI) . . . E.25
6 Object Adaptor . . . E.26
7 Basic Object Adaptor (BOA) . . . E.27
8 Implementation Repository . . . E.30
9 Inter-ORB Communication . . . E.31
E.5 CORBA Services . . . E.32
1 Naming Service . . . E.33
2 Life Cycle Service . . . E.36

F OLE — COM & DCOM . . . F.1
F.1 References . . . F.1
F.2 Overview . . . F.2
F.3 OLE Architecture . . . F.3
F.4 COM IDL & COM ODL . . . F.4
1 IDL . . . F.4
2 ODL . . . F.5
F.5 COM: OLE's Object Bus . . . F.6
1 !!! "COM objects" are not objects . . . F.6
2 COM server . . . F.7
3 COM Client/Server Interaction . . . F.8
F.6 COM/OLE Object Services . . . F.9

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

F.7 OLE Automation . . . F.10
F.8 OLE Uniform Data Transfer . . . F.10
F.9 OLE: Structured Storage . . . F.11
F.10 DCOM . . . F.12

G Frameworks . . . G.1
G.1 Overview . . . G.1
G.2 References . . . G.2
G.3 Frameworks — What, How & Why . . . G.3
1 Classes and Class Libraries . . . G.3
2 Frameworks (What) . . . G.4
3 Frameworks (How) . . . G.6
4 Example . . . G.7
5 Benefits . . . G.8
G.4 Types of Frameworks . . . G.9
1 Application Frameworks . . . G.9
2 Domain Frameworks . . . G.9
3 Support Frameworks . . . G.9
G.5 CORBA & Frameworks . . . G.10
G.6 Java Frameworks . . . G.12

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

H Java & Component Models . . . H.1
H.1 Overview . . . H.1
H.2 Component Models . . . H.2
1 What is a Software Component? . . . H.2
2 Rival Component Architectures . . . H.3
3 Philosophy . . . H.4
H.3 Java — Design Goals? . . . H.5
H.4 Java — Key Properties for Components . . . H.5
H.5 JavaBeans . . . H.6
1 Definition . . . H.6
2 Beans — Architecture . . . H.7
3 Example Beans . . . H.8
4 Properties . . . H.9
5 Events . . . H.12
6 Adaptors . . . H.14
7 Introspection . . . H.15

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