

# B Überblick

---

## B.1 Objektorientierte Programmierung

---

- Motivation
- OOP — grundlegende Begriffe
- Fundamentale Konzepte
  - ◆ Abstraction
  - ◆ Hierarchie
  - ◆ Nebenläufigkeit
  - ◆ Kapselung
  - ◆ Typisierung
  - ◆ Persistenz
  - ◆ Modularisierung
  - ◆ Polymorphismus
- Objektorientierte Softwareentwicklung (OOA/OOD)
- Design Patterns und ihre Bedeutung für verteilte Systeme

## B.2 Verteilte Systeme und Verteilte Objekte

---

- Grundlagen verteilter Systeme (Remote Procedure Call)
- OOP und verteilte Anwendungen  
(Explizite vs. implizite Kommunikation, Transparenz der Verteilung)
- Java RMI
- Design Patterns für nebenläufige und verteilte Anwendungen

## B.3 Middleware am Beispiel CORBA

---

- Motivation und Architekturüberblick
- Der Object Request Broker (ORB)  
(IDL, Remote invocation, Dynamic invocation, Komponenten des ORB)
- Portable Object Adaptor
- CORBA Services

## B.4 Objektorientierte Komponentenmodelle

---

- Komponentenmodelle (Grundkonzept, Übersicht)
- JavaBeans - ein Komponentenmodell für Java (Architektur, Properties, Events, Introspection)
- Spring Framework
- Jini - Middleware und ein verteiltes Komponentenmodell

## B.5 Plattformen für verteilte Dienste

---

- Verteilte Anwendungen, Web-Anwendungen
  - ◆ Entstehung verteilter Anwendungen
  - ◆ Application Server
  - ◆ Das Web als Schnittstelle zu verteilten Anwendungen
- .NET, Enterprise Java Beans, OSGi

## B.6 Peer-to-Peer-Architekturen

---

- JXTA: Middleware für P2P

## B.7 WebServices

---

- viele Standards, wenig Neues — kritischer Überblick

## B.8 Grid-Architekturen

---

- OGSA: (WebServices+P2P)++???

## B.9 Verteilte Betriebssysteme

---

- JX - unser Java-Betriebssystem