

Verteidigung Großer Beleg

„Enterprise JavaBeans und
JavaBeans“

Sean Eikenberg

Gliederung

- Präsentation der Aufgabenstellung
- Wiederholung des Zwischenberichtergebnisses
- Architektur der Zwischenschicht
- Architektur des Generators
- Nachweis der Funktionstüchtigkeit (Zwischenschicht/Generator)
- Zusammenfassung/Ausblick auf EJB 2.0

Aufgabenstellung

- Zusammenhang zwischen EJB und JavaBeans ausarbeiten.
 - Mögliche Vereinigung beider Technologien diskutieren.
-
- Entwurf einer Zwischenschicht, die die Integration von EJB innerhalb des JavaBeans-Komponentenmodells ermöglicht.
 - Implementierung und Nachweis eines Zwischenschichtgenerators.

Gliederung

- Präsentation der Aufgabenstellung
- Wiederholung des Zwischenberichtergebnisses
- Architektur der Zwischenschicht
- Architektur des Generators
- Nachweis der Funktionstüchtigkeit (Zwischenschicht/Generator)
- Zusammenfassung/Ausblick auf EJB 2.0

Abgrenzung: EJB - JavaBeans

- JavaBeans unterstützt keine Containerdienste (Sicherheit, Transaktionen, Verteilung).
- EJB unterstützt nicht freien Ressourcenzugriff, Ereignisverarbeitung oder visuelle Komponenten.
- Grundsätzlich verschiedene Komponentenschnittstellen.
- Fazit: EJB ist keine direkte Erweiterung von JavaBeans!

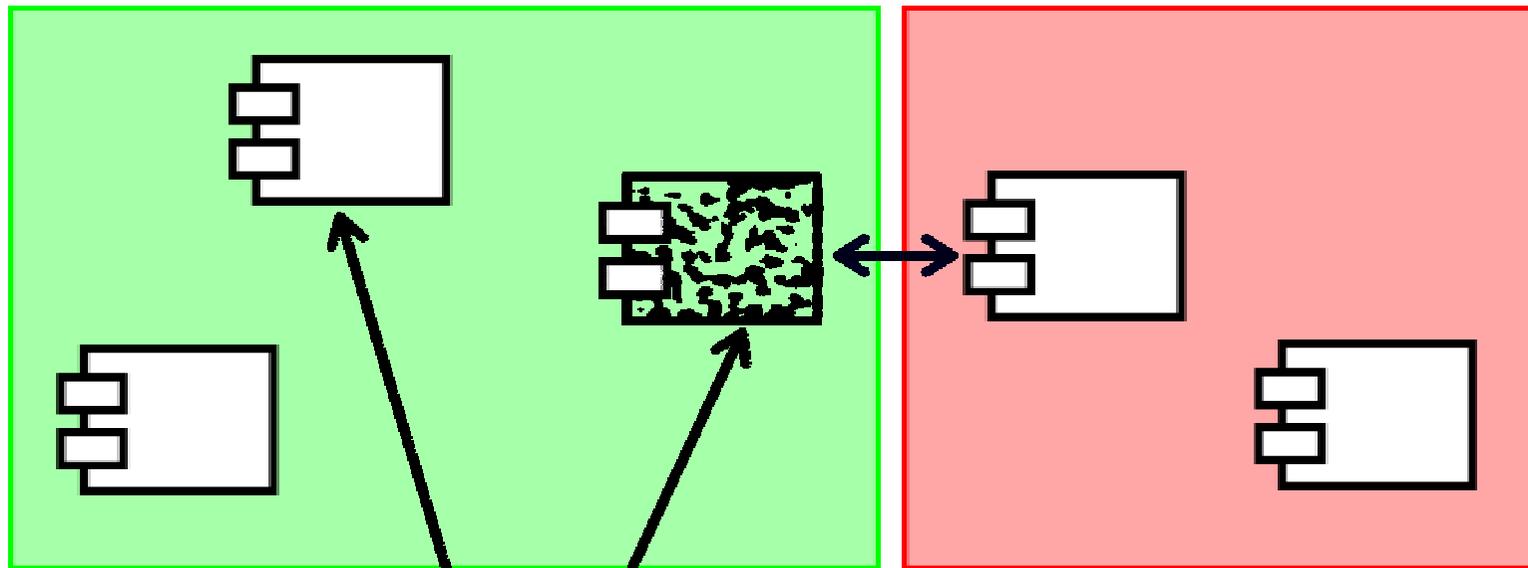
Vereinigungsmodelle

- Vier Vereinigungsmodelle:
 - „EJB & visuelle Anwendungserstellung“
 - „Integratives Komponentenmodell“
 - „Übersetzte JavaBeans“
 - „EJB-Adapter“ (früher: „EJB-Proxies“)
- Ausgangsbasis für die Zwischenschicht ist das Modell „EJB-Adapter“

„EJB-Adapter“

JavaBeans

EJB

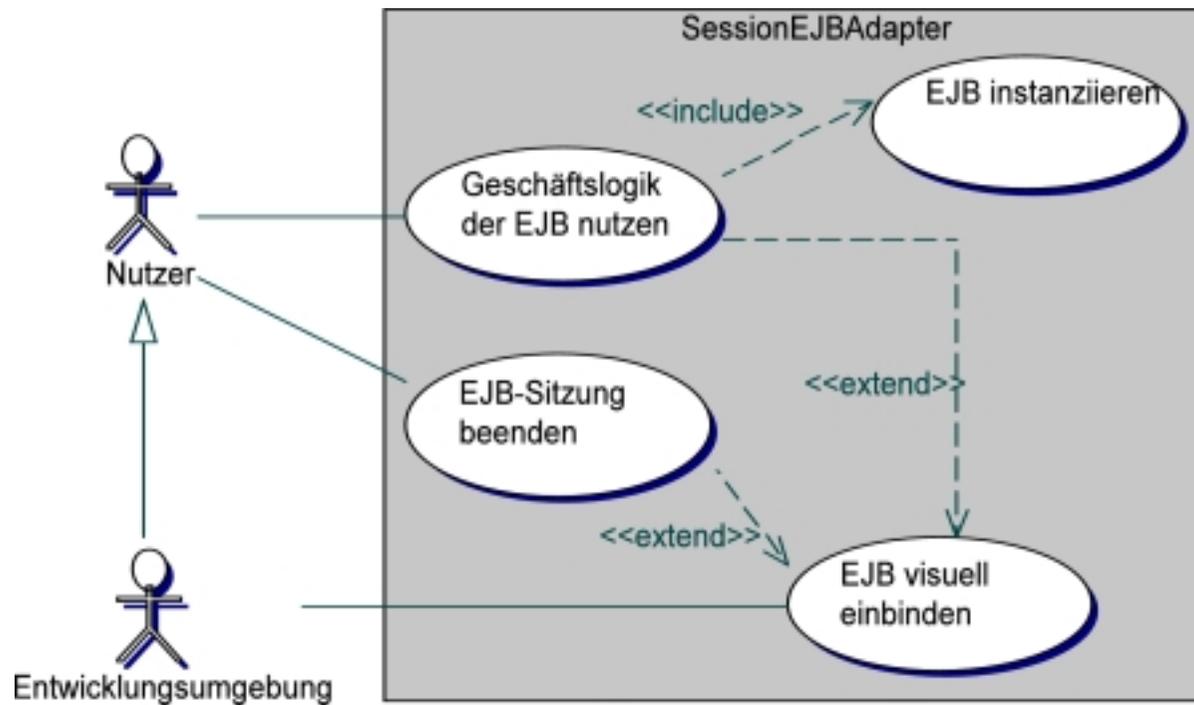


Nutzer

Gliederung

- Präsentation der Aufgabenstellung
- Wiederholung des Zwischenberichtergebnisses
- Architektur der Zwischenschicht
- Architektur des Generators
- Nachweis der Funktionstüchtigkeit (Zwischenschicht/Generator)
- Zusammenfassung/Ausblick auf EJB 2.0

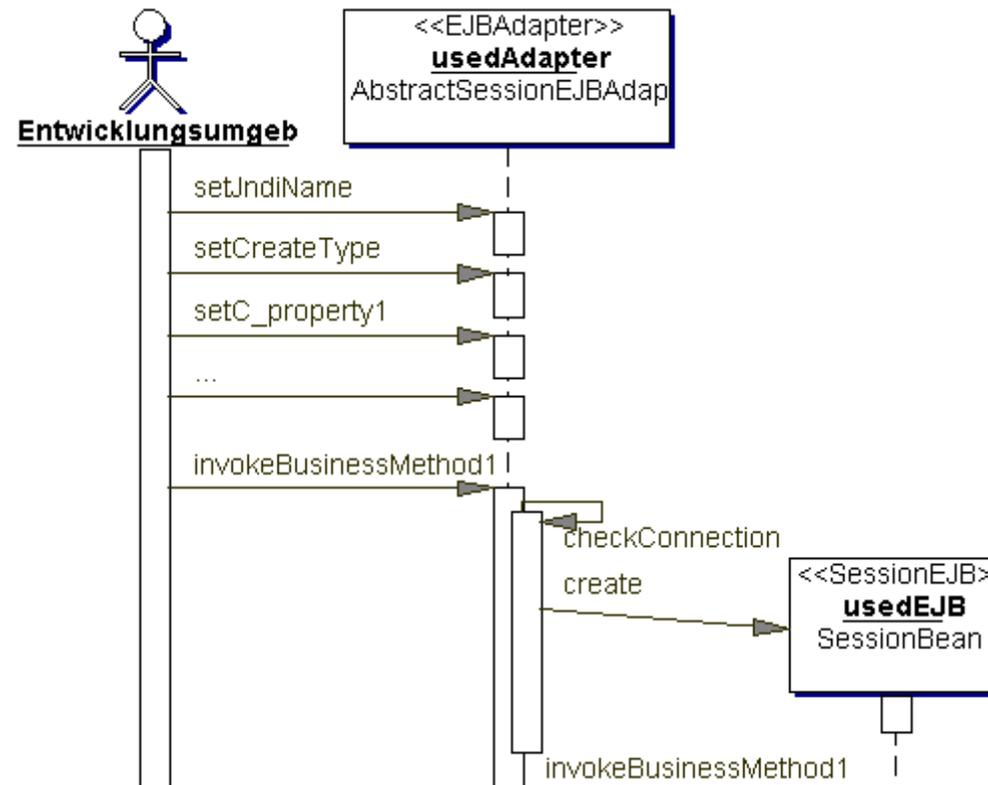
Anwendungsfälle: SessionEJBAdapter



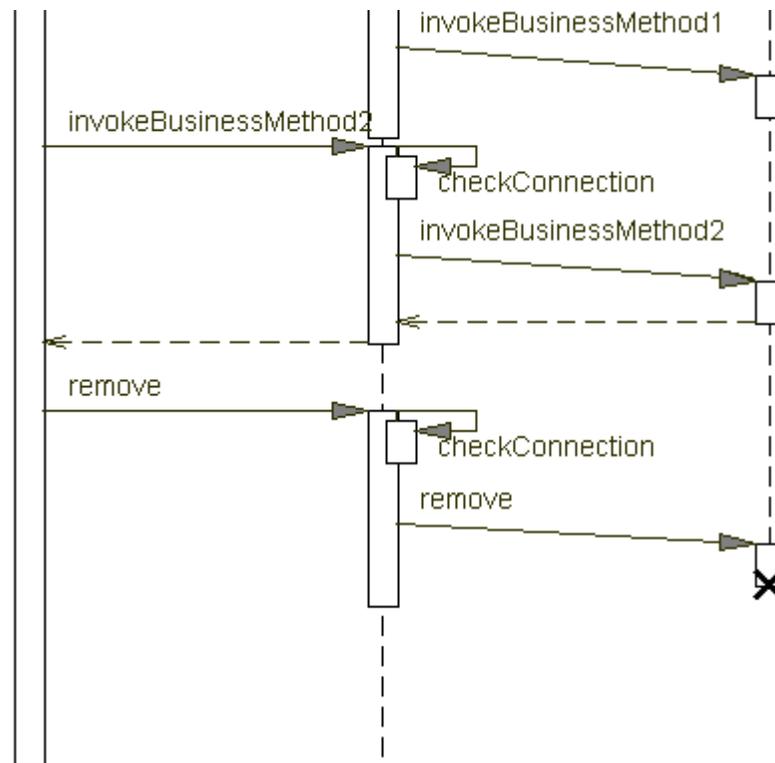
Einschub: Schnittstellenproblematik

- Abbildung von 2 EJB-Schnittstellen auf eine JavaBeans-Schnittstelle!
- „Visuell“ gut nutzbare JavaBeans sind charakterisiert durch:
 - „properties“
 - parameterlose Methoden
- Einführung von: create-/find-properties, „create-/find-type“ & parameterlose create()-, find()- und multifind()-Methoden.
- Prinzip der Modellintegrität!

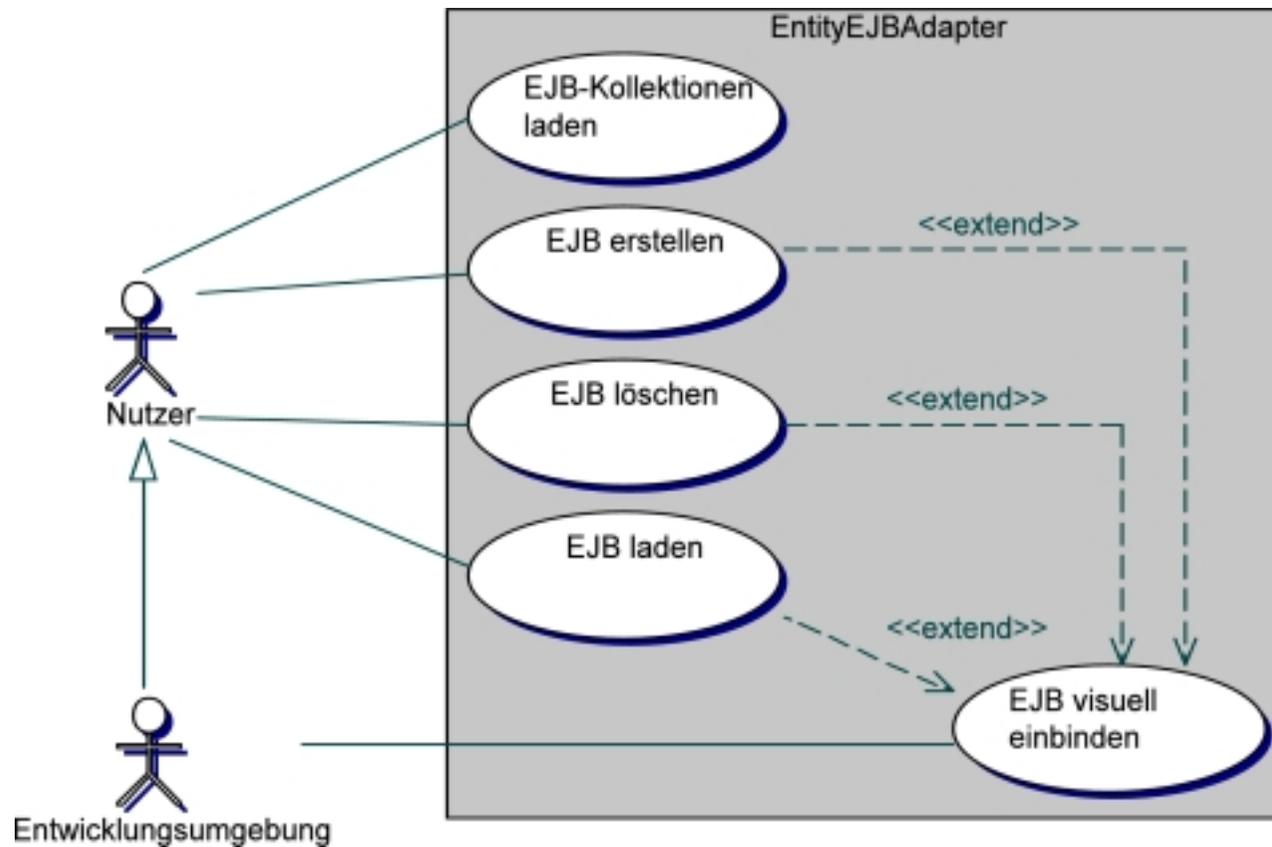
Szenario: SessionEJBAdapter



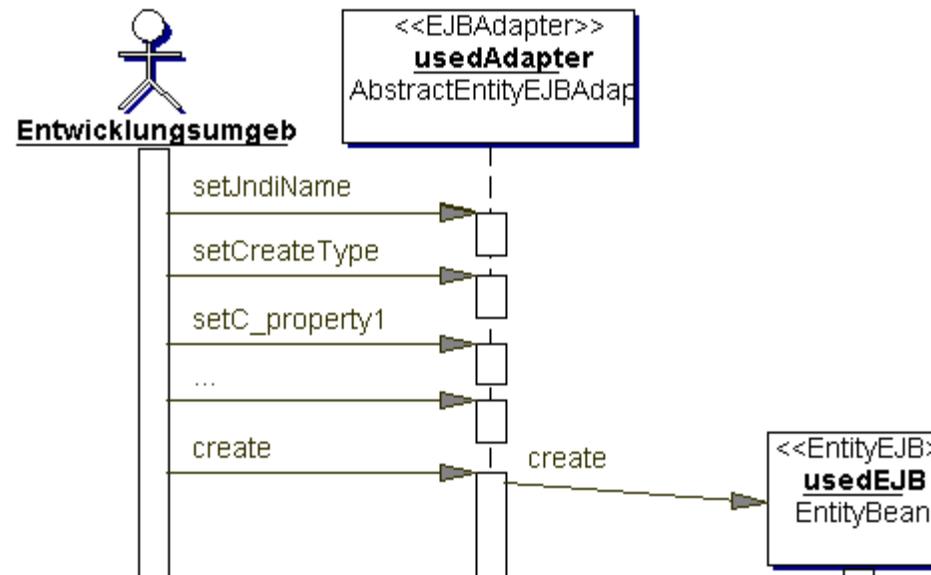
Szenario: SessionEJBAdapter



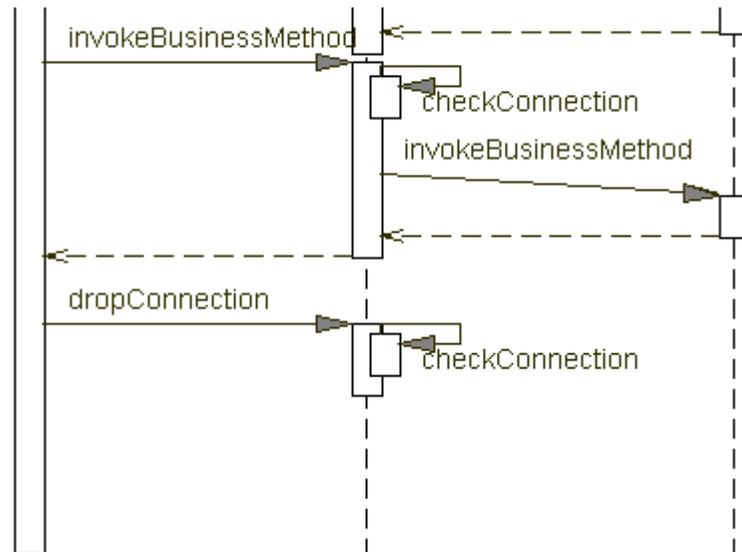
Anwendungsfälle: EntityEJBAdapter



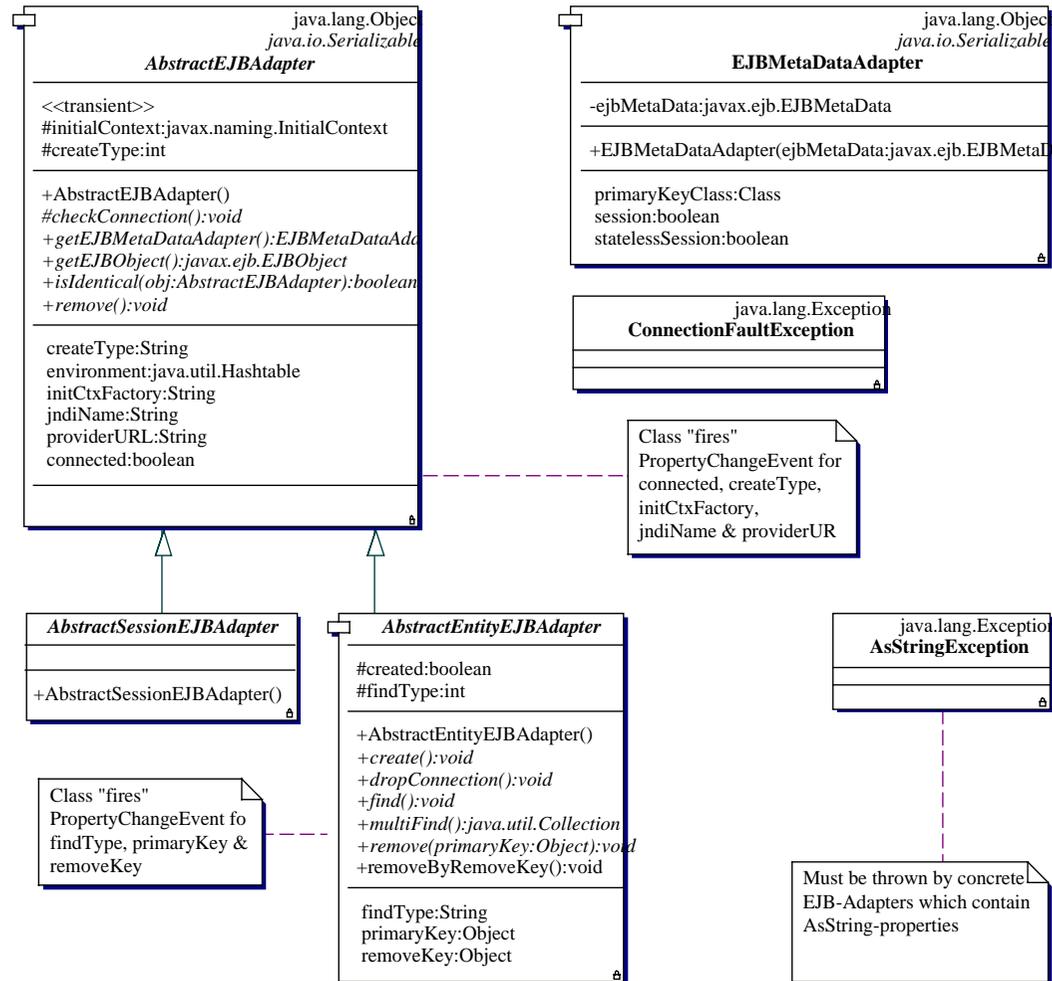
Szenario: EntityEJBAdapter



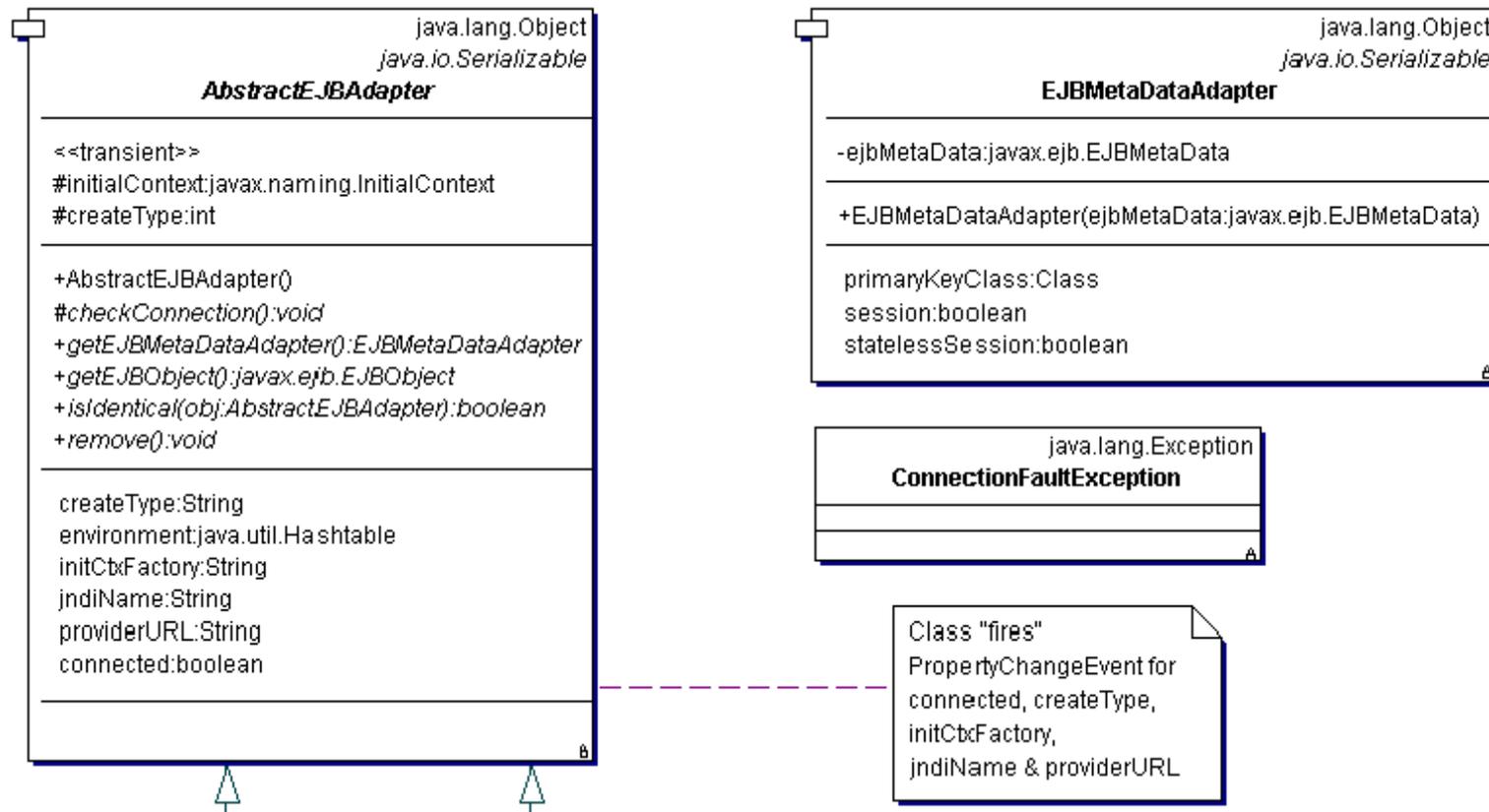
Szenario: EntityEJBAdapter



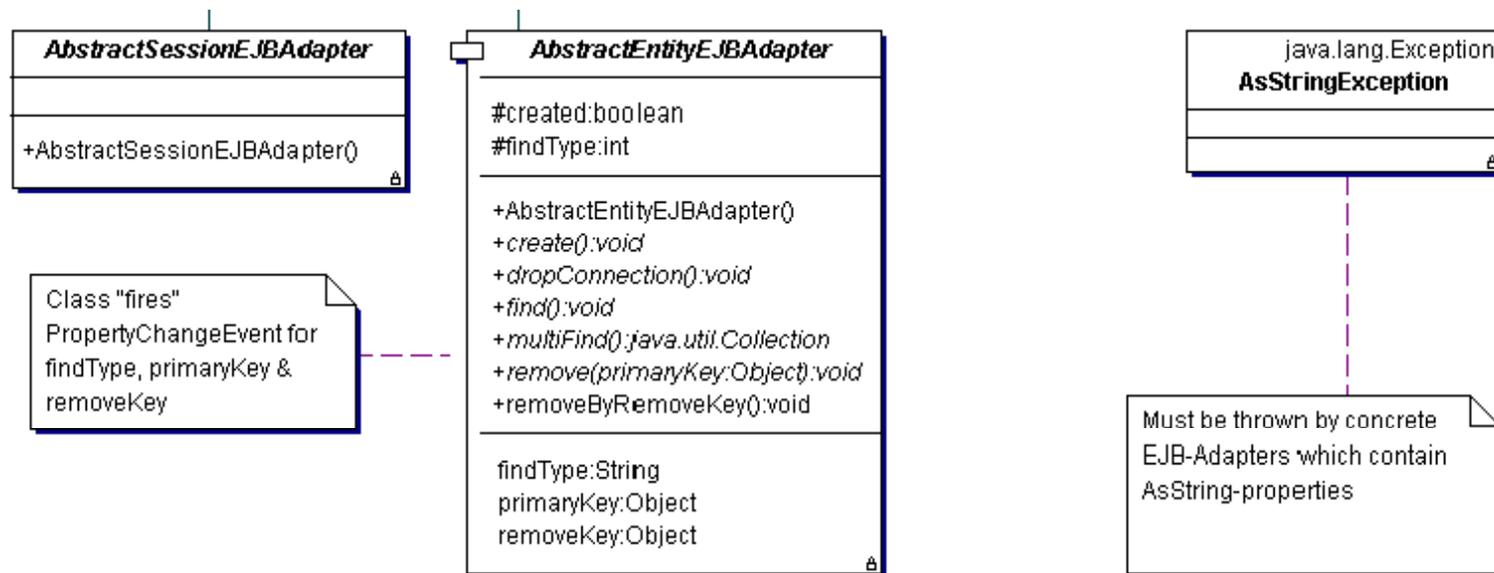
Basis Architektur - Übersicht



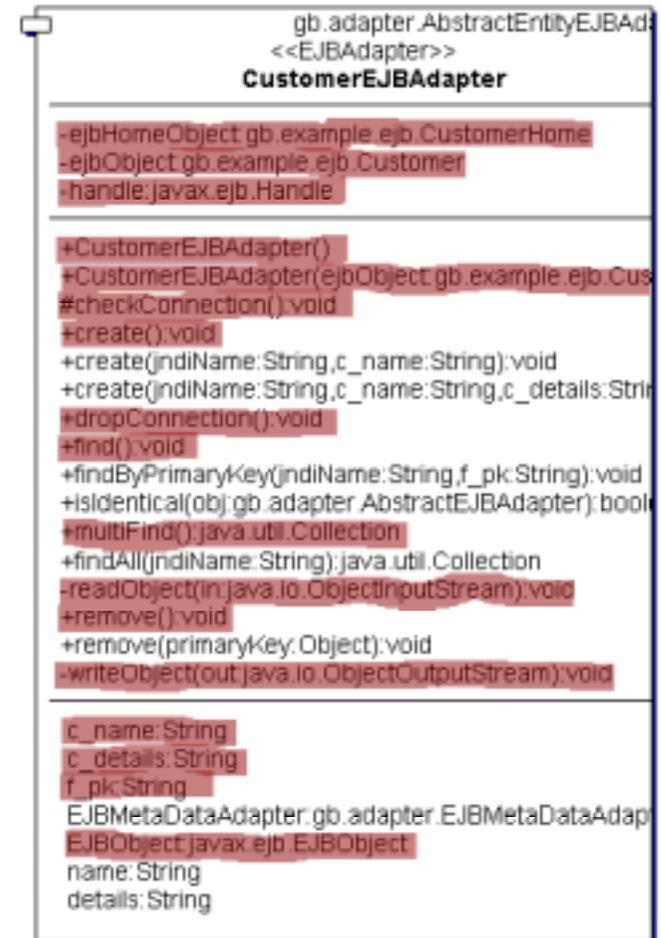
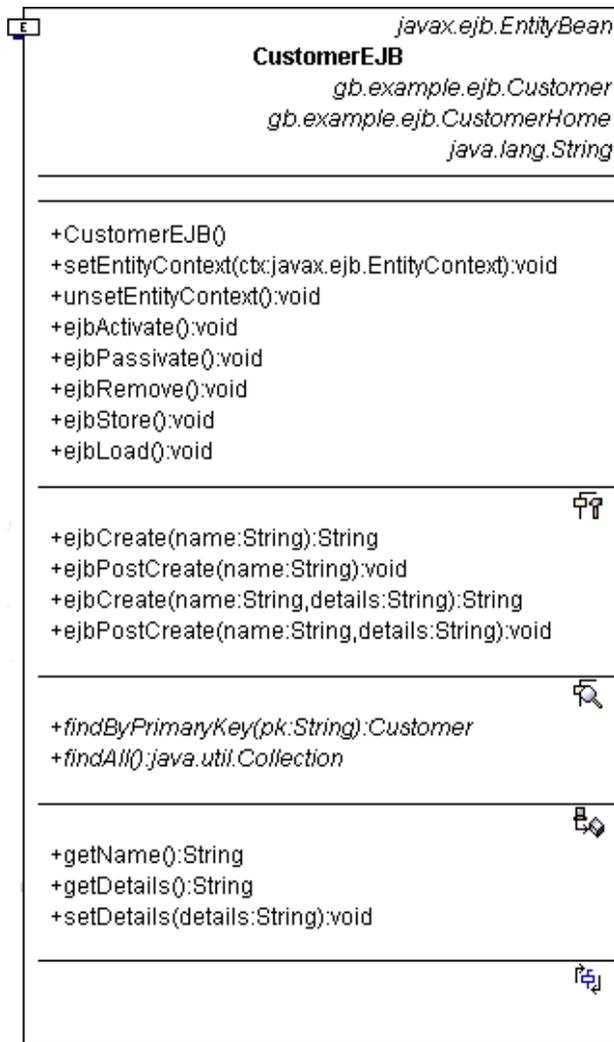
Basis Architektur (oben)



Basis Architektur (unten)



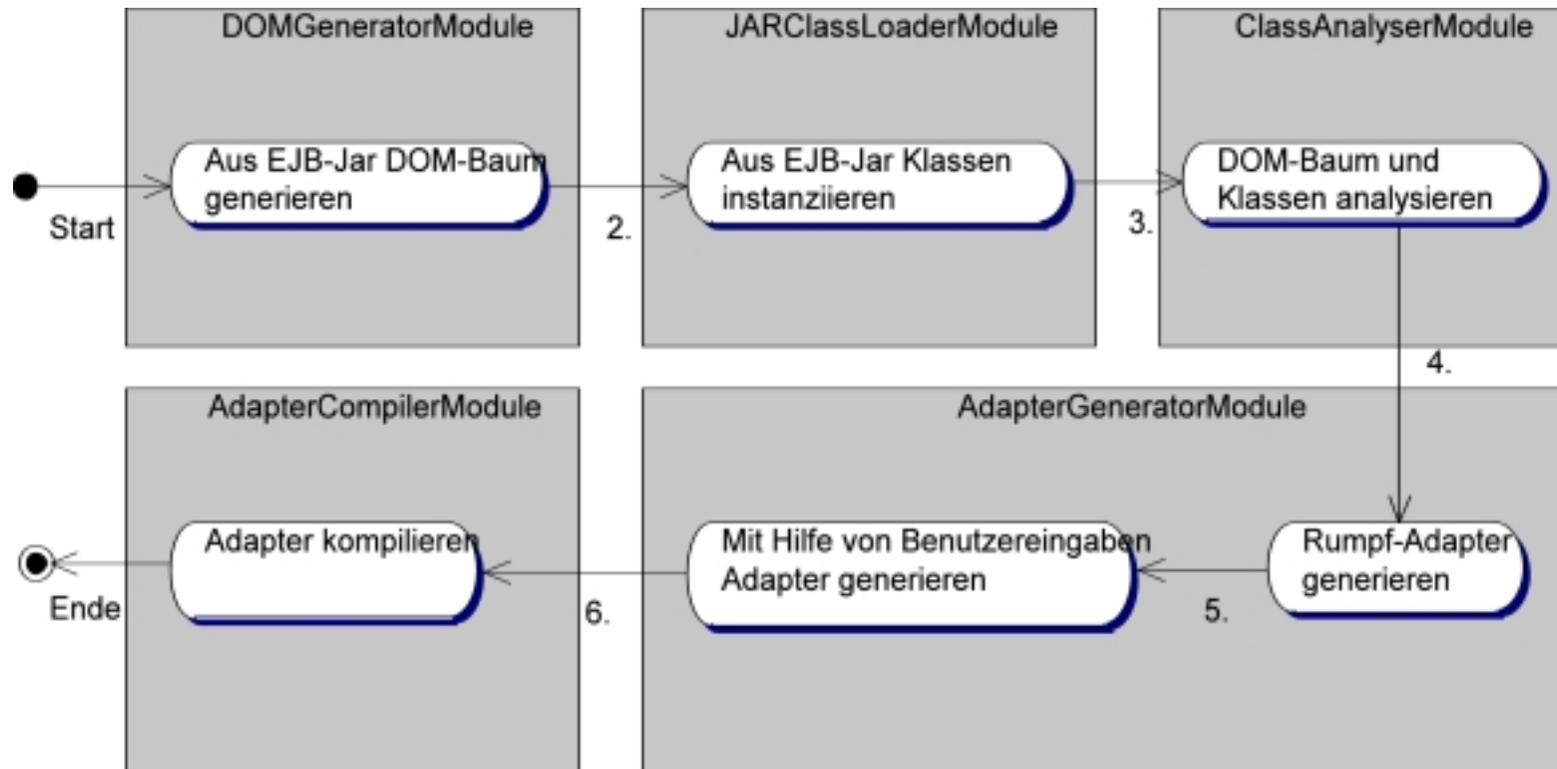
Beispiel: EJB-Adapter



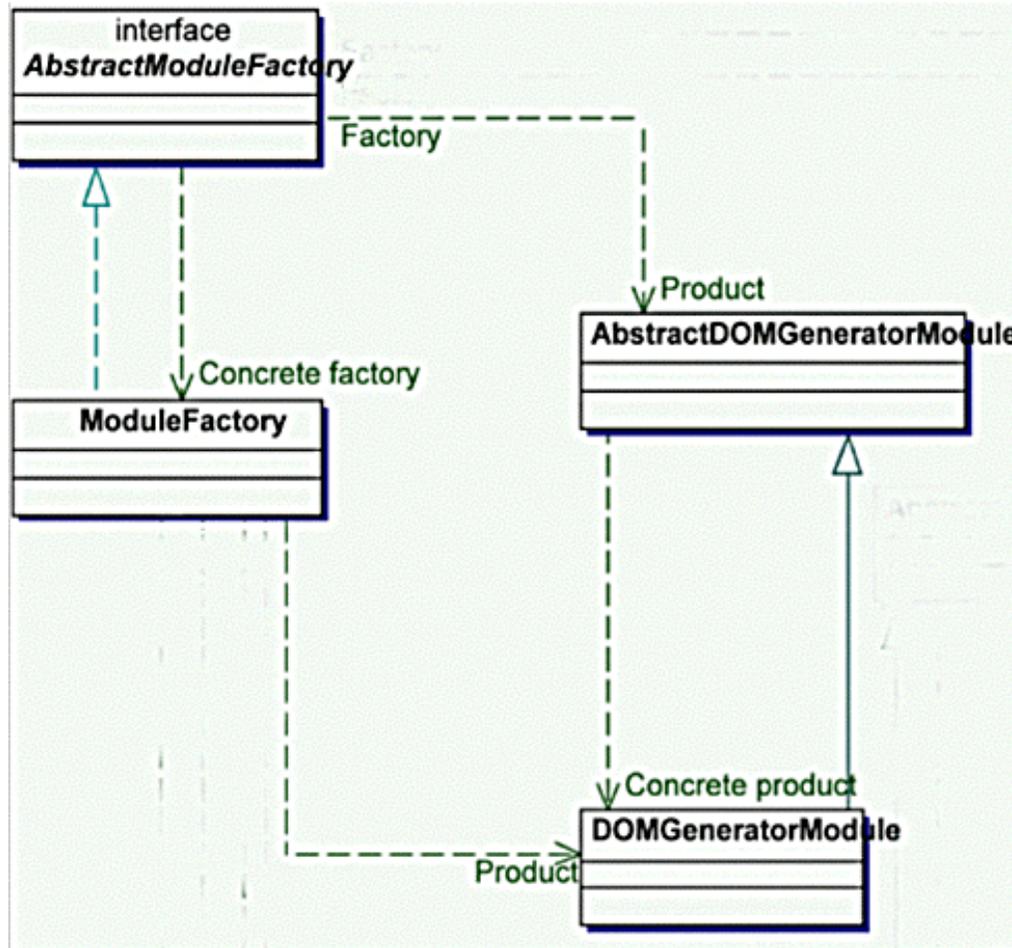
Gliederung

- Präsentation der Aufgabenstellung
- Wiederholung des Zwischenberichtergebnisses
- Architektur der Zwischenschicht
- Architektur des Generators
- Nachweis der Funktionstüchtigkeit (Zwischenschicht/Generator)
- Zusammenfassung/Ausblick auf EJB 2.0

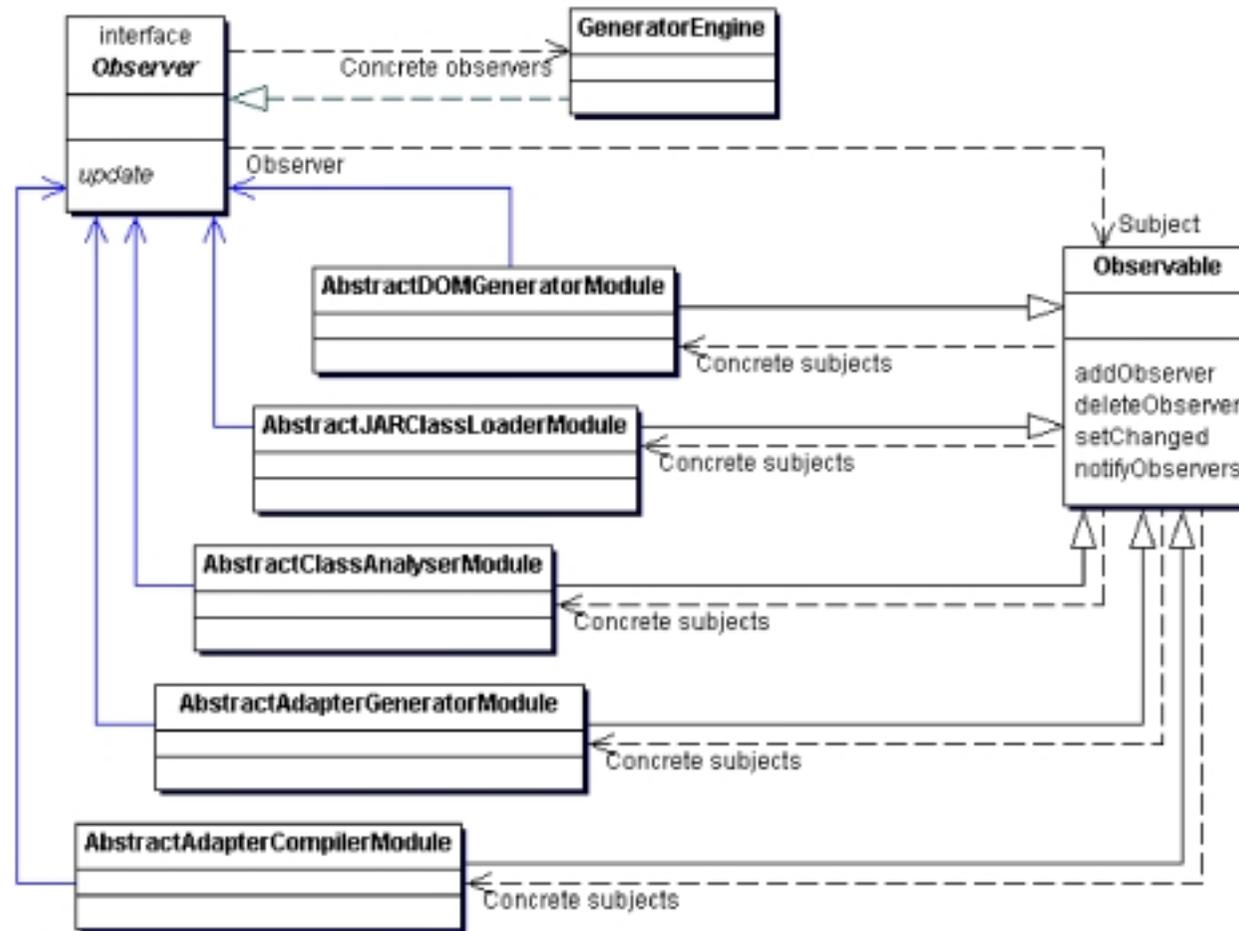
Generatorprozess



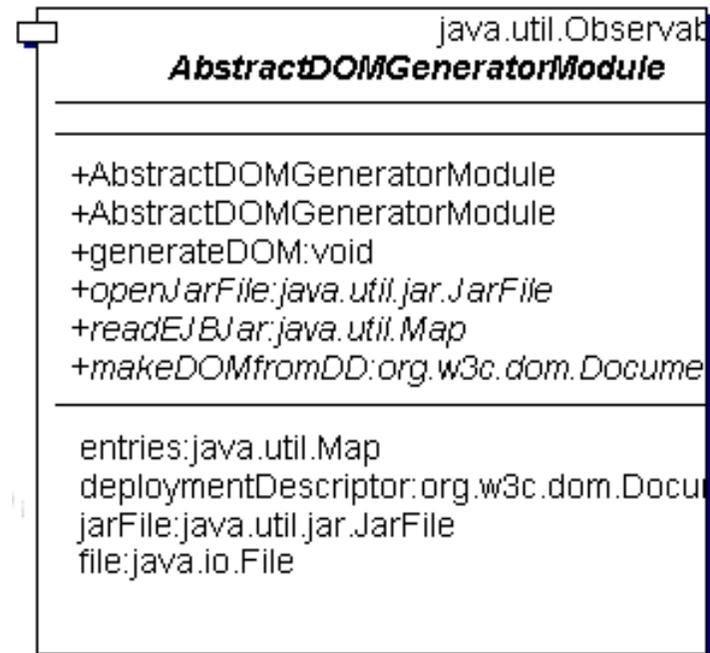
AbstractFactory-Muster



Observer-Muster



Modul-Klassen (Ausschnitt)



Gliederung

- Präsentation der Aufgabenstellung
- Wiederholung des Zwischenberichtergebnisses
- Architektur der Zwischenschicht
- ~~Architektur des Generators~~
- Nachweis der Funktionstüchtigkeit (Zwischenschicht/Generator)
- Zusammenfassung/Ausblick auf EJB 2.0

Nachweis der Funktionstüchtigkeit

- Nachweis erfolgt(e) in vier Schritten:
 - Übereinstimmung manuell/automatisch generierter EJB-Adapter.
 - Überprüfung der EJB-Adapter auf JavaBeans-Konformität (mittels BDK 1.1).
 - Visuelle Anwendungserstellung mittels „VisualAge for Java“ und EJB-Adapter.
 - Prüfung einzelner Eigenschaften mit Hilfe einer EJB-Adapter-basierten Beispielanwendung.

Gliederung

- Präsentation der Aufgabenstellung
- Wiederholung des Zwischenberichtergebnisses
- Architektur der Zwischenschicht
- Architektur des Generators
- Nachweis der Funktionstüchtigkeit (Zwischenschicht/Generator)
- Zusammenfassung/Ausblick auf EJB 2.0

Zusammenfassung

- Aufgabenstellung grundlegend erfüllt!
- „Offene Punkte“:
 - Postulierte Modellintegrität bei „Zugriffsschnittstellen als Rückgabetyt“ durchbrochen!
 - Transaktionsattribut „Mandatory“ ignoriert!
 - Kein automatisches Verpacken in JAR-Archive!
 - BeanInfo-Klasse wird ignoriert!

Ausblick auf EJB 2.0

- Einführung von EJB-QL.
- Einführung von „Local Interfaces“.
- Einführung von globalen „Home-Interface“-Methoden.
- Einführung von „Entity-Relationships“.
- Modifizierung der „Container-Managed Persistence“.
- Fazit: Anpassung nur in geringem Umfang notwendig!

Diskussion

- Fragen?
- Anmerkungen?
- Verbesserungsvorschläge?
- sean.eikenberg@inf.tu-dresden.de