

Vorlesung Internetprogrammierung

Enterprise JavaBeans—Patterns

Peter Thiemann

Universität Freiburg

Vorlesung Internetprogrammierung, SS2006

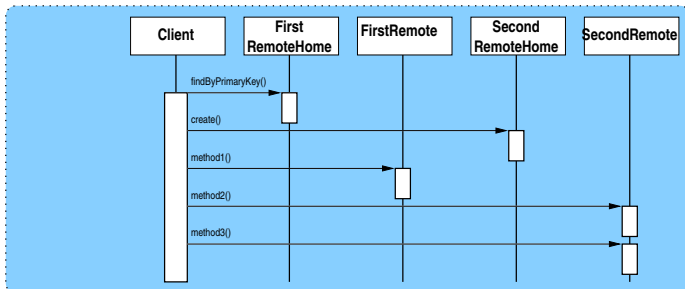
- 1 Designmuster
 - Sitzungsfassade
 - Wertobjekte

- 2 Implementierungsmuster
 - Geschäftsinterfaces
 - Exceptions
 - Primärschlüssel
 - Große Anfragen

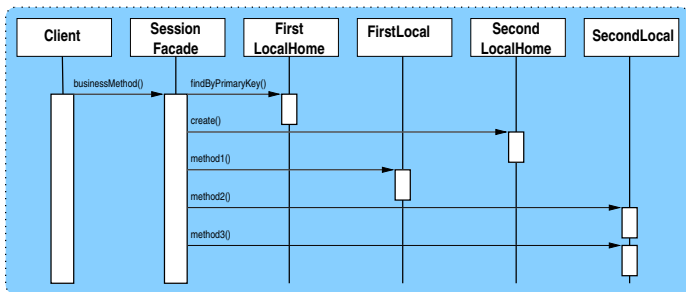
Sitzungsfassade

Problem

- Client hängt direkt von (Entity) Bean Methoden ab
- Viele externe Zugriffe über Remote Interfaces

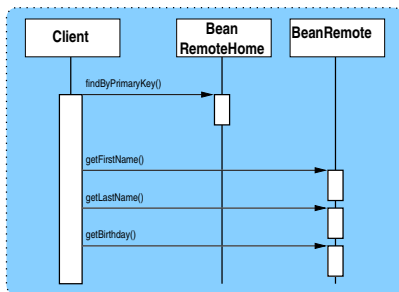


- Einführen einer Sitzungsfassade (session facade)
- Implementiert durch stateless Session-Bean
- Externe Zugriffe \Rightarrow interne Zugriffe über Local Interfaces

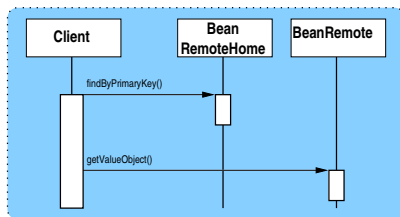


- Vereinfachung des Client-Code durch Kapselung eines Geschäftsablaufs
- Methode im Session-Bean läuft innerhalb einer Transaktion
- Vermeidung von Remote-Zugriffen (teuer)
- Vgl. "Fassade" Designmuster

- Jeder Zugriff auf ein Remote-Interface entspricht einer Remote-Method-Invocation
- Jede Verwendung von RMI hat Overhead (Netzwerk, Serialisierung)
- Zugriff auf Komponenten eines Entity-Bean zu teuer



- Definition von Wertobjekten: (*dependent*) *value objects*
- Ein Wertobjekt fasst mehrere Komponenten (eines Entity-Bean) zusammen
- Wertobjekte müssen serialisierbar sein, da sie im Remote-Interface verwendet werden
- Wertobjekte sind schreibgeschützt, da Änderungen an ihnen nicht persistent sind



Wertobjekte

Beispiel

```
public class PersonDO implements java.io.Serializable {
    private String firstName;
    private String lastName;
    private java.util.Date birthday;

    public PersonDO (String firstName, String lastName,
                    java.util.Date birthday) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.birthday = birthday;
    }

    public String getFirstName () {
        return this.firstName;
    }
    public String getLastName () {
        return this.lastName;
    }
    public java.util.Date getBirthday () {
        return this.birthday;
    }
}
```


- Beschleunigung des Remote-Zugriffs auf Entity-Beans
- (Remote) Datenübermittlung an Session-Beans
- (Allgemein sollten Remote-Aufrufe möglichst viele Daten auf einmal transportieren)
- Anstelle eines Entity-Beans, z.B. als Information, die extern veröffentlicht werden soll
Beispiel: Buchungsbestätigung
- Dürfen als Typ von CMP-Feldern verwendet werden
Nicht empfehlenswert, da Speicherung in der Datenbank in serialisierter Form
- Entkopplung von internem Persistenzmodell und Client-Code
- Unterschiedliche Wertobjekte für verschiedene Sichten auf gleiches Entity-Beans

- Problem
 - Die Bean-Klasse implementiert nicht das Interface des Beans.
 - Inkonsistenzen werden erst beim Deployment bemerkt.
- Lösungsansatz: Definition eines Geschäftsinterface
 - Definiert die Geschäftsmethoden des Beans
 - Die Bean-Interfaces und die Bean-Klasse erben vom Geschäftsinterface
 - Inkonsistenzen werden beim Compilieren bemerkt.
 - **Nur zu diesem Zweck verwenden!**

Geschäftsinterfaces

Beispiel

```
// business interface  
import java.rmi.RemoteException;  
  
public interface OrderBI {  
    public int getQuantity () throws RemoteException;  
    public void setQuantity (int q) throws RemoteException;  
    public double getPricePerItem () throws RemoteException;  
    public void setPricePerItem (double p) throws RemoteException;  
  
    public double getTotalPrice () throws RemoteException;  
}
```

```
// remote interface  
import javax.ejb.EJBObject;  
public interface OrderRemote extends Order, EJBObject {  
}
```

Geschäftsinterfaces

Beispiel/2

```
// bean class
public class OrderBean implements Order, EntityBean {
    private int quantity;
    private double pricePerItem;

    public int getQuantity () {
        return quantity;
    }
    public void setQuantity (int q) {
        quantity = q;
    }
    public double getPricePerItem () {
        return pricePerItem;
    }
    public void setPricePerItem (int p) {
        pricePerItem = p;
    }
    public double getTotalPrice () {
        return quantity * pricePerItem;
    }
    // further entity bean methods
}
```

Exceptions

RemoteException

- Methoden von Remote-Interfaces
- Netzwerkprobleme
- Abgefangen durch Clienten-Code

EJBException

- Ausgelöst in der EJB-Implementierung (Bean-Klasse) bei schwerwiegenden Problemen
- Abgefangen durch EJB-Container, **nicht** durch Entwickler
- Rollback der aktuellen Transaktion

Exceptions der Anwendung

- Spezielle Ergebnisse von Methodenaufrufen
- Werden an die Anwendung weitergereicht

Primärschlüssel

- Primärschlüssel dürfen aus mehreren CMP-Feldern zusammengesetzt werden
- Separate Schlüsselklasse muss definiert werden
- Kann Performanceprobleme ergeben:
 - aus Schlüssel muss `hashCode()` berechnet werden
 - Verwendung als Fremdschlüssel
- Allgemeine Empfehlungen
 - ein CMP-Feld als Primärschlüssel
 - möglichst mit einfachem Typ
 - darf sich nie ändern (dient als Identifikation eines Tupels)
 - oft systemgenerierter Surrogatschlüssel

- Aufgabe: Naives Anzeigen einer Bücherliste
 - Finder-Methode liefert eine Collection von Entity-Beans
 - Hole Daten durch Abfragen von Wertobjekten
 - Problem: Sehr ineffizient
- Lösungsvorschlag
 - Umgehen von Entity-Beans
 - Direkte Anfrage an die Datenbank (nur lesend!)
 - Ergebnis: Liste von Wertobjekten
 - Gekapselt in Session-Bean