

Kod szkolenia: **EJB**

Tytuł szkolenia: **Tworzenie komponentów logiki biznesowej i warstwy dostępu do danych w oparciu o EJB3.0/JPA lub EJB 3.1/JPA2**

Dni: 3

## Opis:

### Adresaci szkolenia:

Szkolenie adresowane jest do programistów Java pragnących zapoznać się z zasadami tworzenia rozproszonych komponentów logiki biznesowej w oparciu o EJB i warstwy trwałości opartej na JPA.

### Cel szkolenia:

Celem szkolenia jest poznanie zasad tworzenia rozproszonych komponentów logiki biznesowej w oparciu o EJB 3.0/3.1 oraz warstwy trwałości w oparciu o JPA 1.0/2.0 wraz z zagadnieniami integracji, testowania oraz skalowania.

W szczególności uczestnicy:

- zapoznają się z problemami architektury rozproszonej oraz sposobem jej realizacji przy pomocy EJB.
- poznają komponenty sesyjne stanowe i bezstanowe oraz komponenty sterowane wiadomością wraz z ich zastosowaniem.
- dowiedzą się jak kontener zarządza transakcjami oraz bezpieczeństwem.
- poznają zasady uruchamiania i integracji aplikacji opartych na EJB.
- poznają zasadę mapowania relacyjno-objektowego, i nauczą się projektować warstwę trwałości w oparciu o JPA począwszy od prostych encji po złożone zagadnienia takie jak dziedziczenie.
- dowiedzą się jak dbać o spójność danych i wydajność aplikacji.
- dowiedzą się jak testować ww. komponenty.

### Wymagania:

Od uczestników szkolenia wymagana jest umiejętność programowania w języku Java (do poznania na kursie J/JP), podstawy relacyjnych baz danych i SQL.

### Parametry szkolenia:

3\*8 godzin (3\*7 godzin netto)(w wariancie EJB3.0/JPA1) lub 4\*8 godzin (4\*7 godzin netto) (w

wariacie EJB3.1/JPA2), wykładów i warsztatów (z wyraźną przewagą warsztatów). W trakcie warsztatów, oprócz prostych ćwiczeń, projektowana i implementowany jest backend prostej aplikacji.

Wielkość grupy: maks. 8-10 osób.

## Program szkolenia:

\* pozycje oznaczone gwiazdką dotyczą jedynie wariantu EJB3.1/JPA2

### 1. Wstęp

- I. Wprowadzenie do platformy Java Enterprise Edition
- II. Omówienie komponentów i specyfikacji JEE w tym JNDI, JMS, RMI, Servlet, JSP, EJB, webserwisy,
- III. Zagadnienia projektowania aplikacji na platformie JEE, doboru technologii, bezpieczeństwa etc.

### 2. EJB

- I. Typy komponentów EJB 3.x
- II. Zasada działania
- III. Komponenty sesyjne
  - i. Stanowe
  - ii. Bezstanowe
  - iii. Komponenty typu Singleton\*
  - iv. cykl życia komponentów, nasłuchiwanie i sterowanie cyklem życia
  - v. Rodzaje dostępu
    - A. Lokalny
    - B. Lokalny bez-interfejsowy\*
    - C. Zdalny
    - D. Zdalny przez usługę web service
- IV. Wywołania asynchroniczne metod komponentów sesyjnych\*
- V. Zagadnienia współbieżności w odwoływaniu się do komponentów
  - i. Zachowanie kontenera
  - ii. Współbieżność zarządzana przez kontener i współbieżność zarządzana przez komponent\*
  - iii. Komponent stanowy w kontekście współbieżności
- VI. Korzystanie z zasobów przez komponenty EJB
  - i. Słowo o JCA
  - ii. Przestrzeń nazw JNDI i wyszukiwanie zasobów
  - iii. Inversion of Control w odwoływaniu do zasobów
- VII. Interceptory
- VIII. Model wyjątków
- IX. Komponenty Sterowane Wiadomością
  - i. Zasada działania
  - ii. Rodzaje komunikacji
- X. Bezpieczeństwo komponentów
- XI. Transakcje zarządzane przez kontener i komponent

- i. Transakcje zarządzane przez kontener (CMT)
        - A. Atrybuty transakcji w kontekście scenariuszy użycia
        - B. Transakcje a wyjątki
        - C. Synchronizacja stanu w komponencie stanowym
      - ii. Transakcje zarządzane przez komponent (BMT) – podstawy JTA
      - iii. Transakcje zarządzane przez klienta
    - XII. Usługi czasowe (TimerService)
      - i. Interfejsy Timer i TimerService
      - ii. Automatyczne uruchamianie usług w oparciu o kalendarz\*
    - XIII. Usługi webservice w kontekście EJB
      - i. Podstawy webservice
      - ii. JAX-WS a EJB
      - iii. Przygotowanie, konfigurowanie i wdrażanie aplikacji
        - A. Jako samodzielny moduł EJB (jar)
        - B. Jako składowa aplikacji Enterprise (ear)
        - C. W obrębie aplikacji Web (war) \*
        - D. Kontener pełny (full) i lekki (light)\*
        - E. EJB w kontenerze zaembedowanym\*
      - iv. Testowanie komponentów EJB
      - v. Wzorce Projektowe warstwy logiki i dostępu do logiki
- 3. Java Persistence API
  - I. Koncepcja Mapowania relacyjno-objektowego
  - II. Klasy Encyjne
    - i. klasy proste
    - ii. klasy zagnieżdżone
    - iii. Pola i dostęp do pól
    - iv. Pola będące kolekcjami \*
  - III. Asocjacje – z uwzględnieniem krotności i kierunkowości
  - IV. Klucze
    - i. Proste
    - ii. Złożone
    - iii. Automatyczne generowanie kluczy
  - V. Nazewnictwo obiektów w bazie danych
    - i. Domyślne mapowanie
    - ii. Zmiana domyślnych mapowań
  - VI. Kontekst trwałości (PersistentContext)
    - i. Zarządzany przez kontener i zarządzany przez aplikację
    - ii. Transakcyjny(Transaction) i rozszerzony(Extended)
  - VII. Cykl życia encji
  - VIII. Operacje na danych
  - IX. Transakcje
  - X. Zapytania
    - i. Tworzenie zapytań w JPQL
      - A. Składnia
      - B. Query i TypedQuery\*
      - C. Zapytania nazwane



- ii. JPA2 Criteria API\*
    - A. Idea Criteria API, podstawowe składowe
    - B. MetaModel
      - a. Koncepcja
      - b. Klasy MetaModel w postaci kanonicznej i niekanonicznej
      - c. MetaModel API
    - C. Tworzenie silnie typowanych zapytań
    - D. Tworzenie słabo typowanych, dynamicznych zapytań
    - E. Edycja zapytań
    - F. Podejścia alternatywne: LIQUidFORM
  - iii. Typowe scenariusze, w tym serach-within i serach-by-example
  - iv. Zapytania Natywne, mapowanie wyników
- XI. Metody nasłuchiwania cyklu życia
- XII. Walidacja
- i. Podstawy Bean Validation 1.0 (JSR 303)
    - A. Koncepcja Bean Validation
    - B. Definiowanie i nakładanie ograniczeń na typy, pola, metody
    - C. Parametryzowanie ograniczeń
    - D. Wbudowane ograniczenia
    - E. Komponowanie ograniczeń złożonych
    - F. Tworzenie Walidatorów
    - G. Proces walidacji
      - a. Pola, obiekty i grafy obiektów
      - b. Grupy i sekwencje
      - c. Blokowanie dostępu do pola oraz kaskadowego dostępu
    - H. API i wyjątki
  - ii. JPA2 a Bean Validation
    - A. Konfiguracja walidacji
    - B. Zasada działania Bean Validation w kontekście JPA
    - C. Ograniczenia walidujące a generacja obiektów bazy danych
- XIII. Modele i implementacja dziedziczenia
- XIV. Zagadnienia spójności danych
- i. Optimistic locking
  - ii. Pessimistic locking
  - iii. Rozszerzenia w JPA2 \*
- XV. Cache drugiego poziomu
- i. Idea cache 2 poziomu
  - ii. Konfiguracja i tryby pracy cache
  - iii. Konfiguracja encji
  - iv. Interakcja cache <-> baza
  - v. Cache w oparciu o Hibernate (dla szkolenia bazującego na hibernate)
  - vi. Cache w oparciu o mechanizmy JPA2\*
  - vii. Zagrożenia
- XVI. Testowanie komponentów wykorzystujących JPA
- XVII. Konfiguracja i deployment

- i. Jako aplikacja Java SE / dektop
  - ii. Jako moduł web
  - iii. W obrębie modułu ejb
- XVIII. JPA a biblioteki dostawców(Hibernate,Toplink,EclipseLink,OpenJPA)
- XIX. Zagadnienia specyficzne dla Hibernate (dla szkolenia bazującego na hibernate)
  - i. Hibernate API
  - ii. Wybrane dodatkowe możliwości
  - iii. Podprojekty Hibernate: Search, Shards, Validator, Envers
- XX. Wzorce Projektowe warstwy dostępu do danych

