

## Kod szkolenia: J/JPA-HIB

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

### Adresaci szkolenia:

Szkolenie adresowane jest do programistów Java pragnących dowiedzieć się jak tworzyć warstwę dostępu do danych w oparciu o koncepcje object relational mapping z użyciem JPA2/Hibernate.

### Cel szkolenia:

Celem szkolenia jest poznanie zasad tworzenia warstwy dostępu do danych w oparciu o koncepcje object relational mapping z użyciem JPA2/Hibernate. wraz z zagadnieniami integracji, testowania oraz skalowania.

Szkolenie może być realizowane w oparciu o API JPA2 z uwzględnieniem specyfików dostawcy lub w oparciu o API Hibernate.

W szczególności uczestnicy:

- 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 odczytywać i utrzymywać encje, zapewniać transakcyjność operacji
- dowiedzą się jak dbać o spójność danych i wydajność aplikacji (w tym cache 2 poziomu)
- dowiedzą się jak integrować opracowane rozwiązania z aplikacją w oparciu o rozwiązania EJB3/Seam/Spring
- 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:

2\*7 godzin wykładów i warsztatów w proporcji 1/3.

## Polecane szkolenia poprzedzające:

J/JP

## Program szkolenia:

- Koncepcja Mapowania relacyjno- obiektowego
- Klasy Encyjne
  - klasy proste
  - klasy zagnieżdżone
  - Pola i dostęp do pól
  - Pola będące kolekcjami \*
- Asocjacje – z uwzględnieniem krotności i kierunkowości
- Klucze
  - Proste
  - Złożone
  - Automatyczne generowanie kluczy
- Nazewnictwo obiektów w bazie danych
  - Domyślne mapowanie
  - Zmiana domyślnych mapowań
- Kontekst trwałości (PersistentContext):
  - Zarządzany przez kontener i zarządzany przez aplikację
  - Transakcyjny(Transaction) i rozszerzony(Extended)
- Cykl życia encji
- Operacje na danych
- Transakcje
- Zapytania
  - Tworzenie zapytań w JPQL
    - Składnia
    - Query i TypedQuery\*
    - Zapytania nazwane
  - JPA2 Criteria API\*
    - Idea Criteria API, podstawowe składowe
    - MetaModel
      - Koncepcja
      - Klasy MetaModel w postaci kanonicznej i niekanonicznej
      - MetaModel API
  - Tworzenie silnie typowanych zapytań

- Tworzenie słabo typowanych, dynamicznych zapytań.
- Edycja zapytań
- Podejścia alternatywne: LIQUidFORM
- Typowe scenariusze, w tym serach-within i serach-by-example
- Zapytania Natywne, mapowanie wyników
- Metody nasłuchiwanie cyklu życia
- Walidacja
  - Podstawy Bean Validation 1.0 (JSR 303)
    - Koncepcja Bean Validation
    - Definiowanie i nakładanie ograniczeń na typy, pola, metody
    - Parametryzowanie ograniczeń
    - Wbudowane ograniczenia
    - Komponowanie ograniczeń złożonych
    - Tworzenie Walidatorów
    - Proces walidacji
      - Pola, obiekty i grafy obiektów
      - Grupy i sekwencje
      - Blokowanie dostępu do pola oraz kaskadowego dostępu
    - API i wyjątki
  - JPA2 a Bean Validation
    - Konfiguracja walidacji
    - Zasada działania Bean Validation w kontekście JPA
    - Ograniczenia walidujące a generacja obiektów bazy danych
- Modele i implementacja dziedziczenia,
- Zagadnienia spójności danych,
  - optimistic locking
  - pesimistic locking
  - Rozszerzenia w JPA2 \*
- Cache drugiego poziomu
  - Idea cache 2 poziomu,
  - Konfiguracja i tryby pracy cache
  - Konfiguracja encji
  - Interakcja cache <-> baza
  - Cache w oparciu o Hibernate (dla szkolenia bazującego na hibernate)
  - Cache w oparciu o mechanizmy JPA2\*
  - Zagrożenia
- Testowanie komponentów wykorzystujących JPA
- Konfiguracja i deployment
  - Jako aplikacja Java SE / dektop

- Jako moduł web
- W obrębie modułu ejb
- JPA a biblioteki dostawców(Hibernate,Toplink,EclipseLink,OpenJPA)
- Zagadnienia specyficzne dla Hibernate (dla szkolenia bazującego na hibernate)
  - Hibernate API
  - Wybrane dodatkowe możliwości
  - Podprojekty Hibernate: Search, Shards, Validator, Envers
- Wzorce Projektowe warstwy dostępu do danych