

TWORZENIE APLIKACJI Z UŻYCIEM EJB3

KOD: JEJB

Profil uczestnika

Uczestnik:

- posiada minimum roczne doświadczenie w programowaniu w języku Java;
- zna podstawy tworzenia aplikacji w Java SE;
- zna założenia technologii Java EE;
- chce poznać efektywne sposoby używania EJB w tworzeniu aplikacji internetowych.

Korzyści ze szkolenia

1. **Duża intensywność szkolenia** – w niedługim czasie uczestnik poznaje najbardziej użyteczne aspekty EJB, umożliwiające efektywne tworzenie aplikacji rozproszonych w technologii Java EE.
2. **Aplikacje korporacyjne tworzone są w uznanym i sprawdzonym standardzie** – EJB to wiodący standard w technologii Java.
3. **Programista potrafi rozwiązywać problemy podczas tworzenia aplikacji** – w trakcie szkolenia uczymy, w jaki sposób radzić sobie z potencjalnymi problemami, które mogą pojawić się w trakcie tworzenia projektu. Pokazujemy sprawdzone techniki i triki programistyczne.

Parametry szkolenia

CZAS TRWANIA: 3 dni – 24 godziny

FORMA ZAJĘĆ: Ćwiczenia - 60%, wykład – 40%.

WIELKOŚĆ GRUPY: ok. 10 osób.

Szczegółowy program

Moduły szkoleniowe	Nabyte wiedza i umiejętności, poruszane zagadnienia
Wprowadzenie	<ul style="list-style-type: none">• EJB w architekturze Java EE• Przegląd EJB<ul style="list-style-type: none">○ Komponenty sesyjne○ Obiekty encyjne○ Komponenty sterowane zdarzeniami
Komponenty sesyjne	<ul style="list-style-type: none">• Przegląd<ul style="list-style-type: none">○ Usługi kontenera○ Komponenty stanowe i bezstanowe○ Definiowanie komponentu sesyjnego – adnotacje EJB○ Implementacja, interfejsy zdalny i lokalny• Model wdrożeniowy: ejb-jar, EAR, kontener• JNDI<ul style="list-style-type: none">○ Koncept JNDI○ InitialContext○ JNDI w praktyce• Tworzenie klienta EJB3• Wstrzykiwanie zależności w EJB3• Korzystanie z zasobów poprzez kontener• Cykl życia komponentu sesyjnego oraz interceptory• Stanowe komponenty sesyjne – tworzenie, cykl życia, aktywa-

	<p>cja i pasywacja</p> <ul style="list-style-type: none">• Usługa mechanizmu Timer
Komponenty sterowane komunikatami	<ul style="list-style-type: none">• Przegląd systemów opartych o komunikaty<ul style="list-style-type: none">○ Komunikacja poprzez kolejki i kanały tematyczne• Przegląd JMS API<ul style="list-style-type: none">○ Connection Factory○ Przesyłanie komunikatów JMS○ Klient JMS• Komponenty MDB<ul style="list-style-type: none">○ Założenia○ Adnotacja @MessageDriven○ Przykład komponentu MDB○ Konfiguracja activeConfig
Transakcje i bezpieczeństwo	<ul style="list-style-type: none">• Transakcje<ul style="list-style-type: none">○ Przegląd koncepcji związanych z systemami transakcyjnymi○ Deklaratywne zarządzanie transakcjami○ Atrybuty transakcji○ Zasięg transakcji• Przegląd zagadnień bezpieczeństwa w systemach rozproszonych<ul style="list-style-type: none">○ Adnotacje @RolesAllowed, @PermitAll○ Bezpieczeństwo obsługiwane w sposób programistyczny
Wprowadzenie do JPA	<ul style="list-style-type: none">• Idea mapowania relacyjno-objektowego• Podstawy mapowania<ul style="list-style-type: none">○ Adnotacje @Entity i @Id○ Generatory identyfikatorów○ Podstawowe typy użyteczne podczas mapowania• EntityManager<ul style="list-style-type: none">○ Rola EntityManager w JPA○ Jednostki trwałości (Persistence Unit) i ich konfigu-

racja

- Wstrzykiwanie obiektu typu EntityManager
- Cykl życia komponentów trwałych
- Podstawowe operacje typu CRUD
- Zapytania w JPA
 - Zapytania obiektowe
 - Instrukcja select
 - Instrukcja where
 - Nazwane zapytania
- Wersjonowanie i blokowanie optymistyczne

Mapowanie relacji

- Przegląd relacji
- Mapowanie 1:1, 1:N, N:1, N-N
- Relacje jedno- i dwukierunkowe
- Adnotacje @OneToOne, @OneToMany, @ManyToMany
- Kolumny złączeń
- Relacje odwrotne
- Opóźnione i wczesne ładowanie (Lazy/Eager Loading)
- Kaskadowość operacji
- Typy złączeń: Inner join, Outer join, Fetch join
- Dziedziczenie
- Najlepsze praktyki mapowania