

Kod szkolenia: **JEE7/JSF**

Tytuł szkolenia: **Nowoczesne aplikacje internetowe oparte o JEE7 (JSF2.2, EJB3.2, JPA2.1, CDI1.1, AJAX)**

Dni: 5

Opis:

Adresaci szkolenia:

Szkolenie adresowane jest do programistów aplikacji internetowych na platformie JEE, pragnących uzupełnić swoje kwalifikacje o najnowsze technologie, a także dla osób znających język Java i pragnących rozpocząć swoją przygodę z aplikacjami internetowymi od najnowszych technologii. Szkolenie realizowane jest z użyciem najnowszej odsłony Java Enterprise Edition – wersji 7, która kusi możliwościami lecz nie ma jeszcze pełnego wsparcia u dostawców serwerów aplikacji. Dla osób pragnących pracować z ugruntowanymi technologiami polecamy szkolenie WEB2.

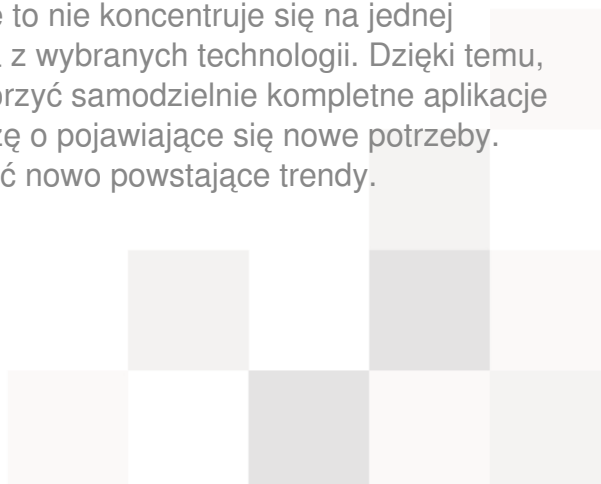
Cel szkolenia:

Uczestnicy dowiedzą się jak projektować i implementować aplikacje na platformie JEE z wykorzystaniem technologii będących treścią szkolenia. Celem szkolenia jest zdobycie umiejętności tworzenia skalowanych aplikacji, zgodnie z arkanami sztuki: od komponentów biznesowych po wysoce ergonomiczne i funkcjonalne interfejsy wpisujące się w nurt WEB 2.0. W szczególności: Uczestnicy szczegółowo poznają technologię prezentacji JSF2.2 włącznie z tworzeniem własnych walidatorów, konwerterów czy komponentów. Zaznajomią się z popularnymi, bogatymi bibliotekami komponentów, takimi jak RichFaces, IceFaces. Nauczą się projektować i implementować warstwę logiki biznesowej w oparciu EJB 3.2, w połączeniu z mechanizmem mapowania relacyjno-obiektowego JPA2.1. Poznają specyfikację CDI.

Mocne strony szkolenia:

Program obejmuje całościowo i wyczerpująco zagadnienia tworzenia aplikacji internetowych. W przeciwieństwie do większości szkoleń, szkolenie to nie koncentruje się na jednej technologii, lecz zawiera najważniejsze zagadnienia z wybranych technologii. Dzięki temu, uczestnicy po skończeniu szkolenia, będą mogli tworzyć samodzielnie kompletne aplikacje WWW, uzupełniając już we własnym zakresie wiedzę o pojawiające się nowe potrzeby. Program jest ciągle uaktualniany tak, by uwzględniać nowo powstające trendy.

Wymagania:



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

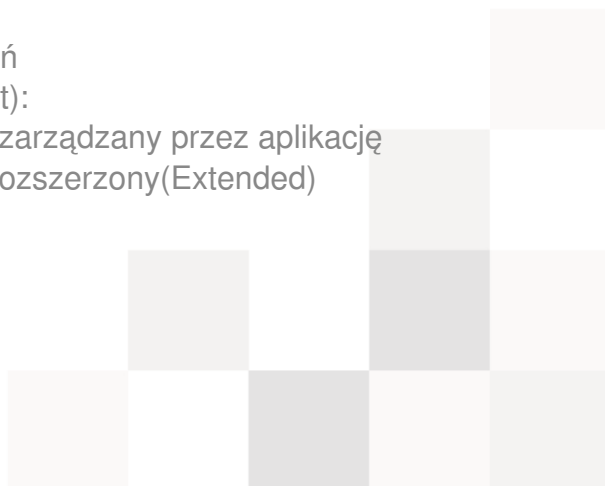
Parametry szkolenia:

5*8 godzin (5*7 netto) wykładów i warsztatów (z wyraźną przewagą warsztatów). W trakcie warsztatów, oprócz prostych ćwiczeń, projektowana i implementowana jest prosta kompleksowa aplikacja. Wielkość grupy: maks. 8-10 osób

Program szkolenia:

1. Wstęp
 - I. Wprowadzenie do platformy Java Enterprise Edition
 - II. Omówienie komponentów i specyfikacji JEE w tym JNDI, JMS, Servlet, JSP, EJB, CDI, webserwis
 - III. Zagadnienia projektowania aplikacji na platformie JEE, doboru technologii, bezpieczeństwa etc.
2. Specyfikacja Managed Beans
3. Warstwa logiki biznesowej – komponenty 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 dostępów
 - A. Lokalny
 - B. Lokalny bez-interfejsowy
 - C. Zdalny
 - D. Zdalny przez usługę web service
 - vi. Wywołania asynchroniczne metod komponentów sesyjnych
 - vii. Zagadnienia współbieżności w odwoływaniu się do komponentów:
 - A. Zachowanie kontenera
 - B. Współbieżność zarządzana przez kontener i współbieżność zarządzana przez komponent
 - C. Komponent stanowy w kontekście współbieżności
 - IV. 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
 - V. Interceptory
 - VI. Model wyjątków
 - VII. Komponenty Sterowane Wiadomością:
 - i. JMS2.0
 - ii. Zasada działania

- iii. Rodzaje komunikacji
 - VIII. Bezpieczeństwo komponentów
 - IX. 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. JTA a JTS, CORBA
 - iv. Transakcje zarządzane przez klienta
 - X. Usługi czasowe (TimerService)
 - i. Interfejsy Timer i TimerService
 - ii. Automatyczne uruchamianie usług w oparciu o kalendarz
 - iii. @Schedule, @Schedules, @Timeout
 - XI. Usługi webservice w kontekście EJB
 - i. Podstawy webservice
 - ii. JAX-WS/JAX-RS a EJB/CDI
 - XII. Przygotowanie, konfigurowanie i wdrażanie aplikacji
 - i. Jako samodzielny moduł EJB (jar)
 - ii. Jako składowa aplikacji Enterprise (ear)
 - iii. W obrębie aplikacji Web (war)
 - iv. Kontener pełny (full) i lekki (light), profile serwera JEE
 - v. EJB w kontenerze zaembedowanym
 - XIII. Testowanie komponentów EJB
 - XIV. Wzorce Projektowe warstwy logiki i dostępu do logiki
4. Java Persistence API 2.1
- 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
 - D. Zapytania natywne
 - ii. JPA2.1 Criteria API
 - A. Idea Criteria API, podstawowe składowe
 - B. MetaModel
 - @. Koncepcja
 - @. Klasy MetaModel w postaci kanonicznej i niekanonicznej
 - @. 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. Bean Validation 1.1
 - 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
 - @. Pola, obiekty i grafy obiektów
 - @. Grupy i sekwencje
 - @. Blokowanie dostępu do pola oraz kaskadowego dostępu
 - H. API i wyjątki
 - I. Dostosowywanie treści błędów walidacyjnych
- XIII. JPA2.1 a Bean Validation
 - i. Konfiguracja walidacji
 - ii. Zasada działania Bean Validation w kontekście JPA
 - iii. Ograniczenia walidujące a generacja obiektów bazy danych
- XIV. Modele i implementacja dziedziczenia
- XV. Zagadnienia spójności danych
 - i. Optimistic locking
 - ii. Pessimistic locking
 - iii. Rozszerzenia w JPA2.1
- XVI. Cache drugiego poziomu
 - i. Idea cache 2 poziomu
 - ii. Konfiguracja i tryby pracy cache

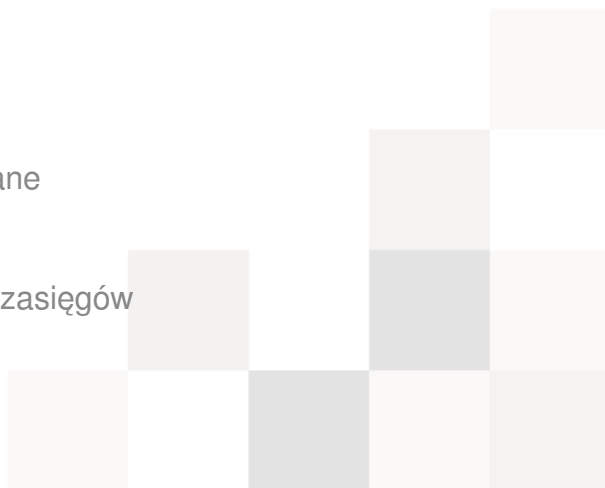


- iii. Konfiguracja encji
 - iv. Interakcja cache <-> baza
 - v. Cache w oparciu o Hibernate (dla szkolenia bazującego na hibernate)
 - vi. Cache w oparciu o mechanizmy JPA2.1
 - vii. Zagrożenia
 - XVII. Testowanie komponentów wykorzystujących JPA
 - XVIII. Konfiguracja i deployment
 - i. Jako aplikacja Java SE/dekstop
 - ii. Jako moduł web
 - iii. W obrębie modułu ejb
 - XIX. JPA a biblioteki dostawców (Hibernate, Toplink, EclipseLink, OpenJPA)
 - XX. Zagadnienia specyficzne dla Hibernate (dla szkolenia bazującego na hibernate)
 - i. Hibernate API
 - ii. Wybrane dodatkowe możliwości
 - iii. Podprojekty Hibernate: Search, Shards, Validator, Envers
 - XXI. Wzorce Projektowe warstwy dostępu do danych
5. Technologia Java Servlet 3.1
- I. Wprowadzenie
 - II. Cykl życia servletu
 - III. Współdzielenie informacji
 - IV. Tworzenie i inicjalizacja servletu
 - V. Obsługa żądań i odpowiedzi
 - VI. Filtrowanie żądań
 - VII. Uploading plików, Media-typy
 - VIII. Dostęp do WebContext
 - IX. Asynchroniczność
 - X. Nieblokujące operacje I/O
 - XI. Wsparcie dla HTTP 1.1
 - XII. Bezpieczeństwo z poziomu servletu
6. Interfejs Webowy
- I. Wprowadzenie do JSF
 - i. Typowe problemy aplikacji Webowej
 - ii. Wprowadzenie do modelu komponentowo-zdarzeniowego
 - II. Cykl przetwarzania żądania
 - i. Rodzaje żądań
 - ii. Fazy cyklu
 - iii. Przetwarzanie pełne i częściowe
 - III. Komponentowy model interfejsu
 - IV. Główne klasy i interfejsy
 - V. Model zarządzania beanami logiki
 - i. Idea i rodzaje zasięgów
 - ii. Konfiguracja
 - iii. Zasięgi „Custom”
 - VI. Expression Language
 - VII. Definiowanie widoku



- i. Przegląd dostępnych komponentów standardowych
 - A. Biblioteka HTML
 - B. Biblioteka Core
- ii. View Declaration Language
- iii. JSF w oparciu o facelets
 - A. Korzyści wynikające z technologii
 - B. Mechanizmy szablonowania
 - C. Wbudowane znaczniki
 - D. Tworzenie komponentów złożonych (Composite Components)
- iv. Historia - JSF w oparciu o JSP
- v. JSF a REST
- vi. Wsparcie dla HTML5
- VIII. Model konwersji
 - i. Konwertery wbudowane
 - ii. Tworzenie konwerterów
- IX. Model walidacji
 - i. Walidatory wbudowane
 - ii. Tworzenie walidatorów
 - iii. Walidator domyślny
 - iv. Bean Validation 1.1 a JSF
 - v. Zdarzenia związane z walidacją
 - vi. Realizacja nietypowych/złożonych scenariuszy walidacji
- X. Model zdarzeń
 - i. Zdarzenia i słuchacze – podstawy modelu zdarzeniowego
 - ii. Zdarzenia akcji i zmiany wartości
 - iii. Zdarzenia zmiany faz
 - iv. Zdarzenia systemowe
- XI. Model nawigacji
 - i. Nawigacja „klasyczna”
 - ii. Nawigacja skrócona
 - iii. Nawigacja warunkowa
 - iv. Dynamiczna konfiguracja nawigacji
 - v. Pull i push MVC
- XII. Główne składowe aplikacji JSF i internacjonalizacja
- XIII. Typowe problemy przy tworzeniu interfejsu min
 - i. Formularz w postaci tabelki
 - ii. „Inteligentna” paginacja
 - iii. Master-Detail, LOV
- XIV. Obsługa błędów
- XV. Zasoby aplikacji
 - i. Żądanie o zasoby (Resource Request) i jego obsługa
 - ii. Zależności od zasobów
 - iii. Znaczniki i obiekty EL
 - iv. Pakowanie i internacjonalizacja zasobów
- XVI. Wzbogacanie komponentów – Behaviour
- XVII. Tworzenie komponentów „od zera”

- i. Tworzenie komponentów
 - ii. Tworzenie rendererów
 - iii. Tworzenie znaczników
- XVIII. Pakowanie i konfiguracja aplikacji, fazy projektu
- XIX. Testowanie interfejsu zbudowanego na JSF (JSFUnit/Selenium)
- XX. Zabezpieczanie aplikacji
- 7. Tworzenie aplikacji WWW z użyciem technologii AJAX i JSF
 - I. Koncepcja AJAX i modele komunikacji
 - II. Przegląd podejść do tworzenia aplikacji AJAX (warstwowe – JSF, klient-serwer – GWT, ...)
 - III. Tworzenie aplikacji AJAX „od zera” (XMLHttpRequest)
 - IV. JSON
 - i. Wprowadzenie
 - ii. Java API for JSON Processing
 - iii. JSON w RESTFull webservice
 - V. Podstawy bibliotek Prototype i JQuery w kontekście manipulacji drzewem DOM i komunikacji AJAX
 - VI. „Natywne” podejście JSF2
 - i. Znacznik f:ajax i Ajax Behaviour
 - ii. Przetwarzanie żądań typu Ajax
 - iii. Przetwarzanie częściowe (Partial processing)
 - iv. Selektywne działania na drzewie komponentów
 - v. Kolejowanie
 - vi. Obsługa błędów
 - VII. Popularne biblioteki współpracujące z JSF
 - i. PrimeFaces
 - ii. OmniFaces
 - iii. RichFaces
 - iv. PrettyFaces
 - v. IceFaces
- 8. CDI (webbeans)
 - I. Podstawowe pojęcia
 - i. Ziarno
 - ii. Kontekst
 - iii. Zasięg
 - iv. Wstrzeliwanie zależności
 - II. Ziarna
 - i. Managed beans
 - ii. Session means
 - iii. Specjalizacja
 - iv. Aktywne i nieaktywne ziarna
 - III. Zasięgi
 - i. „Klasyczne” zasięgi wbudowane
 - ii. Zasięg konwersacji
 - iii. Zasięg zależny
 - iv. Tworzenie niestandardowych zasięgów



- v. Zasięg domyślny
- vi. Cykl życia obiektów kontekstowych
- IV. Wstrzykiwanie zależności
 - i. Miejsca docelowe wstrzeliwania
 - ii. Kwalifikatory, alternatywy, stereotypy
 - iii. Wstrzykiwanie po nazwie
- V. Metody fabrykujące
- VI. Model zdarzeń i słuchaczy
- VII. Użycie CDI w kontekście aplikacji EJB/JSF

