

Kod szkolenia: **SPRING/JSF**

Tytuł szkolenia: **Nowoczesne aplikacje internetowe oparte na Spring Framework, JSF, JPA i AJAX**

Dni: 5

## Opis:

### Cel szkolenia:

Celem szkolenia jest poznanie zasad budowania aplikacji internetowych w oparciu o Spring Framework wraz z zagadnieniami takimi jak: mapowanie obiektowo-relacyjne i tworzenie warstwy dostępu do danych za pomocą JPA, tworzenie warstwy usług przy pomocy kontenera Spring IoC, oprogramowanie frontendu web dzięki JSF.

W szczególności uczestnicy:

- poznają wzorzec Dependency Injection i nauczą się stosować go w praktyce przy użyciu kontenera IoC Spring Framework
- dowiedzą się jak konfigurować komponenty z poziomu deskryptorów XML Spring Framework i za pomocą adnotacji Java
- poznają zasadę mapowania relacyjno-obiektowego, 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
- nauczą się budować bogate interfejsy użytkownika w oparciu o Java Server Faces z użyciem JSF i faceletów
- nauczą się wykorzystywać gotowe komponenty biblioteczne JSF jak również tworzyć własne
- dowiedzą się jak zwiększyć wygodę interfejsu użytkownika za pomocą AJAX
- poznają zaawansowane zagadnienia takie jak AOP
- dowiedzą się jak testować ww. komponenty
- poznają zagadnienia związane z bezpieczeństwem i niezawodnością aplikacji internetowych
- dowiedzą się jak uruchamiać stworzone w powyższy sposób aplikacje na serwerach Tomcat i Jetty

### Adresaci Szkolenia:

Szkolenie adresowane jest do programistów Java pragnących dowiedzieć się jak tworzyć bezpieczne aplikacje internetowe w oparciu o Spring Framework oraz warstwę dostępu do

danych realizowaną przez Hibernate.

## Mocne strony szkolenia:

Program obejmuje całościowo i wyczerpująco zagadnienia tworzenia aplikacji internetowych. W przeciwieństwie do większości szkoleń, szkolenie 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 w miarę potrzeb. Program jest ciągle uaktualniany tak, by uwzględnić nowo powstające trendy.

## Wymagania:

Od uczestników szkolenia wymagana jest umiejętność programowania w języku Java (do poznania na kursie J/JP), podstawy relacyjnych baz danych i SQL, podstawowa znajomość HTML. Zalecana jest również umiejętność tworzenia aplikacji webowych w technologiach Servlet/JSP (do poznania na kursie J/WEB1).

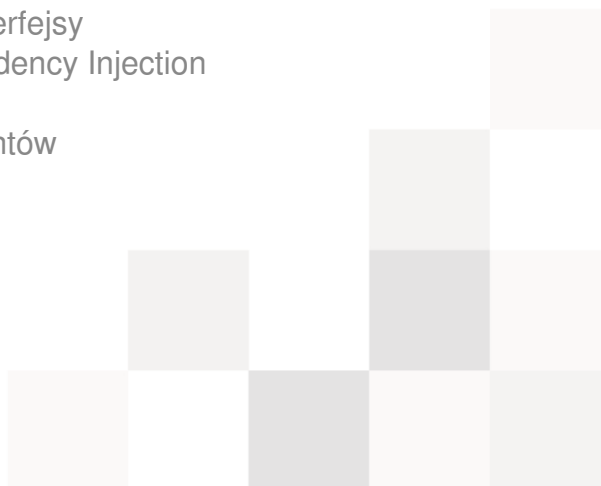
## Parametry szkolenia:

5\*8 godzin (5\*7 godzin 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. Wprowadzenie
  - I. Warstwowa architektura systemów informatycznych
  - II. Architektoniczne wzorce projektowe: IoC, DI, MVC
  - III. Warstwa dostępu do danych: JDBC, DAO, JPA
  - IV. Warstwa webowa: HTML, AJAX
  - V. Zagadnienia integracji systemów
  - VI. Programowanie zorientowane aspektowo: AOP
  - VII. Zagadnienia związane z testowaniem i bezpieczeństwem
2. Kontener obiektów – Spring IoC
  - I. Architektura komponentowa aplikacji
  - II. Programowanie zorientowane na interfejsy
  - III. Wzorce Inversion of Control i Dependency Injection
  - IV. Inicjalizacja kontenera
  - V. Definiowanie i inicjalizacja komponentów
    - i. Konstruktory, fabryki
    - ii. Własności proste
    - iii. Własności złożone, kolekcje
    - iv. Leniwa inicjalizacja



- VI. Zasięg komponentów
  - i. singleton, prototype
  - ii. zasięgi własne
- VII. Zależności pomiędzy komponentami
- VIII. Cykl życia komponentów
- IX. Rozszerzanie funkcjonalności kontenera
- X. Programowanie zdarzeniowe
- XI. Konwersja danych w Spring
  - i. Edytory właściwości
  - ii. Konwersja typów
  - iii. Formatowanie
  - iv. Specyfikowanie ograniczeń
- XII. Spring Expression Language
  - i. Przegląd możliwości
  - ii. Składnia
  - iii. Użycie
- 3. Podstawy Spring AOP
  - I. Pojęcie proxy
  - II. Podstawowa terminologia: aspect, join point, advice, pointcut
  - III. Konfiguracja
  - IV. Zastosowanie w kontenerze Spring
  - V. Zastosowanie we własnych
  - VI. Integracja z AspectJ
- 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
      - 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. 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
      - a. Pola, obiekty i grafy obiektów
      - b. Grupy i sekwencje
      - c. 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/dektop
  - 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. Integracja kontenera Spring z warstwą trwałości
  - I. Warstwa DAO
    - i. Klasyczne wsparcie dla budowy warstwy DAO
    - ii. Spring Data - nowe podejście do realizacji warstwy trwałości w środowisku Spring
    - iii. Spring Data JPA – specjalizowana wersja dla JPA
  - II. Transakcje
    - i. Poziomy izolacji transakcji
    - ii. Podsystem transakcji w Spring
    - iii. Deklaratywna specyfikacja transakcji
    - iv. Blokowanie pesymistyczne i optymistyczne
    - v. Długie konwersacje
  - III. Ujednoczona obsługa wyjątków
- 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. Warstwa web – nawigacja za pomocą Spring WebFlow
  - I. Konfiguracja
  - II. Definiowanie przepływów
  - III. Język wyrażen WebFlow Expression Language
  - IV. Renderowanie widoków
  - V. Akcje
  - VI. Persystentne przepływy
  - VII. Bezpieczeństwo przepływów
  - VIII. Dziedziczenie przepływów
  - IX. Integracja z JSF
  - X. Testowanie przepływów
- 9. Wprowadzenie do Spring Security
  - I. Wprowadzenie – przegląd możliwości
  - II. Konfiguracja
  - III. Bezpieczeństwo warstwy web
    - i. Rodzaje uwierzytelniania
    - ii. Opcje dodatkowe
    - iii. Poufność kanału HTTP/HTTPS



- iv. Zarządzanie sesją
  - v. Wsparcie dla zewnętrznych mechanizmów uwierzytelniania
  - vi. Definiowanie własnych filtrów
- IV. Zabezpieczanie metod komponentów kontenera Spring

