

Kod szkolenia: **SPRING/WKT**

Tytuł szkolenia: **Nowoczesne aplikacje internetowe oparte na Spring, Hibernate, Apache Wicket i Jetty**

Dni: 5

Opis:

Adresaci Szkolenia:

Szkolenie adresowane jest do programistów Java pragnących dowiedzieć się jak tworzyć aplikacje internetowe w oparciu o komponentowy model interfejsu użytkownika wspomagany o warstwę logiki biznesowej realizowanej przez Spring Framework korzystającej z dostępu do bazy danych poprzez mechanizm ORM (Hibernate).

Cel szkolenia:

Celem szkolenia jest poznanie zasad budowania aplikacji internetowych z wykorzystaniem takich zagadnień jak: mapowanie obiektowo-relacyjne i tworzenie warstwy dostępu do danych za pomocą Hibernate, tworzenie warstwy usług przy pomocy kontenera Spring IoC, oprogramowanie interfejsu użytkownika dzięki komponentowemu frameworkowi Wicket.

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 Hibernate 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 interfejs użytkownika przy pomocy Wicket
- nauczą się wykorzystywać gotowe komponenty biblioteczne Wicket jak również tworzyć własne
- dowiedzą się jak zwiększyć wygodę interfejsu użytkownika za pomocą AJAX
- poznają zagadnienia związane z bezpieczeństwem i niezawodnością aplikacji internetowych
- dowiedzą się jak uruchamiać stworzone w powyższy sposób aplikacje na serwerze Jetty

Mocne strony szkolenia:

Program obejmuje całościowo 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.

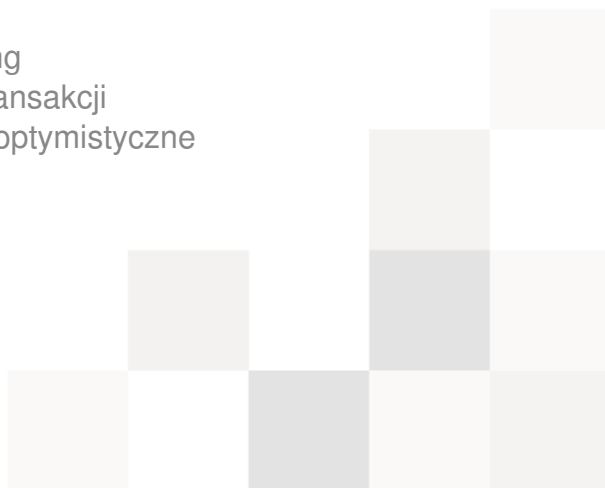
Wielkość grupy: maks. 8-10 osób.

Program szkolenia:

1. Wprowadzenie - przegląd omawianych zagadnień
 - I. Warstwowa architektura aplikacji internetowych
 - II. Wzorce: IoC, DI, MVC
 - III. Warstwa danych: JDBC, myBatis, Hibernate, JPA
 - IV. Warstwa webowa: JSF, AJAX, Wicket
 - V. Zagadnienia integracji systemów, webserwisy
 - VI. Programowanie zorientowane aspektowo: AOP
 - VII. Zagadnienia związane z testowaniem i bezpieczeństwem
2. Warstwa logiki biznesowej – Spring IoC
 - I. Wzorce Inversion of Control i Dependency Injection
 - II. Struktura aplikacji w Spring Framework
 - III. Uruchamianie kontenera
 - IV. Definiowanie komponentów
 - i. Konstruktory, fabryki
 - ii. Własności proste
 - iii. Własności złożone, kolekcje
 - iv. Łączenie kolekcji
 - v. Dziedziczenie
 - vi. p-namespace
 - V. Specyfikacja zależności



- VI. Leniwa inicjalizacja
 - VII. Autowiring
 - VIII. Zasoby
 - IX. Współpraca z kontenerem
 - X. Zasięg komponentów
 - i. singleton, prototype, request, session
 - ii. zasięgi własne
 - XI. Wstrzykiwanie metod
 - XII. Cykl życia fasolki i nasłuchiwanie
 - XIII. Zamykanie kontenera
 - XIV. Wydzielanie części konfiguracji do plików z właściwościami
 - XV. Lokalizacja
3. Warstwa danych – Spring ORM i Hibernate
- 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
 - i. Jednokierunkowe
 - ii. Dwukierunkowe
 - iii. Propagacja kaskadowa
 - IV. Klucze
 - i. Proste
 - ii. Złożone
 - iii. Automatyczne generowanie kluczy
 - V. Kompozycja obiektowa
 - VI. Dziedziczenie
 - VII. Nazewnictwo obiektów w bazie danych, własne systemy nazw
 - VIII. Kontekst trwałości:
 - i. Zarządzany przez kontener i przez aplikację
 - ii. Konfiguracja kontekstu trwałości
 - iii. Integracja ze Spring Framework
 - iv. Konfiguracja DAO
 - IX. Cykl życia encji
 - X. Operacje na danych
 - XI. Transakcje
 - i. Poziomy izolacji transakcji
 - ii. Podsystem transakcji w Spring
 - iii. Deklaratywna specyfikacja transakcji
 - iv. Blokowanie pesymistyczne i optymistyczne
 - v. Długie konwersacje
 - XII. Zapytania HQL
 - XIII. Criteria API
 - XIV. Nasłuchiwanie cyklu życia encji



- XV. Zagrożenia
- 4. Warstwa web – Wicket
 - I. Podstawowa architektura aplikacji
 - i. Konfiguracja
 - ii. Przetwarzanie requestów
 - iii. Rola kodu java
 - iv. Rola html
 - II. Koncepcja modelu danych
 - i. Standardowy model
 - ii. Kontrola zasobów – detachable model
 - iii. Zagnieżdżanie modeli
 - III. Podstawowe komponenty UI
 - i. Wyświetlanie tekstu
 - ii. Obsługa linków
 - iii. Repeaters
 - iv. Modyfikacja atrybutów komponentu
 - IV. Obsługa formularzy
 - i. Budowa formularza
 - ii. Komponenty formularza
 - iii. Obsługa submit
 - iv. Walidacja danych
 - v. Informacje zwrotne
 - V. Grupowanie komponentów
 - i. Dziedziczenie szablonów
 - ii. Panel
 - iii. Border
 - iv. Fragment
- 5. Ajax w Wicket
 - I. Obsługa Ajax w Wicket
 - i. Komponenty z wbudowaną obsługą Ajax
 - ii. Dodawanie obsługi Ajax do własnych komponentów
 - iii. Obsługa formularzy za pomocą Ajax
- 6. Tworzenie własnych komponentów
 - I. Korzyści z tworzenia własnych komponentów
 - i. Praca z zasobami (css, skrypty)
 - ii. Rozszerzanie możliwości komponentów
 - iii. Własne komponenty formularzy
 - iv. Dystrybucja komponentów

