

## Program szkolenia:

# Aplikacje Web oparte o Spring Framework

### Informacje:

<b>Nazwa:</b>	<b>Aplikacje Web oparte o Spring Framework</b>
<b>Kod:</b>	<b>Spring -Spring</b>
<b>Kategoria:</b>	Spring Framework
<b>Grupa docelowa:</b>	developerzy architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

---

Koncepcja szkolenia została oparta o zestaw praktycznych problemów jakie stają przed developerem – prezentujemy sprawdzone i najlepsze ich rozwiązania w Spring.

Szkolenie swoim zakresem wykracza daleko poza powszechnie dostępne materiały dydaktyczne. Szkolenie oprócz prezentacji technologii Spring zostało wzbogacone o aspekty: doboru architektury aplikacji, konfiguracji narzędzi developerskich, praktyk projektowych odpowiednich do produktywnej pracy z Spring, istotnych zaawansowanych praktycznych aspektów JPA/Hibernate.

Wiedza zdobyta podczas szkolenia pozwoli na pełne wykorzystanie frameworka, zwiększenie rozszerzalności systemów oraz racjonalizację procesu testowania - zwiększając jakość z jednoczesną redukcją kosztów utrzymania testów.

Program szkolenia stanowi bazę do której możemy dodawać dowolnie wybrane zagadnienia ze szkolenia [Spring-moduły](#).

### Projekt referencyjny

Sprawdź naszą implementację przykładowego projektu DDD+CqRS opartego o Spring Framework: [Sample Leaven](#).

### Zalety szkolenia:

- Dobór architektury aplikacji
- Wzorce i pułapki
- Nacisk na testability - architektura wspierająca testy
- Elementy Domain Driven Design

## Szczegółowy program:

### 1. Warstwowa architektura systemów opartych o Spring

#### 1.1. Wzorce

##### 1.1.1. Wzorce architektoniczne

##### 1.1.2. Wzorce projektowe stosowane w aplikacjach webowych

##### 1.1.3. Wzorce porządkowania logiki biznesowej

#### 1.2. Architektura dla podejścia Domain Driven Design i CqRS

#### 1.3. Architektura zorientowana na testy

### 2. Kontener

#### 2.1. Konfiguracja

#### 2.2. Techniki uruchamiania

#### 2.3. Zasada działania – Inversion of Control (w szczególności Dependency Injection)

### 3. Komponenty

#### 3.1. Deklaracja - adnotacje i XML (dobór właściwego podejścia w zależności od problemu)

#### 3.2. Cykl życia

#### 3.3. Zależności

#### 3.4. Zasięg komponentów – pułapki

#### 3.5. Pre/post – procesory

##### 3.5.1. Wykorzystanie dla technik Domain Driven Design

### 4. Paradigmat Inversion of Control - przegląd 3 podejść

#### 4.1. Dependency Injection

##### 4.1.1. Wykorzystanie do zmniejszenia poziomu zależności

#### 4.2. Praktyczne przykłady modularyzacji aplikacji biznesowych - wstrzykiwanie strategii (polityk) biznesowych

#### 4.3. Zdarzenia

4.3.1. Zdarzeniowe architektury otwarte na rozbudowę

4.3.2. Asynchroniczne przetwarzanie zdarzeń jako technika zwiększająca skalowalność

4.3.3. Praktyczne przykłady generowania zdarzeń biznesowych z warstwy logiki

4.4. Techniki Aspect Oriented Programming

4.4.1. Podstawy teoretyczne

4.4.2. Praktyczne przykłady wykorzystania w aplikacjach biznesowych

4.4.3. Zastosowanie w Spring

4.4.4. Zastosowanie do budowy własnych frameletów

## 5. Warstw dostępu do danych

5.1. Transakcje (konfiguracja, poziomy izolacji, warstwa abstrakcji Spring)

## 6. Funkcjonalności kontenera

6.1. Zarządzanie zasobami

6.2. Język wyrażeń

## 7. Spring Remoting

7.1. Zastosowanie (applety, aplikacje StandAlone)

7.2. Konfiguracja

7.3. Najlepsze praktyki

## 8. Bezpieczeństwo - Spring Security (dawniej ACEGI)

## 9. Testowanie

9.1. Spring - wsparcie dla testów

9.2. Testowanie jednostkowe - techniki mockowania

9.3. Testowanie integracyjne - wsparcie kontenera

## 10. Narzędzia: Spring Tools Suite