

## Program szkolenia:

# Modelowanie nowoczesnych systemów w kontekście Spring Framework

## Informacje:

<b>Nazwa:</b>	<b>Modelowanie nowoczesnych systemów w kontekście Spring Framework</b>
<b>Kod:</b>	<b>Spring -modeling</b>
<b>Kategoria:</b>	Spring Framework
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	30% wykłady, 70% warsztaty

Zrób krok w kierunku zostania Technicznym Partnerem dla Biznesu i dowiedz się, jak wykorzystać możliwości najpopularniejszego frameworka Java w kontekście tworzenia aplikacji najwyższej jakości.

Szkolenie o warsztatowym charakterze, w trakcie którego razem z trenerem zbudujesz dobrze zaprojektowaną aplikację przy zastosowaniu praktyk DDD, TDD oraz Clean Architecture. Serce aplikacji zostanie zintegrowane ze Spring Framework jedynie w niezbędnych miejscach, zachowując w ten sposób czystość architektury i modułów domenowych.

To szkolenie nie ma na celu omówienia wszystkich mechanizmów i modułów Spring Framework. Zamiast tego, otrzymasz praktyczne wskazówki i wypróbujesz techniki umożliwiające budowanie łatwych w modyfikacji aplikacji najwyższej klasy. Uwaga: Spring pojawi się dopiero wtedy, gdy uznamy to za konieczne.

Szkolenie składa się z dwóch części:

- Serce aplikacji - uniwersalne techniki wytwarzania dobrego oprogramowania ze szczególnym implementacyjnym w postaci Spring Framework
  - Techniki Knowledge Crunching
  - DDD
  - TDD
  - Dobór architektury
  - Modularny monolit
- Deep dive into Spring Framework
  - Szczegóły działania i internale
  - Pułapki
  - Najlepsze praktyki
  - Testy integracyjne
  - Projekty Spring Framework

## Zalety szkolenia:

- Całościowe podejście
- Orientacja architektury na drivery biznesowe
- Pragmatyczne podejście

## Szczegółowy program:

### 1. Część 1 - serce aplikacji

1.1. Wprowadzenie do przykładowej domeny

1.2. Knowledge crunching

1.2.1. Wizja produktu

1.2.2. Big Picture

1.2.3. Odkrywanie krytycznych procesów

1.2.4. Example mapping

1.3. Modelowanie kodu domenowego za pomocą TDD w formule mob programming - Code Storming

1.4. Zasady doboru architektury

1.5. Przykład zastosowania architektury Ports and Adapters

1.6. Testy integracyjne

1.7. Przygotowanie aplikacji jako modularnego monolitu

1.7.1. Dobór architektury do modułu

1.7.2. Wzorce komunikacji między modułami

1.8. Wprowadzenie do Spring Framework jako szczegółu implementacyjnego wybranych modułów

1.9. Organizacja modułów w oparciu o Spring Framework

### 2. Część 2 - Deep dive into Spring Framework

2.1. Kontener

2.1.1. Zróbmy swój własny kontener (opcjonalnie)

2.1.2. Konfiguracja

2.1.3. Implementacje kontenera

2.1.4. Hierarchie

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

2.2. Komponenty

2.2.1. Deklaracja - Java config i adnotacje

2.2.2. Cykl życia

2.2.3. Zależności

2.2.4. Zasięg komponentów

2.2.4.1. Pułapki Scoped Proxy

2.2.4.2. Wstrzykiwanie krócej żyjących komponentów do dłużej żyjących

2.2.5. Pułapki adnotacji (np. @Transactional)

2.2.6. Pre/post – procesory

2.3. AoP

2.4. Spring Data

2.4.1. JPA

2.4.2. MongoDB

2.4.3. JDBC

2.5. Tworzenie prawdziwego RESTful API wg Richardson Maturity Model

2.6. Spring Security

2.7. Wsparcie dla programowania asynchronicznego

2.8. Wsparcie dla programowania zdarzeniowego

2.9. Spring Boot

2.10. Wprowadzenie do Spring Cloud