

## Program szkolenia:

# Testowanie aplikacji w Javie

### Informacje:

<b>Nazwa:</b>	<b>Testowanie aplikacji w Javie</b>
<b>Kod:</b>	<b>Java-test</b>
<b>Kategoria:</b>	Java i JVM
<b>Odbiorcy:</b>	architekci, developerzy, liderzy techniczni
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	30% wykłady / 70% warsztaty

---

Szkolenie wprowadza uczestników do testów jednostkowych, omawiając ich znaczenie i różne typy. Kursanci uczą się pisać dobrze zorganizowane testy, używać mocków i stubów oraz testować kod wielowątkowy.

Uczestnicy poznają również zasady testowania integracji z różnymi komponentami, takimi jak frameworki IoC, serwery HTTP, systemy wiadomości i bazy danych.

W ramach szkolenia testowana będzie również architektura aplikacji, API w ramach testów kontraktowych, przypadki brzegowe oraz pisane będą testy oparte na właściwościach.

#### **Uczestnicy szkolenia nauczą się:**

- efektywnie pisać i strukturyzować testy jednostkowe
- tworzyć testy integracyjne dla różnych typów integracji
- stosować zaawansowane techniki testowania architektury i API

### Zalety szkolenia:

- szkolenie prowadzone przez współtwórcę narzędzi do testowania
- kompleksowe omówienie testów jednostkowych, integracyjnych i zaawansowanych technik testowania
- aktualne technologie i zaawansowane techniki takie jak testowanie architektury, testy mutacyjne i testy oparte o właściwości

## Szczegółowy program:

### 1. Dzień 1: Testy jednostkowe

- 1.1. Wprowadzenie do testowania
- 1.2. Pisanie dobrych testów jednostkowych
- 1.3. Praca z mockami i stubbami
- 1.4. Testowanie wyjątków i kodu wielowątkowego

### 2. Dzień 2: Testy integracyjne

- 2.1. Wprowadzenie do testów integracyjnych
- 2.2. Testowanie integracji z frameworkiem IoC
- 2.3. Testowanie integracji HTTP
- 2.4. Testowanie integracji wiadomości
- 2.5. Testowanie integracji z bazą danych

### 3. Dzień 3: Zaawansowane testy

- 3.1. Testowanie architektury aplikacji
- 3.2. Testowanie API za pomocą testów kontraktowych
- 3.3. Testowanie przypadków brzegowych za pomocą testów mutacyjnych
- 3.4. Testowanie oparte na właściwościach