

## Program szkolenia:

# Refaktoryzacja kodu legacy

### Informacje:

<b>Nazwa:</b>	<b>Refaktoryzacja kodu legacy</b>
<b>Kod:</b>	<b>legacy-refactor</b>
<b>Kategoria:</b>	Refaktoryzacja kodu legacy
<b>Grupa docelowa:</b>	
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	40% wykłady / 60% warsztaty

---

Podczas szkolenia uczestnicy poznają techniki umożliwiające refaktoryzację istniejących systemów w celu zmniejszenia nakładu pracy związanego z ich utrzymaniem.

Dodatkowym celem będzie poznawanie zachowania i struktury w celu naprawiania błędów i wprowadzania nowych funkcjonalności oraz refaktoryzacja designu.

Podczas warsztatów praktycznych uczestnicy będą refaktoryzować istniejący kod i pisać testy regresyjne. Uczestnicy zrefaktoryzują istniejący system pod kątem wprowadzenia nowych funkcjonalności.

### Zalety szkolenia:

- Prowadzący jest pasjonatem archeologii kodu
- Wybór najbardziej użytecznych wzorców oraz technik
- Realne przykłady

## Szczegółowy program:

### 1. Po co i kiedy refaktoryzować

1.1. Analiza kodu

1.2. Metryki kodu

1.3. Wizja kodu

### 2. Refaktoryzacja klas

2.1. Code Smells na poziomie klas

2.2. Zwiększanie czytelności kodu

2.3. Techniki refaktoryzacji metod

2.4. Testy jednostkowe jako weryfikacja zachowania

### 3. Refaktoryzacja modułów

3.1. Code Smells na poziomie modułów

3.2. Refaktoryzacja do wzorców

3.2.1. Adapter

3.2.2. Builder

3.2.3. Strategy

3.2.4. Template Method

3.2.5. Decorator

3.2.6. Factory Method

3.2.7. Observer

3.2.8. State

### 4. Testy integracyjne jako weryfikacja zachowania

4.1. Refaktoryzacja systemów

4.2. Refaktoryzacja architektury

#### 4.3. Techniki podziału aplikacji an moduły

4.3.1. Refaktoryzacja do wzorców architektonicznych

4.3.2. Ports and Adapters

4.3.3. Model-View-Controller

4.3.4. CQRS

#### 4.4. Testy akceptacyjne jako weryfikacja zachowania

### 5. Refaktoryzacja testów

5.1. Code Smells w testach

5.2. F.I.R.S.T. tests

5.3. Refaktoryzacja do wzorców

5.3.1. Assembler

5.3.2. Business Assert

5.3.3. Page Object