

SZKOLENIE ŚREDNIO ZAAWANSOWANE

Praca z kodem odziedziczonym (Legacy Code)

LEGACY

Czas trwania: 3 dni

Cele szkolenia

- Zapoznanie uczestników z zagrożeniami oraz problemami wynikającymi z pracy z kodem odziedziczonym oraz sposobu ich przewyżczenia
- Wypracowanie algorytmu pracy z kodem Legacy
- Nauka pracy i wprowadzania zmiany w LC tak aby zminimalizować ryzyko wprowadzenia błędów
- Poznanie technik refaktoryzacji, pozwalających stopniowo podnosić jakość kodu odziedziczonego

Zalety

- Szkolenie prowadzone jest przez trenerów, którzy mają duże doświadczenie w pracy z kodem odziedziczonym. Przeprowadzają oni krok po kroku uczestników przez tajniki pracy z LC
- Zdecydowana przewaga zajęć warsztatowych pozwala uczestnikom zmierzyć się z nietrywialnymi przykładami LC (pod baczny okiem trenera), nabyć praktyczne umiejętności refaktoryzacji i testowania trudnego kodu

Dla kogo?

- Szkolenie adresowane jest do programistów pracujących na co dzień z kodem odziedziczonym (Legacy Code) lub kodem drogim w utrzymaniu (w którym wprowadzenie drobnych zmian zajmuje wiele wysiłku i/lub skutkuje pojawieniem się wielu błędów)

Wymagania

- Od uczestników szkolenia wymagana jest umiejętność programowania w języku Java oraz znajomość frameworków do testowania jednostkowego (np. JUnit, TestNG)



Program

1. Techniki refaktoryzacji
 - a. Czym jest refaktoryzacja
 - b. Kiedy (nie) refaktoryzować
 - c. Zapachy w kodzie (Code Smells)
 - d. Katalog refaktoryzacji
 - e. Refaktoryzacja do wzorców
 - f. Wykorzystanie IDE
2. Kod odziedziczony (Legacy Code)
 - a. Czym jest Legacy Code
 - b. Ryzyka wynikające z pracy z LC
 - c. Dlaczego praca z LC jest taka trudna
 - d. Metody pracy z LC (Edit and Pray vs. Cover and Modify)
 - e. Algorytm pracy z LC
3. Szwy (Seam Model)
 - a. Czym są szwy
 - b. W jaki sposób szwy pomagają testować
 - c. Jak znajdować szwy
4. Techniki łamania zależności (Dependency Breaking)
 - a. Jakie zależności są najbardziej kłopotliwe
 - b. Jak łamać zależności
 - c. Katalog technik łamania zależności
 - Ukryte i globalne zależności
 - Olbrzymie metody
 - Kłopotliwe zależności do zewnętrznych bibliotek
 - Zbyt wiele odpowiedzialności
 - d. Skutki łamania zależności
5. Analiza efektów (Effect Sketches)
 - a. Wnioskowanie o efektach wprowadzanych zmian
 - b. Znajdowanie najbardziej optymalnych punktów testowania (Interception Points)
6. Testy charakteryzacyjne (Characterization Tests)
 - a. W jaki sposób wprowadzać zmiany aby nie wprowadzać błędów
 - b. Jak dobrać kontekst testów charakteryzacyjnych
7. Wzorce pracy z LC (Reengineering Patterns)

