

Kod szkolenia: **LEGACY**

Tytuł szkolenia: **Praca z kodem odziedziczonym (Legacy Code)**

Dni: 3

Partner merytoryczny



Opis:

Adresaci szkolenia

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).

Cel szkolenia

Celem szkolenia jest zapoznanie uczestników z zagrożeniami oraz problemami wynikającymi z pracy z kodem odziedziczonym oraz sposobu ich przewyciężenia.

Podczas szkolenia wypracowany zostaje algorytm pracy z kodem Legacy. Uczestnicy uczą się jak pracować i wprowadzać zmiany w LC tak aby zminimalizować ryzyko wprowadzenia błędów. Uczestnicy uczą się technik refaktoryzacji, które pozwalają stopniowo podnosić jakość kodu odziedziczonego.

Mocne strony

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 bacznym okiem trenera), nabyć praktyczne umiejętności refaktoryzacji i testowania trudnego kodu.

Wymagania:

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

Parametry szkolenia



3*8 godzin (3*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Wielkość grupy: maks. 8-10 osób.

Program szkolenia:

1. Techniki refaktoryzacji:
 - czym jest refaktoryzacja,
 - kiedy (nie) refaktoryzować,
 - zapachy w kodzie (Code Smells),
 - katalog refaktoryzacji,
 - refaktoryzacja do wzorców,
 - wykorzystanie IDE.
2. Kod odziedziczony (Legacy Code):
 - czym jest Legacy Code,
 - ryzyka wynikające z pracy z LC,
 - dlaczego praca z LC jest taka trudna,
 - metody pracy z LC (Edit and Pray vs. Cover and Modify),
 - algorytm pracy z LC.
3. Szwy (Seam Model):
 - czym są szwy,
 - w jaki sposób szwy pomagają testować,
 - jak znajdować szwy.
4. Techniki łamania zależności (Dependency Breaking):
 - jakie zależności są najbardziej kłopotliwe,
 - jak łamać zależności,
 - katalog technik łamania zależności,
 - ukryte i globalne zależności,
 - olbrzymie metody,
 - kłopotliwe zależności do zewnętrznych bibliotek,
 - zbyt wiele odpowiedzialności.
 - skutki łamania zależności.
5. Analiza efektów (Effect Sketches):
 - wnioskowanie o efektach wprowadzanych zmian,
 - znajdowanie najbardziej optymalnych punktów testowania (Interception Points).
6. Testy charakteryzacyjne (Characterization Tests):
 - w jaki sposób wprowadzać zmiany aby nie wprowadzać błędów,
 - jak dobrać kontekst testów charakteryzacyjnych.
7. Wzorce pracy z LC (Reengineering Patterns).

