

Testy jednostkowe w .NET

Tworzenie precyzyjnych i skutecznych testów w środowisku .NET

Cele szkolenia

Celem szkolenia jest nabycie wiedzy na temat tworzenia precyzyjnych oraz skutecznych testów jednostkowych w celu podniesienia niezawodności i jakości kodu.

Umiejętności

Dzięki szkoleniu uczestnik będzie:

- Tworzył testy zgodnie w cyklu Red-Green-Refactor
- Projektował przypadki testowe, odnajdywał wartości graniczne oraz wyodrębniał kluczowe ścieżki funkcjonalne
- Tworzył testy jednostkowe w frameworku NUnit/xUnit
- Parametryzował testy oraz tworzyć testy wielokrotnego użytku
- Tworzył atrapy obiektów i funkcjonalności
- Przygotowywał istniejący kod pod testy jednostkowe.

Profil uczestników

Szkolenie przeznaczone jest dla osób, które znają podstawy programowania w języku C#, rozumieją paradygmat obiektowości oraz są w stanie napisać prostą aplikację w frameworku .net.

Szkolenie dla informatyków, testerów automatyzujących, testerów manualnych, programistów.

Przygotowanie uczestników

Jeśli użytkownik na co dzień nie programuje w języku C#, powinien przypomnieć sobie tworzenie metod oraz klas w .NET, uruchamianie oraz debugowanie aplikacji.

Szczegółowy program szkolenia

1. *Test-Driven Development*
 - 1.1. *Wprowadzenie do TDD*
 - 1.2. *Cykl Red-Green-Refactor*
 - 1.3. *Zasady FIRST*
 - 1.4. *Metoda AAA*

- 1.5. *Uruchamianie i debugowanie testów jednostkowych*
2. *Biblioteka NUnit / xUnit*
 - 2.1. *Instalacja biblioteki NUnit/xUnit*
 - 2.2. *Prawidłowe nazewnictwo testów jednostkowych*
 - 2.3. *Weryfikacja wyniku*
 - 2.4. *Parametryzacja przypadków testowych*
 - 2.5. *Weryfikacja wartości liczbowych, tekstów oraz dat*
 - 2.6. *Weryfikacja kolekcji*
 - 2.7. *Weryfikacja zwracanego typu*
 - 2.8. *Weryfikacja metod void*
 - 2.9. *Weryfikacja wyjątków*
 - 2.10. *Weryfikacja zdarzeń*
 - 2.11. *Weryfikacja czasu wykonania*
 - 2.12. *Weryfikacja metody asynchronicznej*
3. *Biblioteka Fluent Assertions*
 - 3.1. *Instalacja biblioteki FluentAssertions*
 - 3.2. *Weryfikacja pustych wartości*
 - 3.3. *Weryfikacja zakresów liczbowych, daty i czasu*
 - 3.4. *Weryfikacja kolekcji i słowników*
 - 3.5. *Weryfikacja wyjątków*
 - 3.6. *Weryfikacja zdarzeń*
 - 3.7. *Weryfikacja czasu wykonania*
4. *Tworzenie atrap (Mock)*
 - 4.1. *Instalacja biblioteki Moq*
 - 4.2. *Utworzenie atrapy*
 - 4.3. *Definicja zachowania metody, właściwości, zdarzenia*
 - 4.4. *Weryfikacja wywołania metody, właściwości*
 - 4.5. *Linq To Mocks*
5. *Tworzenie kontenera IoC*
 - 5.1. *Instalacja kontenera DI*
 - 5.2. *Rejestrowanie zależności*
 - 5.3. *Wstrzykiwanie zależności*
 - 5.4. *Wykorzystanie kontenera w testach jednostkowych*
6. *Refaktoryzacja*
 - 6.1. *Dostosowanie kodu w celu przeprowadzenia testów jednostkowych*
 - 6.2. *Przydatne wzorce projektowe*

Metoda realizacji szkolenia

30% wykładu, 30% analizy przypadku, 40% ćwiczeń.

Liczba dni, liczba godzin szkoleniowych

3 dni, 21 godzin szkoleniowych

Ścieżka rozwoju po szkoleniu

- *Programowanie .NET – kurs zaawansowany*

- *Tworzenie aplikacji Internetowych z wykorzystaniem ASP.NET Core*