WI

Zaawansowane techniki programowania C#

Wykład 3

Piotr Błaszyński

Wydział Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego

9 stycznia 2018

TDD w praktyce

- Implementacja drzewa binarnego

TDD - Test Driven Development

- Kent Beck - publikacja książki 2002,
- nie powstało nagle,
- ciągle rozwijane,
- na początku jako fragment Extreme Programming, później oddzielnie,
- pojawia się coraz więcej narzędzi, na które wcześniej nie było mocy (wiedzy, itp.).

Testy jednostkowe

- Test jednostkowy (ang. unit test),
- Testy funkcjonalne,
- Testy integracyjne,
- zasadnicza różnica - złożoność

Testy jednostkowe

Zasadnicza różnica - złożoność:

- testy jednostkowe wykonują tylko małe fragmenty kodu,
- w jednym teście testujemy jedną ścieżkę wykonywania programu,
- powinny sie wykonywać szybko,
- niepowodzenie wykonania jednego testu powinno skutkować jak najmniejszą szkodą dla systemu (idealnie, jezeli jest do zrobienia tylko 1 prosta poprawka),

Testy jednostkowe - podejście praktyczne

- Tests first,
- „the intention was for the original coder to build them as he was building the application to capture his knowledge of how the code is supposed to work and what may break" softwareengineering.stackexchange.com:writing-tests-for-existing-code.

Testy jednostkowe - podejście praktyczne

- piszemy test jednostkowy dla aktualnie pisanej funkcjonalności,
  - między innymi wymusza to określenie podstaw interfejsu (parametry metod, wstępne nazwy klas),
- napisać jak najszybciej kod (najprościej jak to możliwe),
- poprawki/refaktoring kodu/refaktoring podejścia/ponowne testowanie
- pisanie kolejnych testów jeżeli konieczne dla lepszego pokrycia kodu, wyjątkowych przejść kodu,
- powtarzamy powyższe dla kolejnego elementu/funkcjonalności

Testy jednostkowe - podejście praktyczne

Inaczej:

- Red,
- Green,
- Refactor.

Biblioteki dostępne w C#

- NUnit,
- MSTest
- xUnit.

Tendencyjne (bo na stronie xUnit) porównanie frameworków

xUnit - podstawowe elementy

- [Fact] - oznaczenie metody z pojedynczym testem,
- [Theory]
  [NNData] - przydatne, jeżeli chcemy testować z różnymi danymi,
- Assert.Equal - coś ma pewną wartość,
- Assert.True - uniwersalne sprawdzenie wartości boolowskiej
- Assert.XYZ - inne sprawdzenia.

NUnit - podstawowe elementy

- [Test] - klasa z testami,
- [SetUp] - ustawienia dla wszystkich testów z klasy,
- [Theory]- przydatne, jeżeli chcemy testować z różnymi danymi,
- [TestFixture] - przydatne, jeżeli chcemy testować z różnymi danymi,
- Assert.That.Is.EqualTo - coś ma pewną wartość,
- Assert.That.Is.True - uniwersalne sprawdzenie wartości boolowskiej
- Assert.That.Is.XYZ - inne sprawdzenia.

MSTest - podstawowe elementy

- [TestClass] - klasa z testami,
- [TestInitialize] - ustawienia dla wszystkich testów z klasy,
- [TestMethod] - oznaczenie metody z pojedynczym testem,
- [[DataSource]] - przydatne, jeżeli chcemy testować z różnymi danymi,
- Assert.AreEqual - coś ma pewną wartość,
- Assert.IsTrue - uniwersalne sprawdzenie wartości boolowskiej
- Assert.XYZ - inne sprawdzenia.