Sprawozdanie z Test Driven Developement

Ambroziak Paweł 125 NCI B

Test Driven Development, czyli TDD, to technika tworzenia oprogramowania sterowana przez testy. Tworzenie kodu składa się z wielokrotnie wykonywanych trzech głównych kroków:

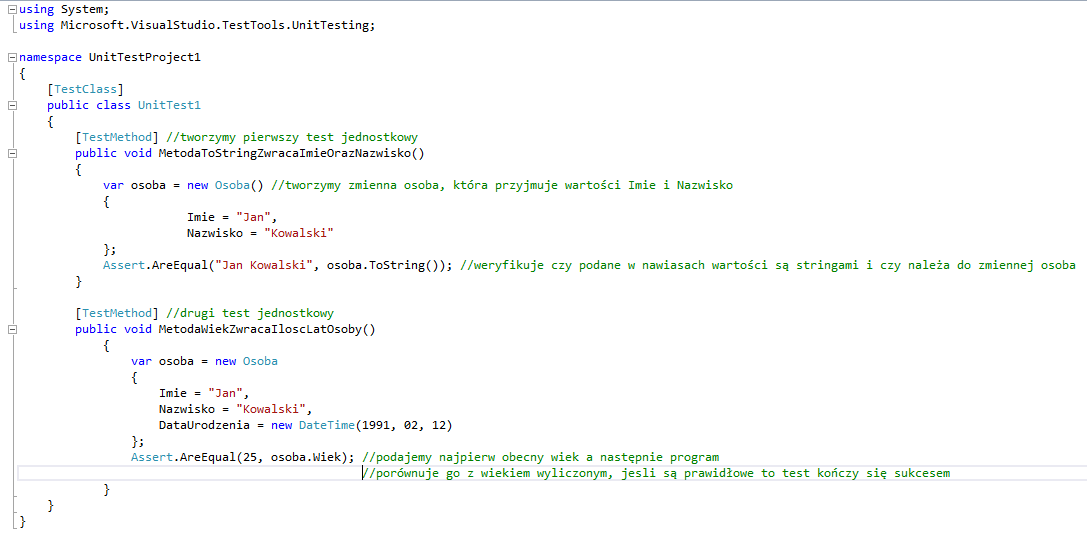
1. Stworzenie testu jednostkowego, który powinien być możliwie najprostszy, aby uniknąć możliwości popełnienia błędu w samym teście. Test ma sprawdzać funkcjonalność, która będzie implementowana w kroku 2.
2. Implementacja funkcjonalności – tworzymy funkcjonalność, którą chcemy zaimplementować. Funkcjonalność ta powinna spełniać założenia testu jednostkowego, a wykonanie testu jednostkowego powinno kończyć się sukcesem.
3. Refaktoryzacja, czyli porządki w stworzonej funkcjonalności. Ma to na celu uporządkowanie kodu, tak aby spełnione były standardy. Czynności wykonywane w tym kroku nie mogą zmienić wyniku testów.

W TDD wyróżniamy cztery główne testy:

* testy jednostkowe (*unit tests*) — testujemy pojedynczą, jednostkową część kodu: zazwyczaj klasę lub metodę;
* testy integracyjne (*integration tests*) — testujemy kilka komponentów systemu jednocześnie;
* testy regresyjne (*regression tests*) — po wprowadzeniu naszej zmiany uruchamiane są wszystkie testy w danej domenie biznesowej celem sprawdzenia czy zmiana nie spowodowała błędu w innej części systemu;
* testy akceptacyjne (*acceptance tests*) — testy mające na celu odpowiedzieć na pytanie czy aplikacja spełnia wymagania biznesowe.

Ważną rzeczą w testach jest również cykl Red/Green/Refactor:

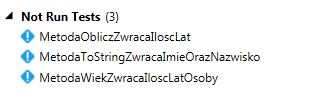
* Każdy test może się nie powieść, to zapewnia nas, ze testy naprawdę działają i mogą złapać błąd. Kiedy to jest pokazane możemy stworzyć funkcjonalność, dzięki której usuniemy błąd i test powiedzie się w 100%. To doprowadziło do wytworzenia mantry TDD, Red/Green/Refactor, gdzie Red oznacza błąd, Green oznacza pomyślne przejście testu.



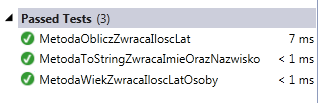
W tej sekcji programu tworzone są nasze testy jednostkowe. Mamy tutaj testy odpowiedzialne za sprawdzenie imienia i nazwiska, metoda obliczająca ilość lat oraz metoda wiek, która zwraca ilość lat.

Testy nie są samodzielnie wykonującym się programem, więc nie możemy ich normalnie uruchomić w środowisku Visual Studio. Służy do tego odpowiednia zakładka o nazwie Test. Właśnie tam puszczamy nasze testy do wykonania. Po chwili otrzymujemy wynik naszych testów.

Przed uruchomieniem Testów:



Po uruchomieniu Testów:



Repozytorium: https://github.com/PowerAmbroz/TDD.git