C# ta Birim Testleri ile Test Güdümlü Uygulama Geliştirme

(Test Driven Development)

Begüm Üner

Unit Test Nedir?

Unit test geliştirmiş olduğunuz kodu test etmek için yazdığınız koddur ve esas olarak kodun veya fonksiyonun belirli bir biriminin davranışını kontrol eder ve geliştirmeyi yapan developer tarafından yazılır.

Test Driven Development olarak ifade edeceğimiz TDD’yi Türkçe’ye Test Güdümlü Geliştirme olarak çevirebiliriz

TDD aslında çok basit birkaç temel adımdan oluşmaktadır:

* Bir test yazılır.
* Test başarısız olursa başarılı hale getirilir.
* Mevcut bütün testlerin başarılı olması sağlanır.
* Kod refactor edilir.

TDD’de her zaman önce test yazılır ardından geliştirme yapılır. Aksi durum aslında TDD’ye uygun bir geliştirme değildir. TDD kimi ekipler tarafından zaman kaybı olarak görünse de aslında manuel test ile geçirilen zamanla hemen hemen aynı sürede testlerimizi yazabiliriz.

**.NET Unit Test in VSCode (for Mac)**

**Adım 1:** VSCode da bir c# dosyası oluşturuyoruz.

A circuit board

Description automatically generated

**Adım 2:** VSCode’ a girip o dosyayı açıyoruz.

A screenshot of a cell phone screen with text

Description automatically generated

**Adım 3:** Basit bir kod yazıyoruz.

A screenshot of a computer screen

Description automatically generated

**Adım 4:** testDeneme.csproj a girip view ‘den command palette açıyoruz.

A screenshot of a cell phone

Description automatically generated

**5.Adım:** NuGet Package Manager:Add Package’e tıklıyoruz.

**A screenshot of a cell phone

Description automatically generated**

**6.Adım:** Böyle bir sayfa açılıyor.

A screenshot of a cell phone

Description automatically generated

**7.Adım:** O ekrana Microsoft.NET.Test.Sdk yazıp entera basıyoruz.

A screenshot of a cell phone

Description automatically generated

**8.Adım:** Böyle bir ekran çıkıyor en üsttekini seçiyoruz.

A screenshot of a computer

Description automatically generated

**9.Adım:** Bu ekrandan en üsttekini seçiyoruz.

**A screenshot of a computer screen

Description automatically generated**

**10.Adım:** Aynı işlemleri xunit yazıp yapıyoruz.

**A screenshot of a cell phone

Description automatically generated**

**11.Adım:** Aynı işlemleri xunit.runner.visualstudio yazıp yapıyoruz.

**A screenshot of a cell phone

Description automatically generated**

**12.Adım:** terminale dotnet build yazıyoruz.

A screenshot of a cell phone

Description automatically generated

**13.Adım:** New File a basıp yeni bir file oluşturuyoruz.

**A screenshot of a cell phone

Description automatically generated**

**14.Adım:** testclass.cs adında bir dosya oluşturduk.

**A screenshot of a cell phone

Description automatically generated**

**15.Adım:** İçerisine aşağidaki yapıyı yazıyoruz.

**A screenshot of a cell phone

Description automatically generated**

**16.Adım:** İçine test fonksiyonumuzu yazıyoruz. Başına [fact] koymak zorundayız.

**A screenshot of a cell phone

Description automatically generated**

**17.Adım:** Hata alıyoruz bunun için using unitTestDotNet ekliyoruz.

**A screenshot of a cell phone

Description automatically generated**

**18.Adım:** Programın başına tıklayıp using testdeneme; ekliyoruz.

**A screenshot of a cell phone

Description automatically generated**

**19.Adım:** build, run yapıp dotnet test yazıyoruz.

**A screenshot of a cell phone

Description automatically generated**