

C# Programlama

Birim Testi ve Performans Analizi

Emir Öztürk

Birim Testi

- Unit Test
 - Modül Testi
 - Fonksiyon - Prosedür Testi
 - Sınıf Testi
- Parçaların sistemden ayrı çalışmasının kontrolü
- Test Durumları

Birim Testinin Faydaları






- Hata tespiti
- Kod değişikliğinde eskiye uyumluluk
- Refactoring işleminde testlerin sağlanıp sağlanmadığının kontrolü
- Yeni özelliklerin eski sistem üzerinde testi
- Debug işlemi alanının daraltılması
- Zaman kazancı

Test Driven Development

- Testlerin Önce yazılması
- İhtiyaçların çıkartılıp ihtiyaç testlerinin hazırlanması
- Kodun ihtiyaçları karşılayacak şekilde yazılması
- Teorik?

Birim Testi

Test için Kisi sınıfı içeriği

```
public class Kisi {  
     5 usages  
    public string Ad { get; set; }  
     5 usages  
    public string Soyad { get; set; }  
     4 usages  
    public string Sifre { get; set; }  
     3 usages  
    public int Yas { get; set; }  
     11 usages  
    public Kisi(string ad, string soyad, string sifre, int yas) {  
        Ad = ad;  
        Soyad = soyad;  
        Sifre = sifre;  
        Yas = yas;  
    }  
  
    public Kisi(string ad, string soyad, string sifre)  
    {  
        Ad = ad;  
        Soyad = soyad;  
        Sifre = sifre;  
    }  
}
```

Birim Testi

Test için Login sınıfı içeriği

```
public class Login
{
    private static List<Kisi> kisiListesi;

    public Login()
    {
        kisiListesi = new List<Kisi>()
        {
            new Kisi(ad: "Emir", soyad: "Öztürk", sifre: "1234", yas: 42),
            new Kisi(ad: "Esat", soyad: "Tufan", sifre: "123", yas: 42),
            new Kisi(ad: "Ahmet", soyad: "Dursun", sifre: "456", yas: 16)
        };
    }

    public int CheckLogin(Kisi k)
    {
        var sonuc:Kisi? = kisiListesi.FirstOrDefault(x:Kisi => x.Ad == k.Ad && x.Soyad == k.Soyad);
        if (sonuc.Sifre == k.Sifre) return 1;
        else return 0;
    }
}
```

Birim Testi

Senaryo

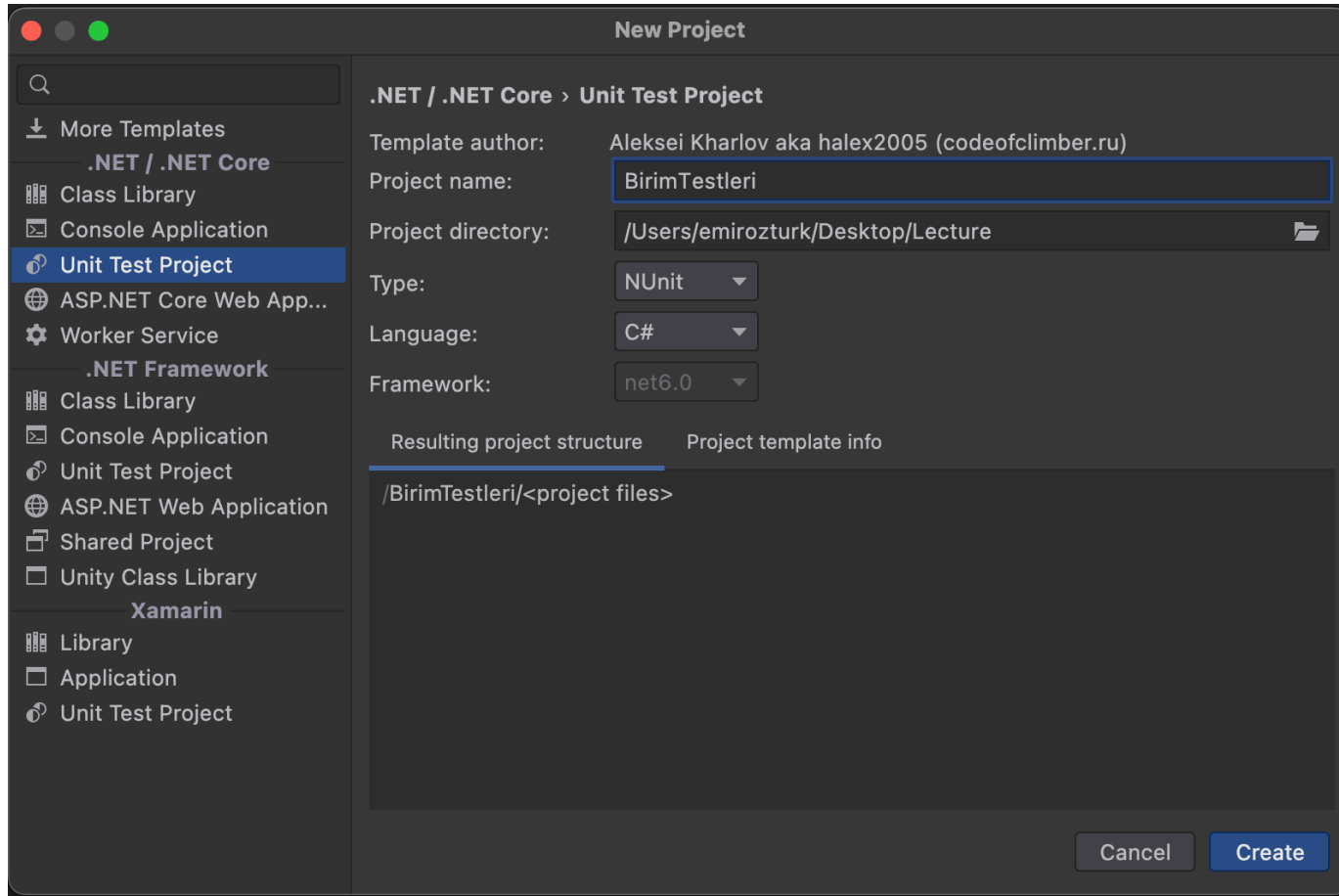
- Kullanıcı adı ve şifre girildiğinde kullanıcı adı ve şifre doğruysa "1" yanlıışsa "0" kodu döndüren fonksiyon

```
public int CheckLogin(Kisi k)
{
    var sonuc:Kisi? = kisiListesi.FirstOrDefault(x:Kisi => x.Ad == k.Ad && x.Soyad == k.Soyad);
    if (sonuc.Sifre == k.Sifre) return 1;
    else return 0;
}
```

- Daha sonra fonksiyon kullanıcının yaşı 18'den küçükse 0, 18'den büyükse kullanıcı adı ve şifre kontrolüne göre gerekli değeri döndürecek şekilde değiştirilecek

Birim Testi

Test Oluřturma



Birim Testi

İlk kontroller

```
using Lecture;

namespace BirimTestleri;

public class Tests
{
    private Login login;
    [SetUp]
    public void Setup()
    {
        login = new Login();
    }

    [Test]
    public void EmptyNameCheck()
    {
        Kisi k = new Kisi(ad: null, soyad: "Öztürk", sifre: "123");
        var sonuc: int = login.CheckLogin(k);
        if(sonuc == 0) Assert.Pass();
        else Assert.Fail(message: "Sonuç 0 dönmeliydi");
    }

    [Test]
    public void EmptySurnameCheck()
    {
        Kisi k = new Kisi(ad: "Emir", soyad: null, sifre: "123");
        var sonuc: int = login.CheckLogin(k);
        if(sonuc == 0) Assert.Pass();
        else Assert.Fail(message: "Sonuç 0 dönmeliydi");
    }
}
```

```
[Test]
public void EmptyPasswordCheck()
{
    Kisi k = new Kisi(ad: "Emir", soyad: "Öztürk", sifre: null);
    var sonuc: int = login.CheckLogin(k);
    if(sonuc == 0) Assert.Pass();
    else Assert.Fail(message: "Sonuç 0 dönmeliydi");
}

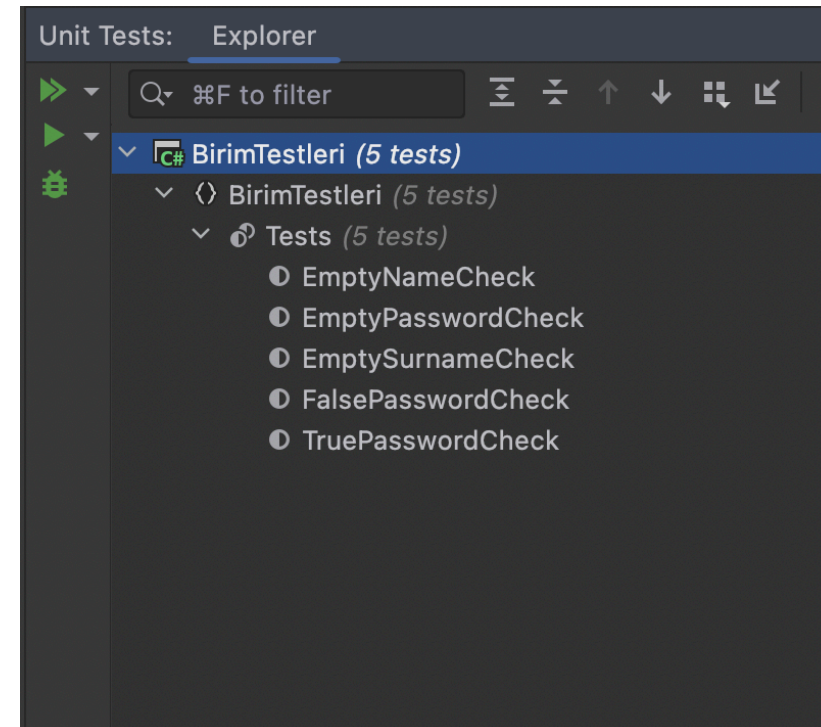
[Test]
public void FalsePasswordCheck()
{
    Kisi k = new Kisi(ad: "Emir", soyad: "Öztürk", sifre: "123");
    var sonuc: int = login.CheckLogin(k);
    if(sonuc == 0) Assert.Pass();
    else Assert.Fail(message: "Sonuç 0 dönmeliydi");
}

[Test]
public void TruePasswordCheck()
{
    Kisi k = new Kisi(ad: "Emir", soyad: "Öztürk", sifre: "1234" );
    var sonuc: int = login.CheckLogin(k);
    if(sonuc == 1) Assert.Pass();
    else Assert.Fail(message: "Sonuç 0 dönmeliydi");
}
```

Birim Testi









Test Ekranı

- Tüm testler çalıştırılabilir
- Tekrarlanabilir
- Kontrol edilebilir



Birim Testi

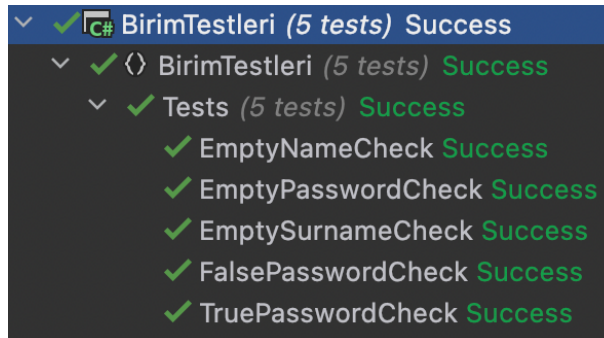
ilk test sonucu

- ✓  BirimTestleri (5 tests) Failed: 2 tests failed
 - ✓  {} BirimTestleri (5 tests) Failed: 2 tests failed
 - ✓  Tests (5 tests) Failed: One or more child tests had errors: 2 tests failed
 -  EmptyNameCheck Failed: System.NullReferenceException : Object reference not set to an instance of an object.
 -  EmptyPasswordCheck Success
 -  EmptySurnameCheck Failed: System.NullReferenceException : Object reference not set to an instance of an object.
 -  FalsePasswordCheck Success
 -  TruePasswordCheck Success

Birim Testi

Testleri sağlamak için kodun değiştirilmiş hali

```
public int CheckLogin(Kisi k)
{
    if (k.Ad != null && k.Soyad != null)
    {
        var sonuc:Kisi? = kisiListesi.FirstOrDefault(x:Kisi => x.Ad == k.Ad && x.Soyad == k.Soyad);
        if (sonuc.Sifre == k.Sifre) return 1;
        return 0;
    }
    return 0;
}
```



Birim Testi

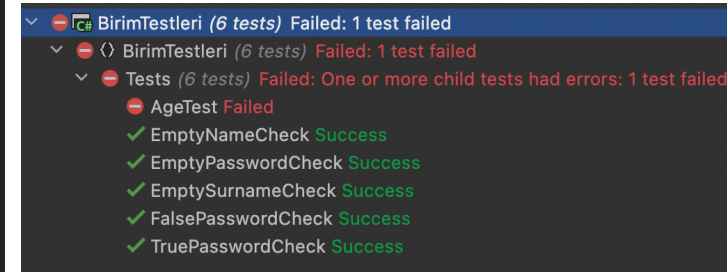
Yaş eklenmesi

```
public int CheckLogin(Kisi k)
{
    if (k.Ad != null && k.Soyad != null)
    {
        var sonuc:Kisi? = kisiListesi.FirstOrDefault(x:Kisi => x.Ad == k.Ad && x.Soyad == k.Soyad);
        if (sonuc.Yas > 18)
        {
            if (sonuc.Sifre == k.Sifre) return 1;
            return 0;
        }
        else if(sonuc.Yas < 18)
            return 0;
    }
    return 0;
}
```

Birim Testi

Yaş testleri

```
[Test]
public void AgeTest()
{
    Kisi k1 = new Kisi(ad: "Emir", soyad: "Öztürk", sifre: "1234");
    Kisi k2 = new Kisi(ad: "Esat", soyad: "Tufan", sifre: "123");
    Kisi k3 = new Kisi(ad: "Ahmet", soyad: "Dursun", sifre: "456");
    var sonuc1:int = login.CheckLogin(k1);
    var sonuc2:int = login.CheckLogin(k2);
    var sonuc3:int = login.CheckLogin(k3);
    if(sonuc1 == 1 && sonuc2 == 0 && sonuc3 == 1)Assert.Pass();
    else Assert.Fail();
}
```



Birim Testi

Düzeltilme

```
public int CheckLogin(Kisi k)
{
    if (k.Ad != null && k.Soyad != null)
    {
        var sonuc:Kisi? = kisiListesi.FirstOrDefault(x:Kisi => x.Ad == k.Ad && x.Soyad == k.Soyad);
        if (sonuc.Yas >= 18)
        {
            if (sonuc.Sifre == k.Sifre) return 1;
            return 0;
        }
        return 0;
    }
    return 0;
}
```

```
✓ C# BirimTestleri (6 tests) Success
  ✓ {} BirimTestleri (6 tests) Success
    ✓ Tests (6 tests) Success
      ✓ AgeTest Success
      ✓ EmptyNameCheck Success
      ✓ EmptyPasswordCheck Success
      ✓ EmptySurnameCheck Success
      ✓ FalsePasswordCheck Success
      ✓ TruePasswordCheck Success
```

Birim Testi

Refactoring

- Kodun mantığını değiştirmeden içeriğini değiştirmek
 - Performans arttırmak veya okunurluk arttırmak
- Testlerle fonksiyonalitenin bozulup bozulmadığı tespit edilebilir

Birim Testi

Refactoring

```
public int CheckLogin(Kisi k)
{
    if (k.Ad != null && k.Soyad != null)
    {
        var sonuc:Kisi? = kisiListesi.
            FirstOrDefault(x:Kisi => x.Ad == k.Ad
                           && x.Soyad == k.Soyad);

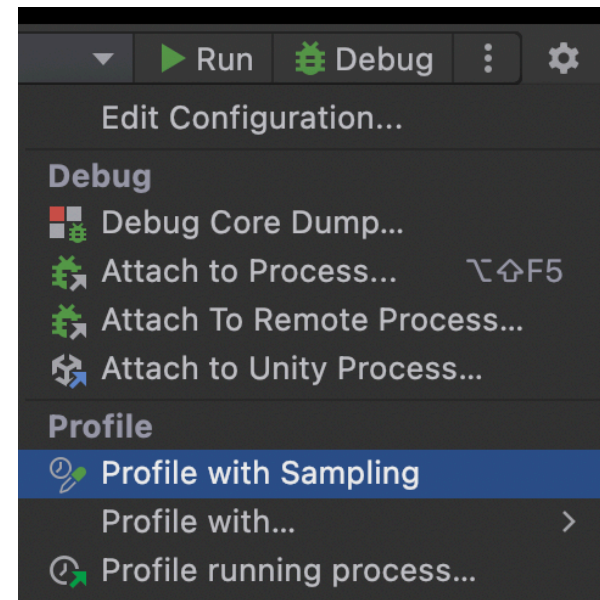
        if (sonuc.Yas >= 18)
        {
            if (sonuc.Sifre == k.Sifre) return 1;
            return 0;
        }
        return 0;
    }
    return 0;
}
```

```
public int CheckLogin(Kisi k)
{
    return Convert.ToInt32(
        kisiListesi.Count(x:Kisi => x.Ad == k.Ad
                                   && x.Soyad == k.Soyad
                                   && x.Yas >= 18
                                   && x.Sifre == k.Sifre)>0);
}
```

```
✓ [C#] BirimTestleri (6 tests) Success
  ✓ {} BirimTestleri (6 tests) Success
    ✓ Tests (6 tests) Success
      ✓ AgeTest Success
      ✓ EmptyNameCheck Success
      ✓ EmptyPasswordCheck Success
      ✓ EmptySurnameCheck Success
      ✓ FalsePasswordCheck Success
      ✓ TruePasswordCheck Success
```

Performans Analizi

- Kod performansı
- Hangi kısmın ne kadar kaynak tükettiğinin tespiti



Performans Analizi

Örnek Kod

```
namespace Lecture;

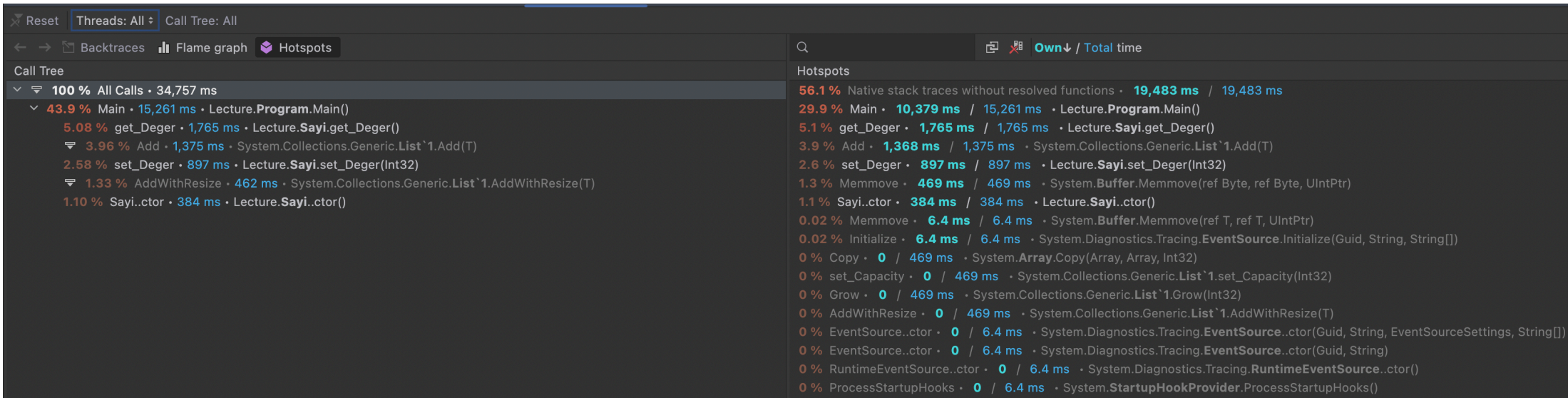
[2 usages]
class Sayi
{
    [4 usages]
    public int Deger { get; set; }
}

class Program {
    public static void Main()
    {
        List<int> liste = new List<int>();
        for (int i = 0; i < 1_000_000_000; i++)
        {
            Sayi s = new Sayi();
            s.Deger = i;
            s.Deger = s.Deger * 2 % 10;
            liste.Add(s.Deger);
        }
    }
}
```

Time ▲	Snapshot	Duration	Size	
20:53:40	Lecture		14s	2.1 MB

Performans Analizi

Örnek Kod



Performans Analizi

İyileştirme

```
1 usage More...
class Sayi
{
    1 usage
    public static int Hesapla(int deger) => deger * 2 % 10;
}

class Program {
    public static void Main()
    {
        List<int> liste = new List<int>();
        for (int i = 0; i < 1_000_000_000; i++)
            liste.Add(item: Sayi.Hesapla(i));
    }
}
```

Time ▲	Snapshot	Duration	Size
20:57:09	Lecture	3s	2.1 MB

Call Tree

```
100 % All Calls • 12,687 ms
  33.3 % Main • 4,227 ms • Lecture.Program.Main()
    11.5 % Add • 1,461 ms • System.Collections.Generic.List`1.Add(T)
    3.24 % AddWithResize • 411 ms • System.Collections.Generic.List`1.AddWithResize(T)
    2.22 % Hesapla • 282 ms • Lecture.Sayi.Hesapla(Int32)
```

Hotspots

```
66.7 % Native stack traces without resolved functions • 8,460 ms / 8,460 ms
16.3 % Main • 2,073 ms / 4,227 ms • Lecture.Program.Main()
11.3 % Add • 1,435 ms / 1,461 ms • System.Collections.Generic.List`1.Add(T)
3.2 % Memmove • 411 ms / 411 ms • System.Buffer.Memmove(ref Byte, ref Byte, UIntPtr)
2.2 % Hesapla • 282 ms / 282 ms • Lecture.Sayi.Hesapla(Int32)
0.2 % set_Capacity • 26 ms / 436 ms • System.Collections.Generic.List`1.set_Capacity(Int32)
0 % Copy • 0 / 411 ms • System.Array.Copy(Array, Array, Int32)
0 % Grow • 0 / 436 ms • System.Collections.Generic.List`1.Grow(Int32)
0 % AddWithResize • 0 / 436 ms • System.Collections.Generic.List`1.AddWithResize(T)
```