Nesne Tabanlı Programlama
Bölüm 2
Sınıflar ve Nesneler

Dr. Öğr. Üyesi Murat TAŞYÜREK (kayubmprogramlama1@gmail.com)

9 Ekim 2023

Kayseri Üniversitesi, Bilgisayar Mühendisliği Bölümü

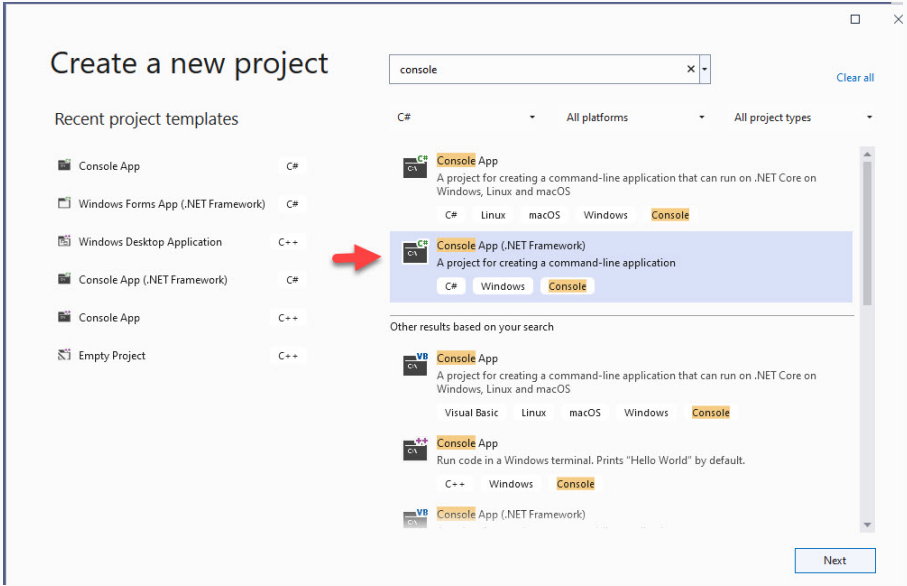
Sınıflar ve Nesneler

- C# programlama dilinde nesneleri oluşturmak için sınıf (class) adı verilen veri türü kullanılmaktadır.
- Bu veri türü nesneyi oluşturacak olan verileri ve fonksiyonları birlikte barındırmaktadır.
- Sınıflar nesnelerin modelidir.
- Nesne tabanlı programlama mantığından bir defa sınıf yazılır ve ihtiyaç olduğunda tekrar tekrar kullanılır.

Örnek

- Bir kişi (insan) türünde bir nesne oluşturmak istiyoruz.
- Bunun için öncelikle bir sınıf oluşturunuz.
- İnsan çocuk olarak dünya geldiğinde önceklikle adı ve soyadı belirlenir.
- Adi ve soyadi adında iki özelliği olmalı.
- Gerçek hayattaki adının konunmasına benzer şekilde metot (olay, etkinlik) tanımlanmalı ?
- Senin adın nedir ? Adını söyleyebilmeli (bilgisayar dilinde ekrana yazdırmalı)
- Senin adın var mı ? Adın konuldumu konulacak mı ?

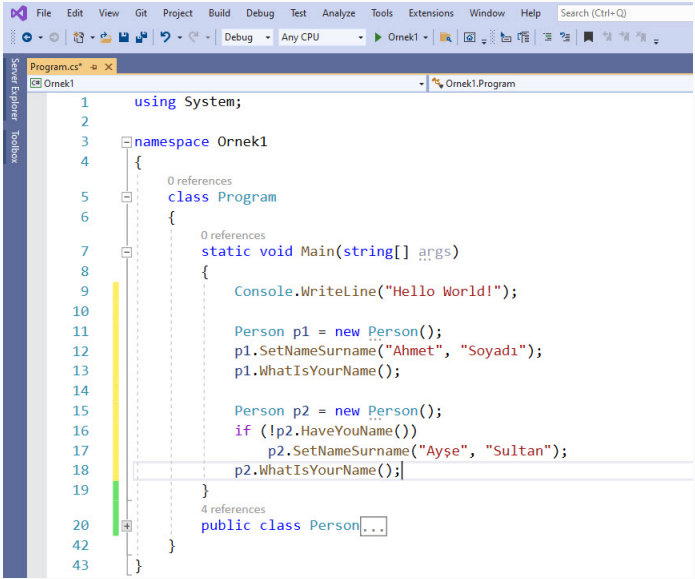
Visual Studio ile C# dilinde Console App Oluşturma



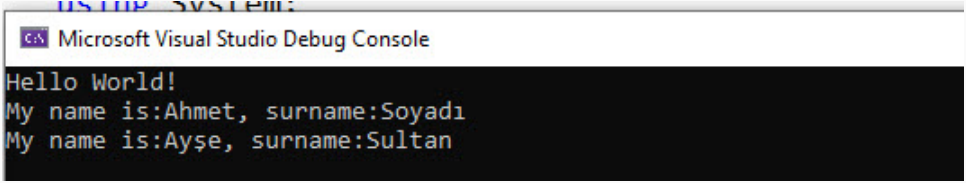
Person Class

```
4 references
public class Person
{
    string name, surname;
    2 references
    public void SetNameSurname(string name, string surname)
    {
        this.name = name;
        this.surname = surname;
    }
    2 references
    public void WhatIsYourName()
    {
        Console.WriteLine("My name is:" + this.name + ", surname:" + this.surname);
    }
    1 reference
    public bool HaveYouName()
    {
        bool found = false;
        if (string.IsNullOrEmpty(this.name))
            found = false;
        else
            found = true;
        return found;
    }
}
```

Source Code


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Debug
Any CPU
Ornek1
Program.cs*
Ornek1
Ornek1.Program
Server Explorer
Toolbox
using System;
namespace Ornek1
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
Console.WriteLine("Hello World!");
Person p1 = new Person();
p1.SetNameSurname("Ahmet", "Soyadı");
p1.WhatIsYourName();
Person p2 = new Person();
if (!p2.HaveYouName())
p2.SetNameSurname("Ayşe", "Sultan");
p2.WhatIsYourName();
}
4 references
public class Person...
}
}

Çıktı


Microsoft Visual Studio Debug Console
Hello World!
My name is:Ahmet, surname:Soyadı
My name is:Ayşe, surname:Sultan

Değerlendirme

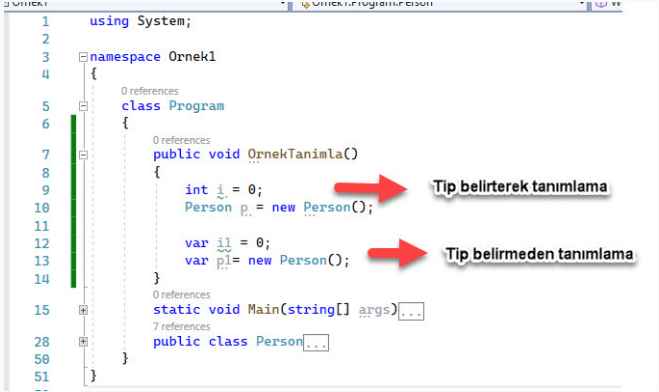
- Programda p1 ve p2 olmak üzere iki adet nesne tanımlanmıştır.
- Bu nesnelerin üye fonksiyonları çağrılarak nesnelerin belirli davranışlarda bulunması sağlanmıştır.
- Nesnelerin metotlarının çağrılması (canlandırılması) o nesneye mesaj göndermek olarak da ifade edilmektedir.
- Sınıflar aktif veri türüdür. Sınıflardan nesneler üretilir.
- Nesnelere mesaj gönderildiğinde nesneler bir davranışta bulunarak cevap verir.

Encapsulation

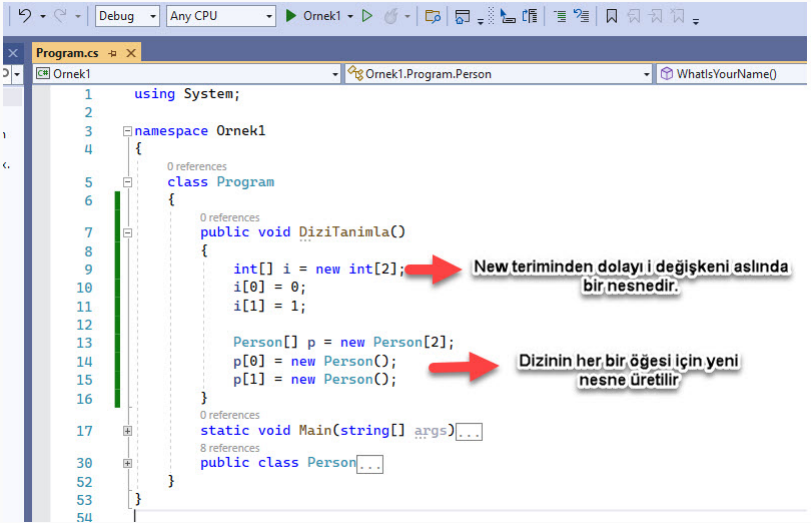
- Sınıflar hem verileri hemde bu veriler ile işlem yapan fonskiyonları aynı paket içinde barındırır.
- Bu barındırma işlemi encapsulation (kapsülleme) olarak tanımlanır.
- Kapsülleme, C# bağlamında, bir nesnenin, kullanıcısı için gerekli olmayan verileri ve davranışları gizleme yeteneğini ifade eder. Kapsülleme, bir grup özellik, yöntem ve diğer üyelerin tek bir birim veya nesne olarak kabul edilmesini sağlar.

Tür belirtmeden nesne tanımlama

- Derleyiciler int, float, string gibi veri türleri için nasıl kullanılıyorsa nesne tanımlamak için aynı şekilde kullanılır.
- C# diline nesne oluşturmak içi new terimi kullanılır.
-



Nesnelerin Dizi Olarak Tanımlanması



Sınıf üyelerine erişimin denetlenmesi

- Yazılım geliştiriciler (programcılar) sınıfların oluşturulması ve kullanılması açısından ikiye ayrılır.
- İlk grup bu sınıfları yazar. İkinci grup ise bu sınıfları kullanarak nesne oluşturur. İkinci grup sınıfların orjinal yapısına asla müdahale edemez. Sadece sınıf tanımlayıcısının yani birinci grubun izin verdiği ölçüde nesneleri kullanabilir.
- Günümüz şartlarındaki backend, frontend terimine benzetilebilir. Backend arkaplanda servis yazar veritabanına bağları verileri çeker. Veritabanı vb. bütün işlemeri backend yapar.
- Frontend tarafında ise servis kullanılır. Yani aslında servis (class) türünden bir nesne üretilir ve o nesnenin ilgili evantı çağrılır (mesaj) gönderilir. Gelen bilgi frontend tarafından tasarlanan kullanıcı dostu arayüz ile kullanıcılara sunulur.
- Sınıfların kullanımı da benzer şekildedir. İlk grup oluşturur. İkinci grup kullanır.

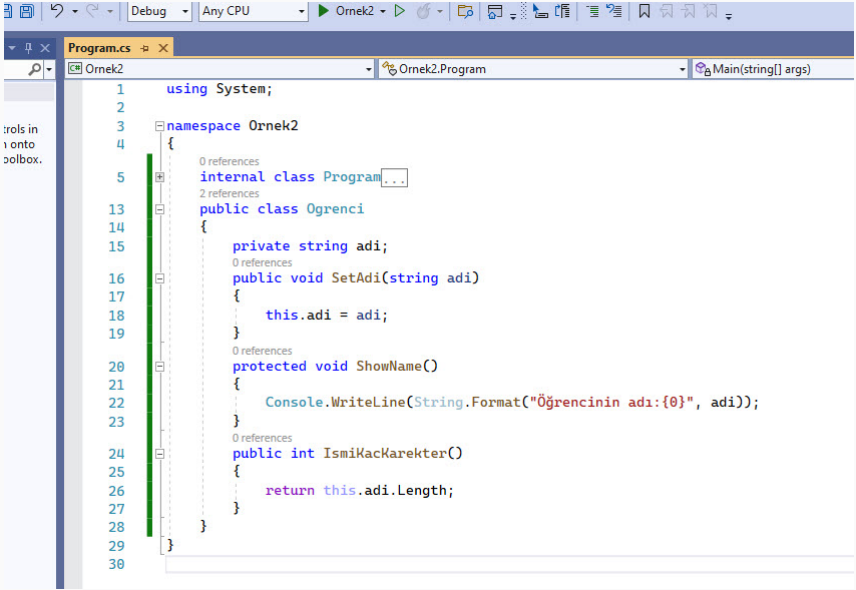
Public, Protected, private

- Sınıfların özelliklerin sağlamak için sınıfı yazan yazılımcı sınıfın bazı üyelerini (özellik, fonskiyon,procedure vb.) gizleyerek (data hiding) onlara sınıf dışından erişilmesini engelleyebilir.
- Bir sınıfın içindeki üyelere erişimi denetleyen üç farklı etiket bulunmaktadır. Bu özellikler, **public** (açık), **protected** (korumalı), **private** (özel).
- **Public** olarak tanımlanan üyerele bu sınıftan türetilen bütün nesneler ve sınıflar erişebilir.
- **Protected** olarak tanımlanan üyelere sadece bu sınıftan kalıtım (inheritence) yoluyla üretilen sınıflar erişebilir. Inheritence ilerleyen haftalarda full detayıcı ile anlatılacak.
- **Private** olarak tanımlanan üylere sadece bu sınıfın içerisinden erişebilir. Bu sınıftan türetilen nesneler ve sınıflar erişemez.

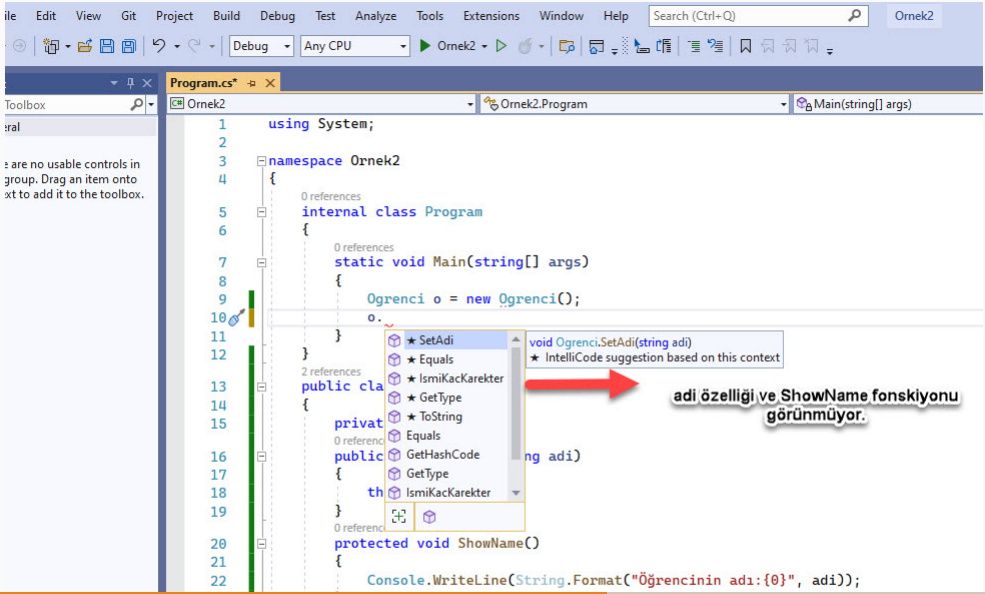
Örnek

- Bir öğrenci olduğunu ve öğrencileririn sadece isim bilgisinin tutulduğunu varsaylım.
- Öğrencinin adı dışardan set edilebilsin ama dışardan hiç bir şekilde okunamasın. Sadece kalıtım yoluyla öğrencinin adının ne olduğu öğrenilebilsin.
- Örğencinin adının kaç karekter olduğunu öğrenilebilsin.
- Öğrenci türünde bir sınıfımızın olması gerektiğini anlıyoruz.
- adı isminde bir özellik olmalı ve erişim olmaması için **private** olması gerektiği anlıyoruz. İsmini set etmek için **public** bir procedüre, isimini göstermek için **protected** bir procedure ve isminin kaç karekter olduğunu göstermek için de **public** bir fonksiyon yazmamız gerektiğini anlıyoruz.

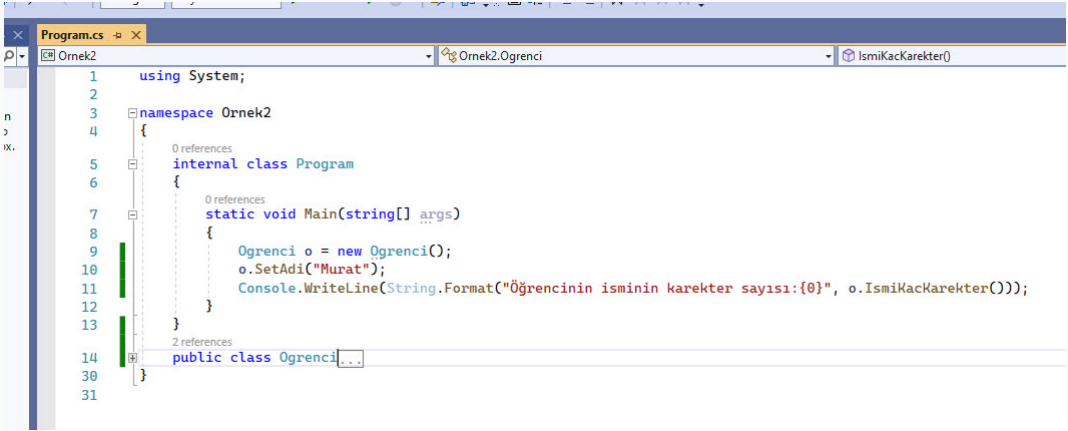
Öğrenci classı



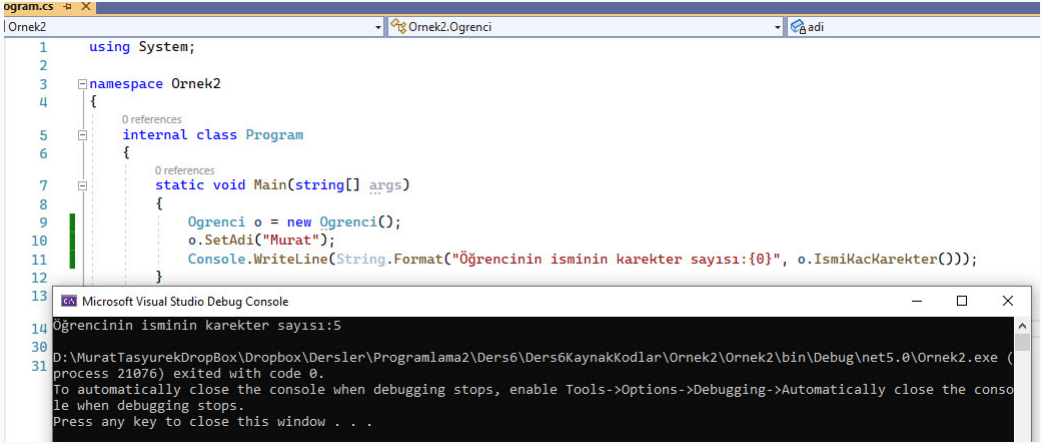
Öğrenci classının üyelerine erişim



Kaynak Kod


Program.cs
Ornek2
Ornek2.Ogrenci
IsmiKacKarekter()
using System;
namespace Ornek2
{
0 references
internal class Program
{
0 references
static void Main(string[] args)
{
Ogrenci o = new Ogrenci();
o.SetAdi("Murat");
Console.WriteLine(String.Format("Öğrencinin isminin karekter sayısı:{0}", o.IsmiKacKarekter()));
}
}
2 references
public class Ogrenci...
}

Çıktı


ogram.cs
Ornek2
Ornek2.Ogrenci
adi
using System;
namespace Ornek2
{
0 references
internal class Program
{
0 references
static void Main(string[] args)
{
Ogrenci o = new Ogrenci();
o.SetAdi("Murat");
Console.WriteLine(String.Format("Öğrencinin isminin karekter sayısı:{0}", o.IsmiKacKarekter()));
}
Microsoft Visual Studio Debug Console
Öğrencinin isminin karekter sayısı:5
D:\MuratTasyurekDropBox\Dropbox\Dersler\Programlama2\Ders6\Ders6KaynakKodlar\Ornek2\Ornek2\bin\Debug\net5.0\Ornek2.exe (
process 21076) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .

Sınıf ile ilgili bilgiler

- Bir sınıfın üyelerinin dışarıya erişim tipi normalde **private** (özel) moddadır.
- Sınıf oluşturan kişi sınıfın üyelerini dışarıya açmak istiyorsa **public** veya protected ile **yetki** çerçevesinde açar.
- Benzer şekilde sınıf tanımlarken **public** ibaresi kullanılanılmadan **class Ogrenci** formatında sınıf tanımlarsanız bu sınıfta **private** modda olur. Böyle olduğu durumda kullanıldığı blok dışında diğer sınıf veya kod blokları tarafından erişilemez.
- Sınıfın içerisine başka bir sınıf değişken olarak tanımlanabilir. Burada aynı özellikler geçerlidir. Sonuçta sınıfta bir tür değişkendir.

Örnek

- Önceki öğrenci bilgilerine ilaveten bir öğrencinin iki tane velisinin olduğunu varsayalım.
- Veli bilgileri okunabilir ancak daha sonra değiştirilemez olduğunu farzedelim.
- Veli adının sadece kendi içerisinde değiştirebildiğini düşünelim.
- Bir öğrencinin velisinin bilgisinin dışardan erişilebilir olduğunu varsayalım.

Veli Sınıfı

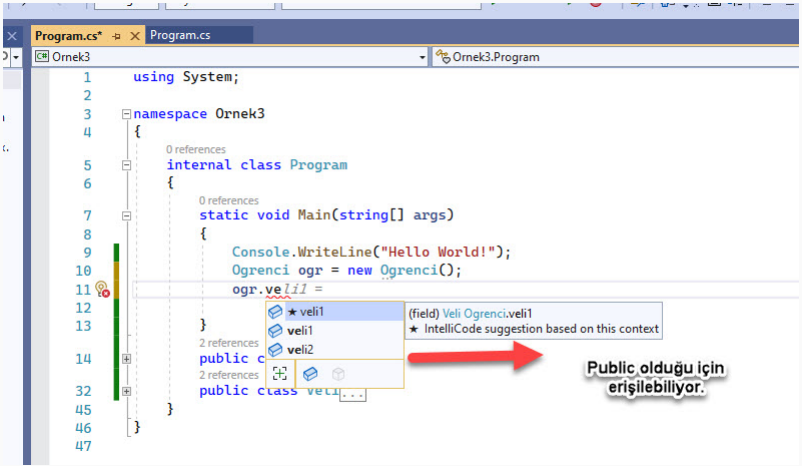

2 references
public class Veli
{
string veliAdi="Ahmet";
0 references
public string GetVeliAdi()
{
return this.veliAdi;
}
0 references
private void SetVeliAdi(string adi)
{
this.veliAdi = adi;
}
}
Public terimi kullanıdlından dışardan erişebilir
Terim kullanılmadığından private'dır.

Öğrenci Sınıfı

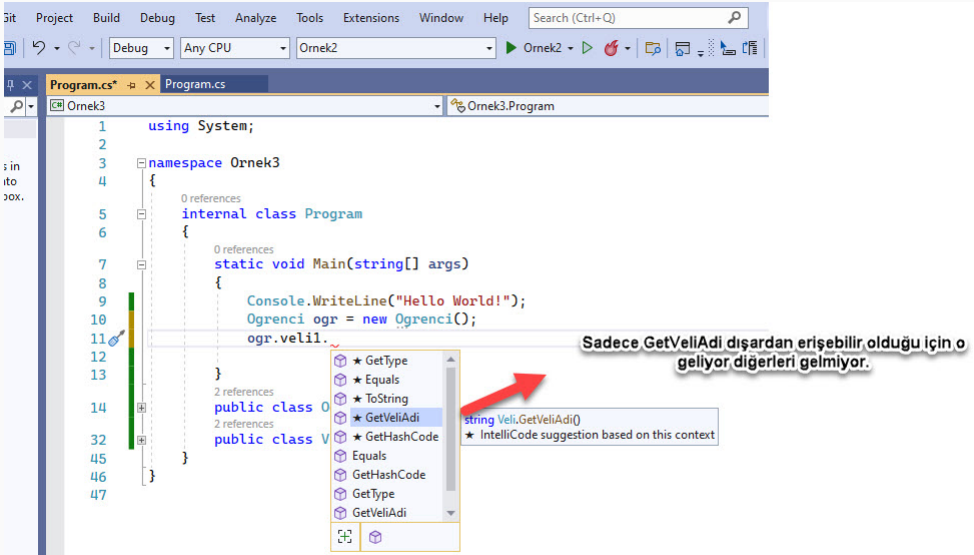
```
2 references
public class Ogrenci
{
    private string adi;
    public Veli veli1= new Veli();
    public Veli veli2= new Veli();
    0 references
    public void SetAdi(string adi)
    {
        this.adi = adi;
    }
    0 references
    protected void ShowName()
    {
        Console.WriteLine(String.Format("Öğrencinin adı:{0}", adi));
    }
    0 references
    public int IsmiKacKarekter()
    {
        return this.adi.Length;
    }
}
```

Veli türünde iki adet öznitelik. Public olduğundan dışardan erişilebilir.

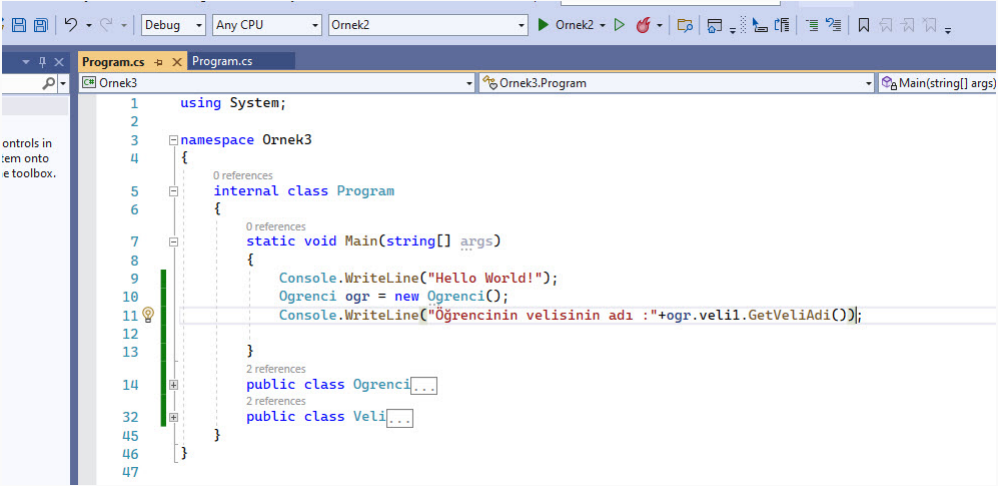
Öğrencinin velisine erişim



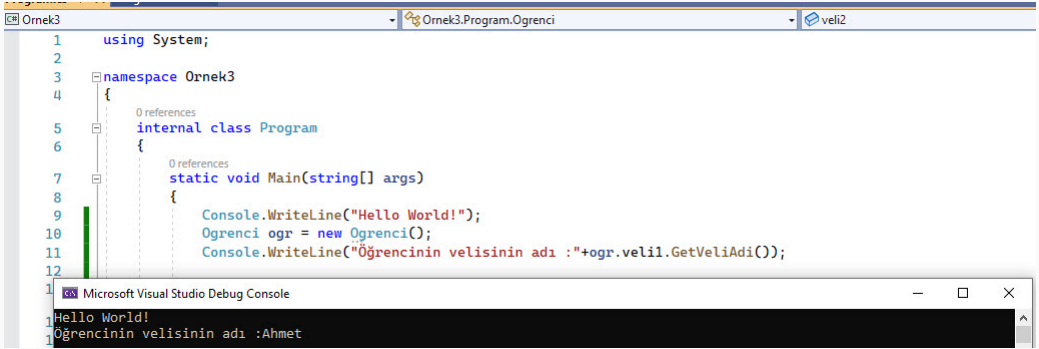
Öğrencinin velisinin adına erişim



Kaynak Kod


Debug
Any CPU
Ornek2
Ornek2
Program.cs
Program.cs
Ornek3
Ornek3.Program
Main(string[] args)
using System;
namespace Ornek3
{
0 references
internal class Program
{
0 references
static void Main(string[] args)
{
Console.WriteLine("Hello World!");
Ogrenci ogr = new Ogrenci();
Console.WriteLine("Öğrencinin velisinin adı :"+ogr.veli1.GetVeliAdi());
}
2 references
public class Ogrenci...
2 references
public class Veli...
}
}

Çıktı


Ornek3
Ornek3.Program.Ogrenci
veli2
using System;
namespace Ornek3
{
0 references
internal class Program
{
0 references
static void Main(string[] args)
{
Console.WriteLine("Hello World!");
Ogrenci ogr = new Ogrenci();
Console.WriteLine("Öğrencinin velisinin adı :"+ogr.veli1.GetVeliAdi());
Microsoft Visual Studio Debug Console
Hello World!
Öğrencinin velisinin adı :Ahmet

Ödev

- Ödev: kişi ve araç bilgilerinin tutulduğu bir sistem isteniyor. Bir kişinin birden çok arabası olabilir.
- Araç ise taksi, kamyon, otobüs vb. bir çok araç olabilir. Ayrıca, araç ile ilgili modeli, yılı, markası, edinme tarihi ve edinme fiyatı tutulmalıdır.
- Kişi bilgisi olarak kişinin adı, soyadı, doğum yılı, TC Kimlik numarası tutulmak isteniyor. Ancak bu bilgiler gizli olmalı.
- Dört adet farklı türden aracınızın olduğunu ilk aracın 4., ikinci aracın 5., üçünkü aracın 6. ve dördüncü aracın 3. sahibi yani sizden önceki 2 sabibi vardı.
- Kişi arabalarımı listele dediğinde aracın şasi numarası: ***A****9 şeklinde, Murat Taşyürek, Edinme Tarihi:01.01.2020, Model Yılı: Bir önceki sahibinin adı:... şeklinde göstermeli
- Bütün veriler dinamik olarak girilmeli, ödevler özgün olmalı, bütün işlemler class içerisinde olmalı
- Aynı ödevler **sıfır** puan olarak değerlendirilecektir.
- 16 Ekim 06.59'a kadar gönderilenler **8+2** puan, bu tarihten sonra vize sınavına kadar gönderenler **8** puan (kayubmprogramlama1@gmail.com).