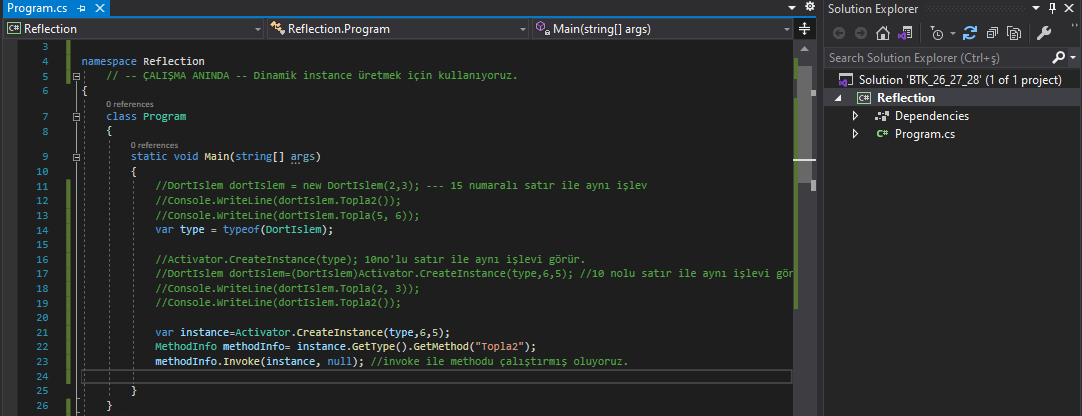
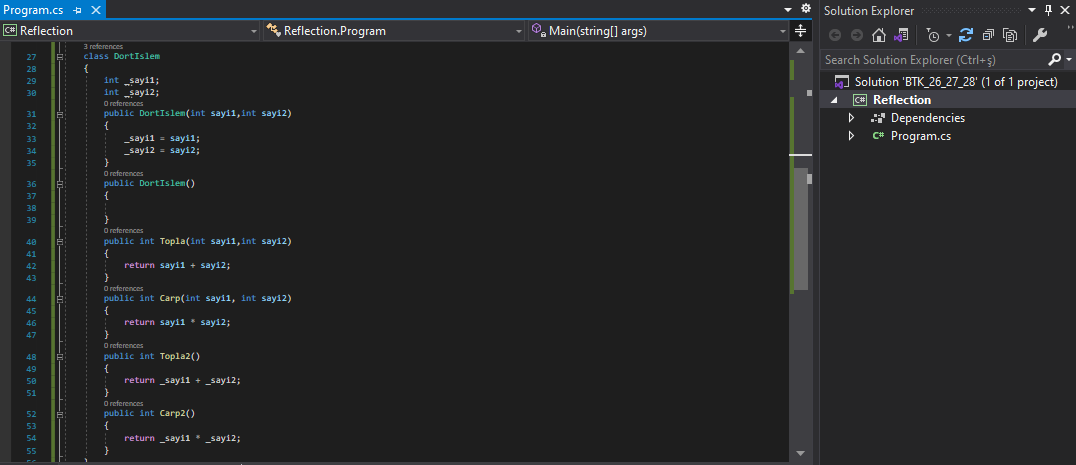
1-REFLECTION

Main bloğu

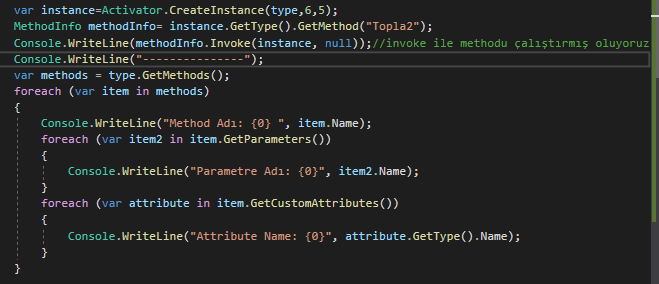
#Method Info ve Invoke



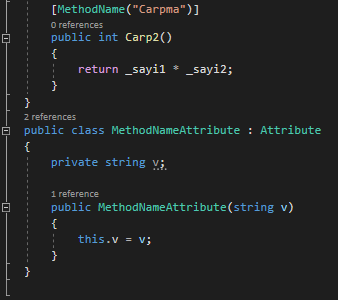
DortIslem Class bloğu 

* 1. Reflection ile detaylar

Bir methods tanımladık ve buna type.GetMethods(); formatını biçtik. Bunun ardından kod bloğunu foreach döngüsüne sokarak projede bulunan tüm methodları bu methodların isimlerini, methodların parametre isimlerini ve -varsa- attributeleri listelemiş olduk. Bu da Reflection sayesinde oldu.

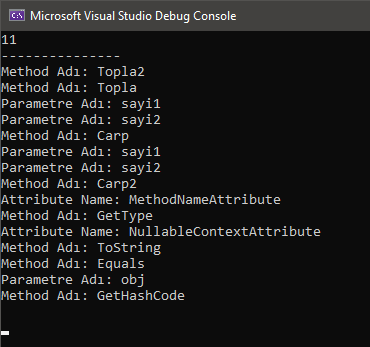


Attribute listesini daha iyi görebilmek için kafadan bir Attribute yazdık ve Carp2’e bu attribute’i tanımlamış olduk. Böylece burada yazdığımız attribute’i console üzerinde görebilmiş oluyoruz.



En son görüntü bu oluyor;

Ve çıktıda görüldüğü gibi tüm methodlar, methodların parametreleri ve -varsa- attributeleri gözükmüş oluyor.



1. – Delegates

Burada toplamda 3 adet delege oluşturduk. Her delege tanımlandığı türe göre methodlara delegelik yapar. MyDelegate void döndüren ve parametresiz methodlara delegelik eder.

MyDelegate2 void döndüren ve string olarak parametreli methodlara delegelik yapar.

MyDelegate3 int döndüren ve yine int parametreli methodlara delegelik yapar.

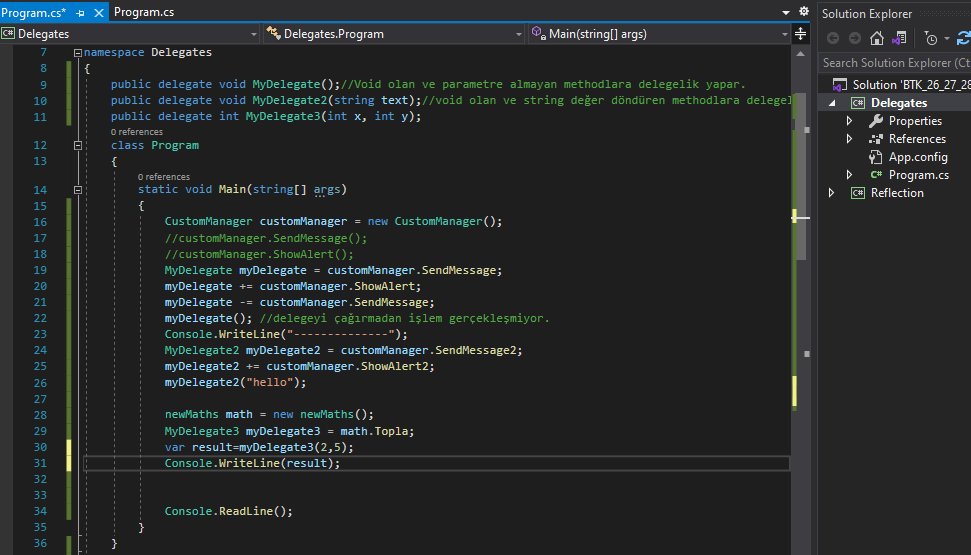
İşin özü tanımlamada yatmaktadır. Basit şekilde anlatacak olursam gerçek hayattan örnek olaraktan gerçek bir elçi görevi görmektedir. Classlar tanımlandıktan sonra;

MyDelegate myDelegate = customManager.SendMessage;

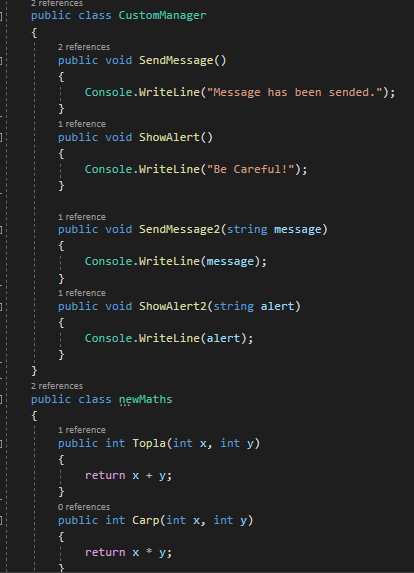
myDelegate += customManager.ShowAlert;

myDelegate();

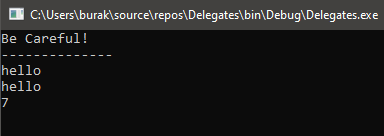
kullanım olarak bu bu şekildedir. CustomManager içerisinde SendMessage ve ShowAlert methodlarını göndermiş oluyoruz. Bunu da tabi ki elçiden gönderek yaptırıyoruz. Ana kullanım kod bloğu ise bu şekilde olmaktadır.



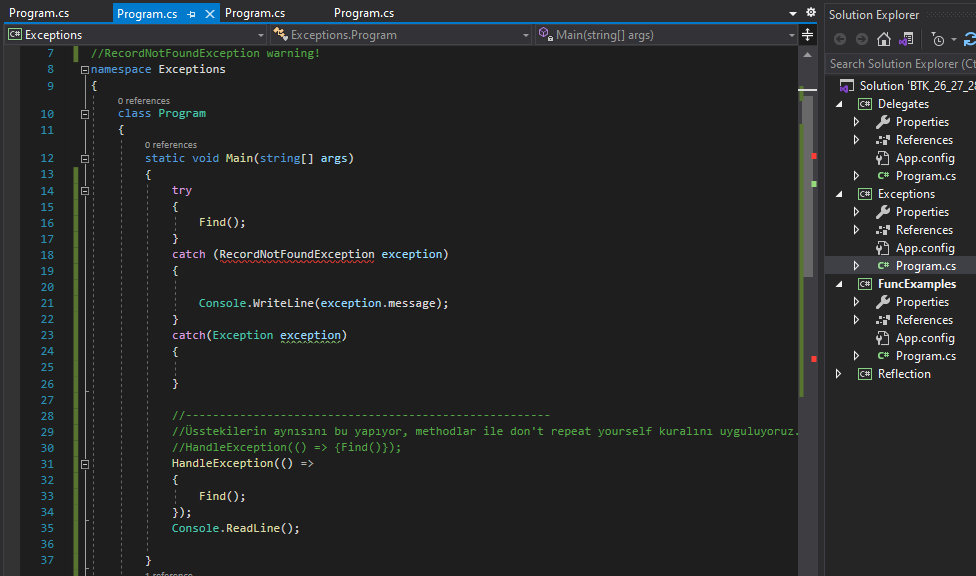
Class blokları



Kod çıktısı;



* 1. Delegates-Action

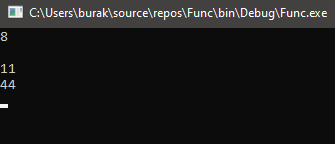
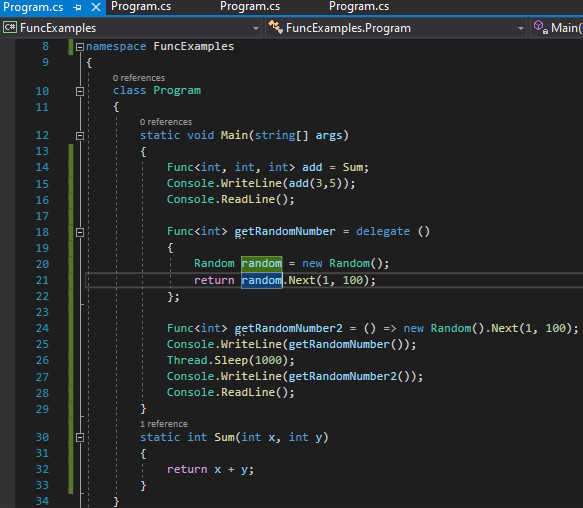
Action delege türü hiçbir parametre almayan ve değer döndürmeyen metodları temsil eder. Adından da anlaşılacağı üzere bir fonksiyonu değil bir eylemi(action) temsil etmektedirler. Action<T> delegeleri birden fazla parametre alabilecek metodları da temsil edebilecek şekilde tasarlanmıştır.



1. – FUNC

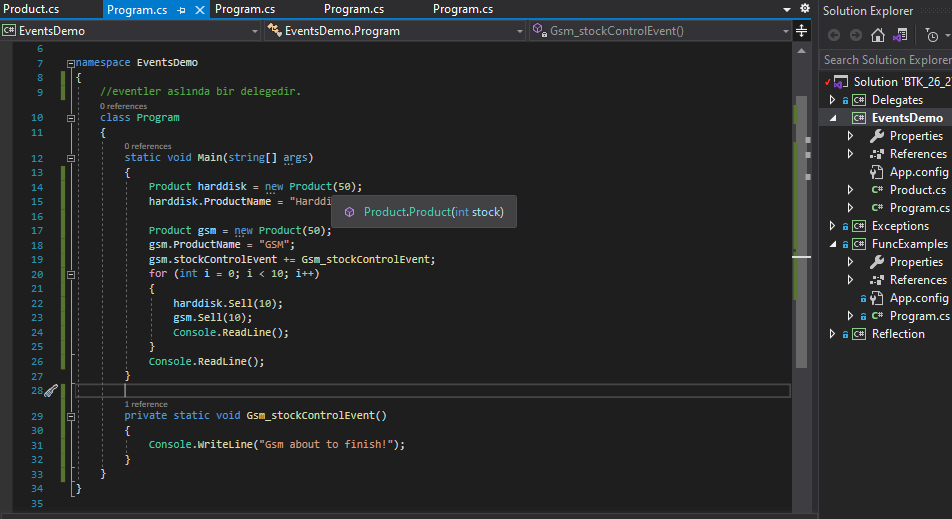
Func ise Action’dan farklı olarak parametresiz methodları değil parametreleri döndürür.

Func tipi, bir veya birden fazla(16 haneye kadar) parametre alabilen ve geriye T tipinde değer döndüren bir delege yapısıdır.Bu delege sayesinde, alacağı parametreler ve geriye döndüreceği değeri simgeleyen delegemizi tanımlayabilir ve o tanımladığımız yapıdaki herhangi bir metodu bu delegeye işaretletebiliriz



4.EVENTS

C# içerisinde yer alan event anahtar kelimesi delegate anahtar kelimesi ile tanımlanan bir değişkenin olay olduğunu açıkca belirtmek ve yönetmek için kullanılan anahtar kelimedir. Buradaki örnekte iki ürünümüz var ve birini elimizden çıkartmayı istiyoruz diğerinin ise hiç bitmemesini ve stok adedi 15’e düştüğünde uyarı vermesini istiyoruz. Bunun içinde event kullanıyoruz ve bununla sağlam bir şekilde ilerliyoruz.



Burada public int Stock get set yerinde normal get set’den farklı bir şekilde veririz. Bunun nedeni burada stok değeri olarak üzerinde değişiklik yapılacağından kaynaklanmaktadır. Ve buna göre de niteliklerini ve uygulanması gereken özelliklerini ona göre vererek yolumuza devam ederiz. Burada get kısmında direkt olarak const’dan \_stock’u getiririz yani dışarıdan verilecek değeri. Set kısmında ise \_stock ile girilen değer olarak value’i eşitleriz. Bunun ardından da bir şart bloğuna koymamız gerektiğinden if-else yapısı oluşturarak burada koşullarımızı belirleriz. Stok adedimiz 15’den az olduğunda ve kişi de evente abone ise burada eventimizi çağırıyoruz. Aşağısında da zaten standart Sell methodu bulunmakta.

