Delegates

Delegeler metodların adreslerini tutarlar. Bir delegate imzası aşağıdaki gibi tanımlanmaktadır.

Example

public delegate void TestHandler(string str);

Example

public delegate void TestHandler(string str);

class Program{

static void Main(){

Program obj = new Program();

TestHandler delegate1 = new TestHandler(obj.Message);

obj.Message("Metod kullanıldı.");

*//delegate1.Invoke() komutuyla da çalıştırabilirsiniz.*

delegate1("Delegate kullanıldı.");

Console.ReadLine();

}

void Message(string message){

Console.WriteLine(message);

}

}

* delegate’in imzası temsil edeceği metoda uygun olmalıdır.
* Delegeler erişim belirteçlerine sahip olabilirler.
* Delege isimleri genellikle **Handler** ile bitmektedir.
* Bir delegate birden fazla metodun adresini tutabilir. Metod ekleme ve çıkarma işlemleri += ve -= operatörleriyle yapılmaktadır.
* Bir delegate birden fazla metodun adresini tutuyorsa delegate çalıştırıldığında bağlanan son metod çalışır.

Example

public delegate int TestHandler(int x, int y);

class Program{

static void Main(){

Program obj = new Program();

TestHandler delegate1 = new TestHandler(obj.AddNumbers);

delegate1 += new TestHandler(obj.MultiplyNumbers);

delegate1 += new TestHandler(obj.DivideNumbers);

Console.Write(delegate1(10, 5));

Console.ReadLine();

}

int AddNumbers(int x, int y){

return x + y;

}

int MultiplyNumbers(int x, int y){

return x \* y;

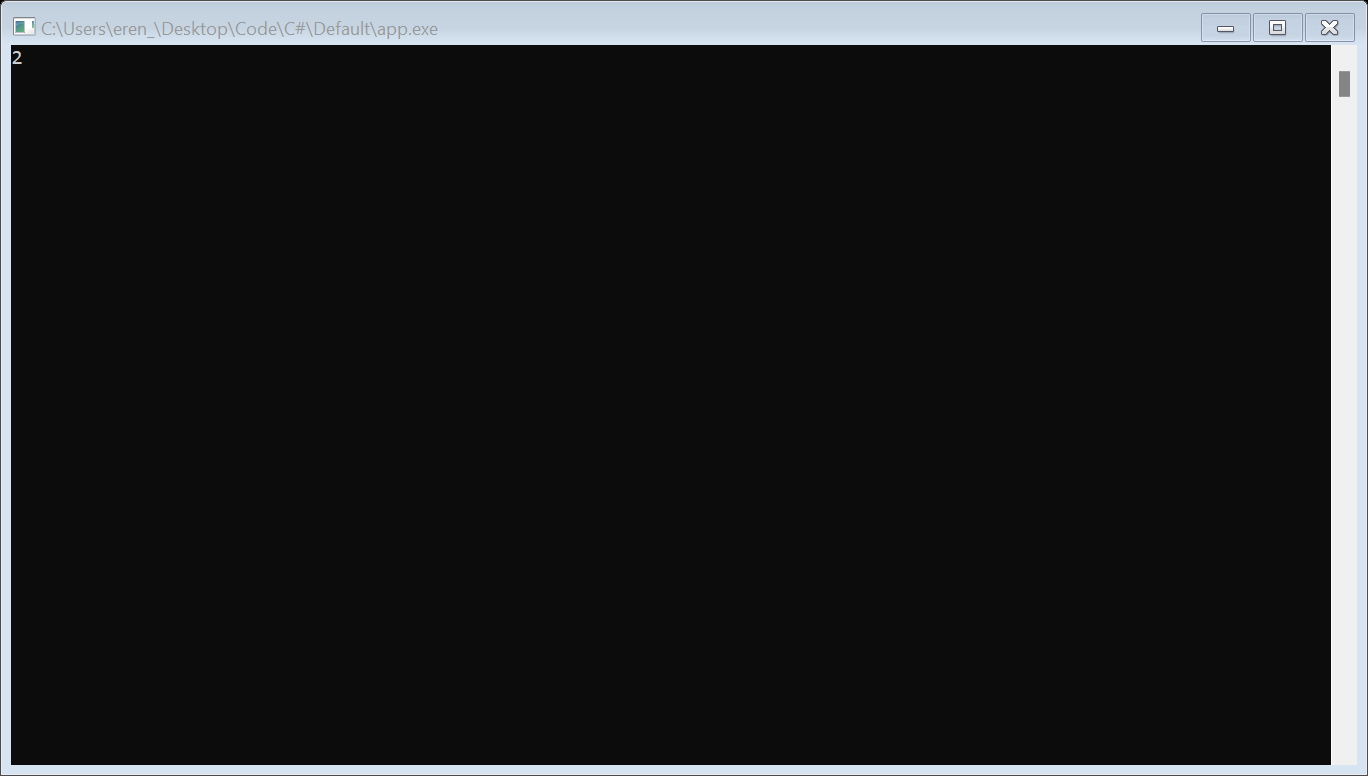
}

int DivideNumbers(int x, int y){

return x / y;

}

}



* delegate içindeki metodlar üzerinde gezinebilirsiniz. Bunun için delegate referansı üzerinden GetInvocationList() metodu kullanılır. Bu metod geriye Delegate[] dizisi döndürür.

Example

Delegate[] arr = delegate1.GetInvocationList();

foreach(Delegate item in arr){

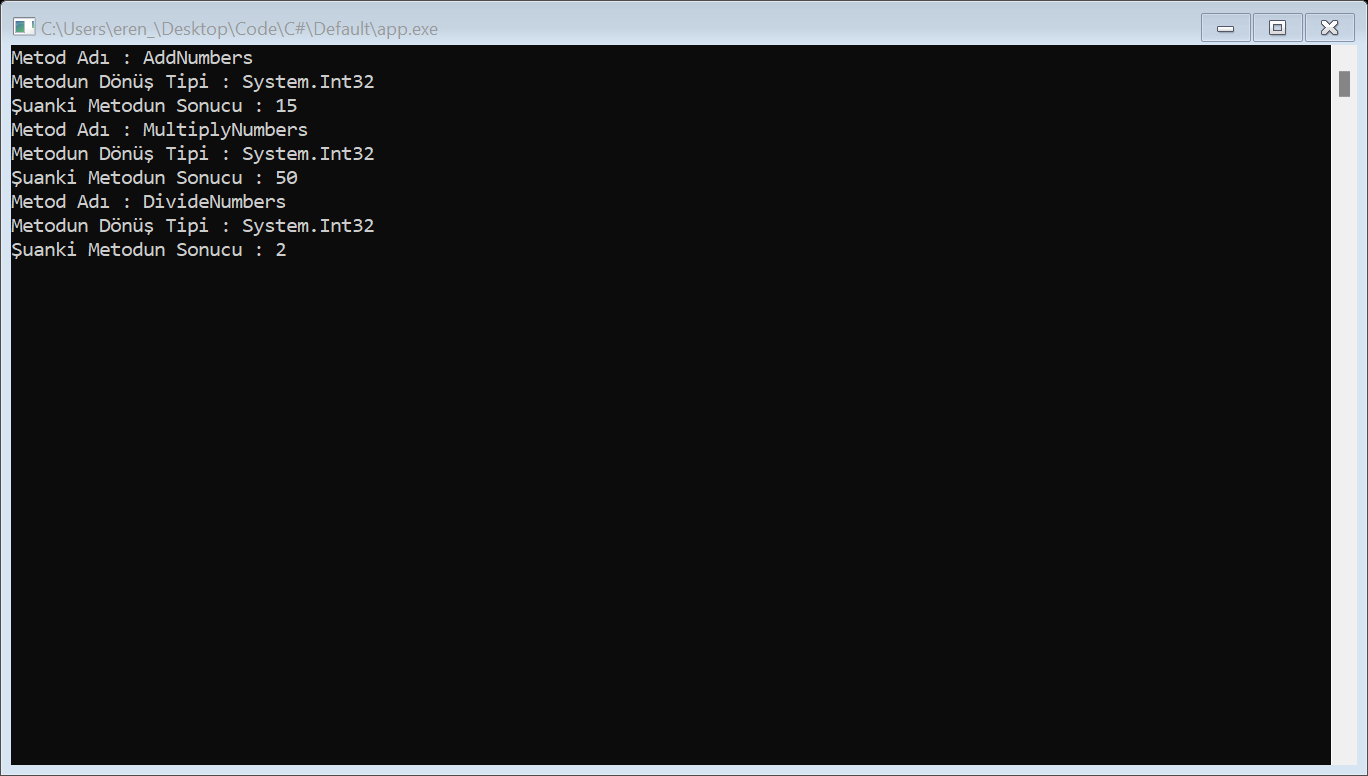
Console.WriteLine("Metod Adı : " + item.Method.Name);

Console.WriteLine("Metodun Dönüş Tipi : " + item.Method.ReturnType);

int result = (int)item.DynamicInvoke(10, 5);

Console.WriteLine("Şuanki Metodun Sonucu : " + result.ToString());

}



Events

eventlar delegeler ile birlikte kullanılır.

* event’in tipi kendisine metod bağlayacak delegate’in tipinde olur.
* eventlar erişim belirteçlerine sahip olabilirler.

Example

class Person{

public string name;

private int \_age;

public int age{

get{

return \_age;

}

set{

\_age = value;

if(value > 18){

Adult();

}

}

}

public delegate void TestHandler();

public event TestHandler Adult;

}

class Program{

static void Main(){

Person obj = new Person();

obj.Adult += new Person.TestHandler(Method1);

obj.age = 20; *//Event Çalıştı.*

Console.ReadLine();

}

static void Method1(){

Console.Write("Event Çalıştı.");

}

}