Object Class

Object sınıfın orijinal halinin bir kısmı aşağıda verilmiştir.

Example

public class Object

{

public virtual int GetHashCode()

{

}

public virtual string ToString()

{

}

public Type GetType()

{

}

public virtual bool Equals(Object obj)

{

}

public static bool Equals(Object objA, Object objB)

{

}

public static bool ReferenceEquals(Object objA, Object objB)

{

}

}

* C#’da tüm sınıflar Object sınıfının mirasçısıdır.
* Object sınıfı içerisinde virtual olarak tanımlanan metodları kendi sınıflarınız içerisinde override edebilirsiniz.
* Object sınıfının ToString() metodu varsayılan olarak olarak kendi sınıfınızın tam adını vermektedir.

Example

namespace Main

{

class Program

{

static void Main(string[] args)

{

Class1 obj = new Class1();

Console.WriteLine(obj.ToString()); //Main.Class1

Console.ReadLine();

}

}

class Class1

{

}

}

Kendi sınıfınız içerisinde public override yazıp ToString() metodunu seçtiğinizde Visual Studio otomatik olarak ToString() metodunun Object sınıfından alınan orijinal halini hazırlar.

Example

public override string ToString()

{

return base.ToString();

}

Görüldüğü gibi sınıfın adı geriye döndürülmektedir. İsterseniz bu metodu override ederek içerisinde başka işlemler yapabilirsiniz.

* Object sınıfının GetType() metodu bir verinin hangi tipte olduğu öğrenilmek istendiğinde kullanılmaktadır.
* Equals() metodu ise kendisine parametre olarak verilen iki değerin hem referanslarına hem de içeriklerine bakarak karşılaştırma yapmakta ve geriye bool bir değer döndürmektedir.

Example

Object obj1 = new Object();

Object obj2 = new Object();

Console.WriteLine(Object.Equals(obj1, obj2)); //False

Yukarıdaki örnek aşağıdaki gibi düzenlendiğinde Equals() metodu geriye True değerini döndürmektedir. Değişkenlerin referansları farklı bile olsa aynı değeri içerdikleri için Equals() metodu True değer üretmektedir.

Example

Object obj1 = new Object();

Object obj2 = new Object();

obj1 = "Karpuz";

obj2 = "Karpuz";

Console.WriteLine(Object.Equals(obj1, obj2)); //True

Example

Object obj1 = new Object();

Object obj2 = new Object();

obj1 = null;

obj2 = null;

Console.WriteLine(Object.Equals(obj1, obj2)); //True

int türündeki değişkenleri karşılaştırmak için aşağıdaki örneği hazırladık.

Example

int number1 = 1;

int number2 = 1;

Console.WriteLine(Object.Equals(number1, number2)); //True

int türündeki değişkenlere constructor tarafından başlangıç değeri aktarıldığı için aşağıdaki örnekte Equals() metodu geriye True değer döndürmüştür.

Example

int number1 = new int();

int number2 = new int();

Console.WriteLine(Object.Equals(number1, number2)); //True

string türündeki değişkenleri karşılaştırmak için aşağıdaki örneği hazırladık.

Example

string str1 = "Edirne";

string str2 = "Edirne";

Console.WriteLine(Object.Equals(str1, str2)); //True

Equals() metodunun kendi oluşturduğumuz tiplere davranışını göstermek için aşağıdaki örnekleri hazırladık.

Example

class Program

{

static void Main(string[] args)

{

Class1 obj1 = new Class1();

Class1 obj2 = new Class1();

Console.WriteLine(Object.Equals(obj1, obj2)); //False

Console.ReadLine();

}

}

class Class1

{

public string name;

}

Example

class Program

{

static void Main(string[] args)

{

Class1 obj1 = new Class1();

Class1 obj2 = new Class1();

obj1.name = "Ahmet";

obj2.name = "Ahmet";

Console.WriteLine(Object.Equals(obj1, obj2)); //False

Console.ReadLine();

}

}

class Class1

{

public string name;

}

Yukarıdaki örneklerde nesnelerimiz aynı olmasına rağmen Equals() metodu geriye false değer döndürdü. Bu problemin çözümü için Object sınıfının statik olmayan Equals() metodunu kendi sınıfımız içerisinde override edip karşılaştırma işlemi için gerekli kodları yazmamız gerekmektedir.

Example

class Program

{

static void Main(string[] args)

{

Class1 obj1 = new Class1();

Class1 obj2 = new Class1();

obj1.name = "Ahmet";

obj2.name = "Ahmet";

Console.WriteLine(obj1.Equals(obj2)); //True

Console.ReadLine();

}

}

class Class1

{

public string name;

public override bool Equals(object obj)

{

Class1 E = (Class1)obj;

if(E.name == this.name)

{

return true;

}

else

{

return false;

}

}

}

* Object sınıfının statik ReferenceEquals() metodu kendisine parametre olarak verilen iki değerin referanslarını karşılaştırıp geriye bool bir değer döndürmektedir.

Example

int number1 = 1;

int number2 = 1;

Console.WriteLine(Object.ReferenceEquals(number1, number2)); //False

Yukarıdaki örnekte iki değişkenin referansları farklı olduğu için geriye False değer dönmektedir.

Konu string türündeki veriler olduğunda ReferenceEquals() metodu farklı davranmaktadır.

Example

string str1 = "Elma";

string str2 = "Elma";

Console.WriteLine(Object.ReferenceEquals(str1, str2)); //True

Bu değişkenlerin referansları farklı olmasına rağmen ReferenceEquals() metodu geri True değer döndürmektedir. Çünkü String sınıfı içerisinde ReferenceEquals() metodu override edilmiş ve aynı içeriğe sahip string verilerin birbirine eşit olması sağlanmıştır. Aynı durum object türündeki veriler için de geçerlidir.

Example

object obj1 = "Elma";

object obj2 = "Elma";

Console.WriteLine(Object.ReferenceEquals(obj1, obj2)); //True

ReferenceEquals() metodunu kendi hazırladığınız tipler üzerinde kullanabilirsiniz.

Example

class Program

{

static void Main(string[] args)

{

Class1 obj1 = new Class1();

Class1 obj2 = new Class1();

obj1.name = "Emre";

obj2.name = "Emre";

Console.WriteLine(Object.ReferenceEquals(obj1, obj2)); //False

Console.ReadLine();

}

}

class Class1

{

public string name;

}