Generic Class

Generic sınıflar sayesinde nesne hazırlama işlemi sırasında programcıya tip seçme imkanı verilmektedir.

Example

class Class1<T>

{

}

* Generic sınıflar hazırlanırken diamond operatörü(<>) kullanılmaktadır.
* Generic sınıflar hazırlanırken kullanılan T harfi tip anlamına gelmektedir. Burada başka bir harf ya da kelime kullanılabilmektedir ancak T harfi C# programcıları tarafından yaygın olarak kullanılmaktadır.
* Diamond operatörü içerisinde birden fazla tip kullanılabilmektedir.

Example

class Class1<T, T2>

{

}

* İsimleri aynı ancak parametre yapıları farklı olan generic sınıflar hazırlanabilmektedir.

Example

class Class1<T>

{

}

class Class1<T, T2>

{

}

* Generic sınıflar sayesinde kendi koleksiyon sınıflarınızı hazırlayabilirsiniz.

Example

class Program

{

static void Main()

{

Class1<int> numbers = new Class1<int>();

Console.WriteLine(numbers.Length); //4

numbers.Add(10);

numbers.Add(20);

numbers.Add(30);

numbers.Add(40);

Console.WriteLine(numbers.Length); //8

Console.WriteLine(numbers[0]); //10

Console.WriteLine(numbers[1]); //20

Console.WriteLine(numbers[2]); //30

Console.WriteLine(numbers[3]); //40

Console.ReadLine();

}

}

class Class1<T>

{

private T[] arr;

private int index;

public Class1()

{

arr = new T[4];

}

public Class1(int capacity)

{

arr = new T[capacity];

}

public int Length

{

get

{

return arr.Length;

}

}

public T this[int index]

{

get

{

return arr[index];

}

}

public void Add(T item)

{

arr[index] = item;

index++;

if(index >= arr.Length)

{

T[] temp\_arr = new T[arr.Length \* 2];

for(int i = 0; i < arr.Length; i++)

{

temp\_arr[i] = arr[i];

}

arr = temp\_arr;

}

}

}

* Generic sınıflarda kalıtım:

Normal bir sınıf generic bir sınıfı miras alabilir. Bu durumda generic sınıfın hangi tipteki versiyonuna gerek duyulduğu belirtilmelidir.

Example

class Class1<T>

{

}

class Class2 : Class1<int>

{

}

Bir generic sınıf başka bir generic sınıfın mirasçısı olabilir. Bu durumda sınıfların parametre yapıları aynı olmalıdır ya da hangi parametrelerin base class’a ait olduğu işaret edilmelidir.

Example

class Class1<T>

{

}

class Class2<T> : Class1<T>

{

}

Example

class Class1<T, T2>

{

}

class Class2<T,T2, T3> : Class1<T2, T3>

{

}

* Generic sınıflarda kullanılacak tiplere kısıtlamalar uygulayabilirsiniz. Bunun için where anahtar kelimesinden yararlanılmaktadır. Birden fazla kısıtlama için virgül kullanılmaktadır.

Example

//kullanılacak tip class türünde olmalıdır.

class Class1<T> where T : class

{

}

//kullanılacak tip struct türünde olmalıdır.

class Class2<T> where T : struct

{

}

Generic Interface

Example

interface IInterface1<T>

{

T Add(T x, T y);

}

class Class1 : IInterface1<int>

{

public int Add(int x, int y)

{

return x + y;

}

}

* Generic arayüzler normal sınıflara uygulanabildiği gibi generic sınıflara da uygulanabilmektedir.

Generic Delegate

Example

class Program

{

public delegate T Delegate1<T>(T x, T y);

public static int AddNumbers(int x, int y)

{

return x + y;

}

static void Main()

{

Delegate1<int> delegate1 = new Delegate1<int>(AddNumbers);

Console.WriteLine(delegate1(1, 2));

Console.ReadLine();

}

}

Generic Method

Example

class Class1

{

public void Add<T>(T x, T y)

{

}

}

* Generic metodlara where operatörü ile kısıtlamalar uygulayabilirsiniz.