Класс можно объявлять как static. Статический класс обладает двумя основными

свойствами. Во-первых, объекты статического класса создавать нельзя. И во-вторых,

статический класс должен содержать только статические члены.

В таком классе все члены должны быть объявлены как static. Ведь если класс становится

статическим, то это означает, что статическими становятся и все его члены.

Статические классы применяются главным образом в двух случаях. Во-первых,

статический класс требуется при создании *метода расширения.* Методы расширения

связаны в основном с языком LINQ и поэтому подробнее рассматриваются в главе 19.

И во-вторых, статический класс служит для хранения совокупности связанных друг с

другом статических методов.

В приведенном ниже примере программы класс NumericFn типа static служит

для хранения ряда статических методов, оперирующих числовым значением.

А поскольку все члены класса NumericFn объявлены как static, то этот класс также

объявлен как static, чтобы исключить получение экземпляров его объектов. Таким

образом, класс NumericFn выполняет организационную роль, предоставляя удобные

средства для группирования логически связанных методов. (***glava8\_18***)

static class NumericFn

{

//возвратить обратное числовое значение

static public double Reciprocal(double num)

{

return 1 / num;

}

//вовзратить дробную часть числового значения

static public double FracPart(double num)

{

return num - (int)num;

}

//return true, if num % == 0

static public bool IsEven(double num)

{

return (num % 2) == 0 ? true : false;

}

//return true if num % != 0

static public bool IsOdd(double num)

{

return !IsEven(num);

}

}

class StaticClassDemo

{

static void Main()

{

Console.WriteLine("Backside of 5 = "

+ NumericFn.Reciprocal(5));

Console.WriteLine("Fraction part of 4.234 = "

+ NumericFn.FracPart(4.234));

if(NumericFn.IsEven(10))

Console.WriteLine("10 is Even");

if(NumericFn.IsOdd(5))

Console.WriteLine("5 is Odd"); }}

Класс NumericFn является статическим, и поэтому

любая попытка создать объект этого класса может привести к ошибке во время

компиляции. Ошибкой будет также считаться попытка сделать нестатическим член

класса NumericFn.

И последнее замечание: несмотря на то, что для статического класса не допускается

наличие конструктора экземпляра, у него может быть статический конструктор.