Структура Decimal немного сложнее, чем ее аналоги для целочисленных типов

данных, а также типов данных с плавающей точкой. Она содержит немало конструкторов,

полей, методов и операторов, способствующих использованию типа decimal

вместе с другими числовыми типами, поддерживаемыми в С#. Так, целый ряд методов

из этой структуры обеспечивает преобразование типа decimal в другие числовые

типы.

В структуре Decimal определено восемь открытых конструкторов. Ниже приведены

шесть наиболее часто используемых из них.

public Decimal(int значение)

public Decimal(uint значение)

public Decimal(long значение)

public Decimal(ulong значение)

public Decimal(float значение)

public Decimal(double значение)

Каждый из этих конструкторов создает объект типа Decimal из значения указанного

типа.

Кроме того, объект типа Decimal может быть создан из отдельно указываемых составляющих

с помощью следующего конструктора.

public Decimal(int lo, int mid, int hi, bool IsNegative, byte scale)

Десятичное значение состоит из трех частей. Первую часть составляет 96-разрядное

целое значение, вторую — флаг знака, третью — масштабный коэффициент. В частности,

96-разрядное целое значение передается конструктору тремя 32-разрядными

фрагментами с помощью параметров *lo, mid* и *hi*; знак флага — с помощью параметра

*IsNegative,* причем логическое значение false этого параметра обозначает

положительное число, тогда как логическое значение true обозначает отрицательное

число; а масштабный коэффициент — с помощью параметра *scale,* принимающего

значения от 0 до 28. Этот коэффициент обозначает степень числа 10 (т.е. 10scale), на которую

делится число для получения его дробной части.

Вместо того чтобы передавать каждую составляющую объекта типа Decimal отдельно,

все его составляющие можно указать в массиве, используя следующий конструктор.

public Decimal(int[] bits)

Три первых элемента типа int в массиве *bits* содержат 96-разрядное целое значение;

31-й разряд содержимого элемента bits[3] обозначает флаг знака (0 — положительное

число, 1 — отрицательное число); а в разрядах 16-23 содержится масштабный

коэффициент.

В структуре Decimal реализуются следующие интерфейсы: IComparable,

IComparable<decimal>, IConvertible, IFormattable, IEquatable<decimal>,

а также IDeserializationCallback.

В приведенном ниже примере программы значение типа decimal формируется

вручную.

(***glava21\_1***)

class CreateDec

{

static void Main()

{

decimal d = new decimal(12345,0,0,false,2);

Console.WriteLine(d);

}

}

В данном примере значение 96-разрядного целого числа равно 12345. У него положительный

знак и два десятичных разряда в дробной части.

Методы, определенные в структуре Decimal, приведены в табл. 21.7, а поля —

в табл. 21.8. Кроме того, в структуре Decimal определяется обширный ряд операторов

и преобразований, позволяющих использовать десятичные значения вместе со значениями

других типов в выражениях. Правила, устанавливающие порядок присваивания

десятичных значений и их применения в выражениях, представлены в главе 3.

