Структура Char соответствует типу char и применяется довольно часто, поскольку

предоставляет немало методов, позволяющих обрабатывать символы и распределять

их по отдельным категориям. Например, символ строчной буквы можно преобразовать

в символ прописной буквы, вызвав метод ToUpper(), а с помощью метода

IsDigit() можно определить, обозначает ли символ цифру.

Методы, определенные в структуре Char, приведены в табл. 21.9. Следует, однако,

иметь в виду, что некоторые методы, например ConvertFromUtf32()

и ConvertToUtf32(), позволяют обрабатывать символы уникода в форматах UTF-16

и UTF-32. Раньше все символы уникода могли быть представлены 16 разрядами, что

соответствует величине значения типа char. Но несколько лет назад набор символов

уникода был расширен, для чего потребовалось более 16 разрядов. Каждый символ

уникода представлен *кодовой точкой,* а способ кодирования кодовой точки зависит от

используемого формата преобразования уникода (UTF). Так, в формате UTF-16 для

кодирования большинства кодовых точек требуется одно 16-разрядное значение, а для

кодирования остальных кодовых точек — два 16-разрядных значения. Если для этой

цели требуются два 16-разрядных значения, то для их представления служат два значения

типа char. Первое символьное значение называется *старшим суррогатом,* а второе

— *младшим суррогатом.* В формате UTF-32 каждая кодовая точка кодируется с помощью

одного 32-разрядного значения. В структуре Char предоставляются все необходимые

средства для преобразования из формата UTF-16 в формат UTF-32 и обратно.

В отношении методов структуры Char необходимо также отметить следующее:

в используемых по умолчанию формах методов ToUpper() и ToLower() применяются

текущие настройки культурной среды (языки и региональные стандарты), чтобы

указать способ представления символов верхнего и нижнего регистра. На момент написания

этой книги рекомендовалось явно указывать текущие настройки культурной

среды, используя для этой цели параметр типа CultureInfo во второй форме обоих

упоминаемых методов. Класс CultureInfo относится к пространству имен System.

Globalization, а для указания текущей культурной среды следует передать свойство

CultureInfo.CurrentCulture соответствующему методу.

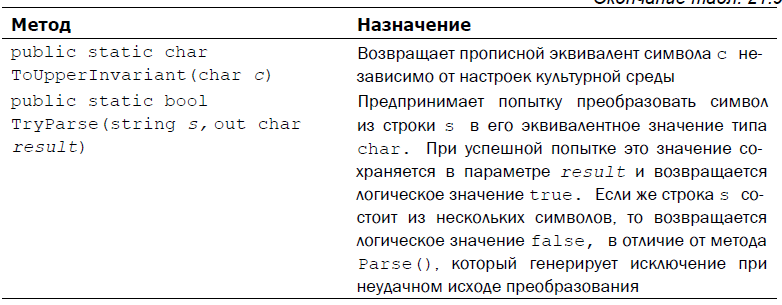
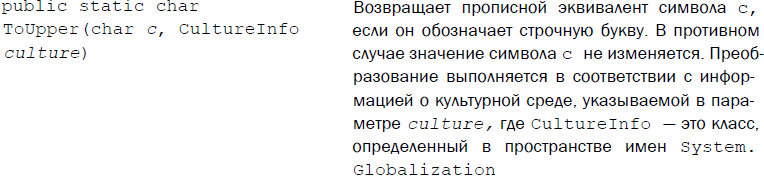
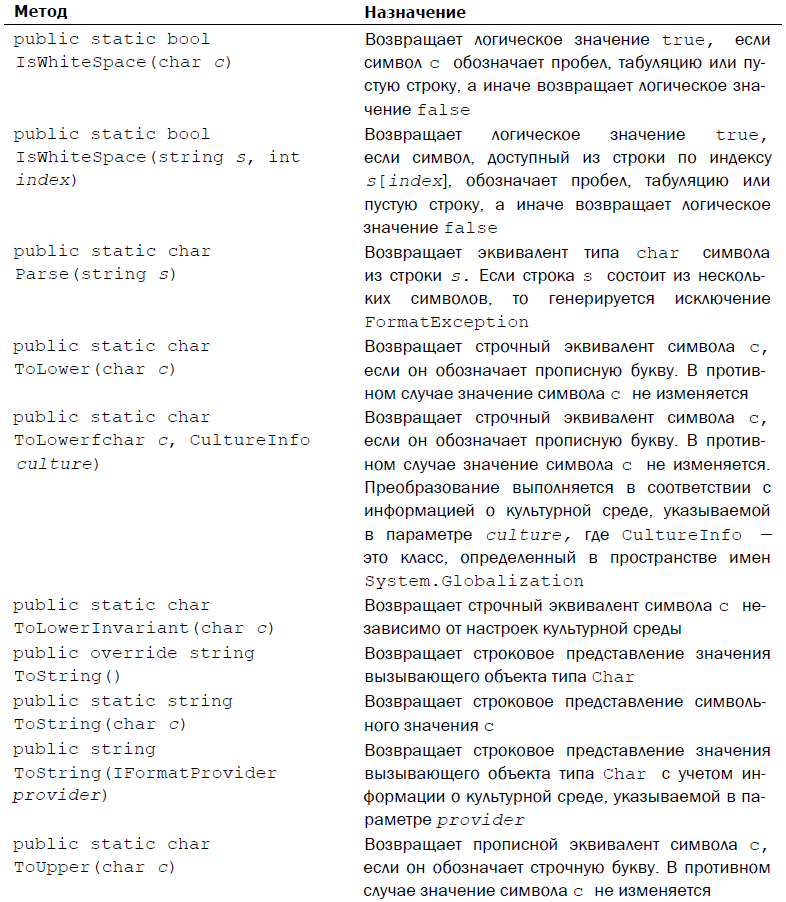
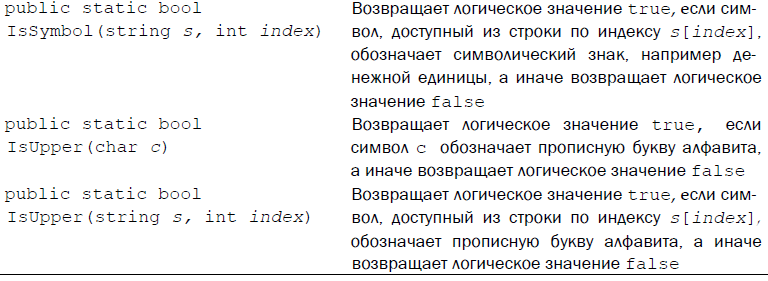
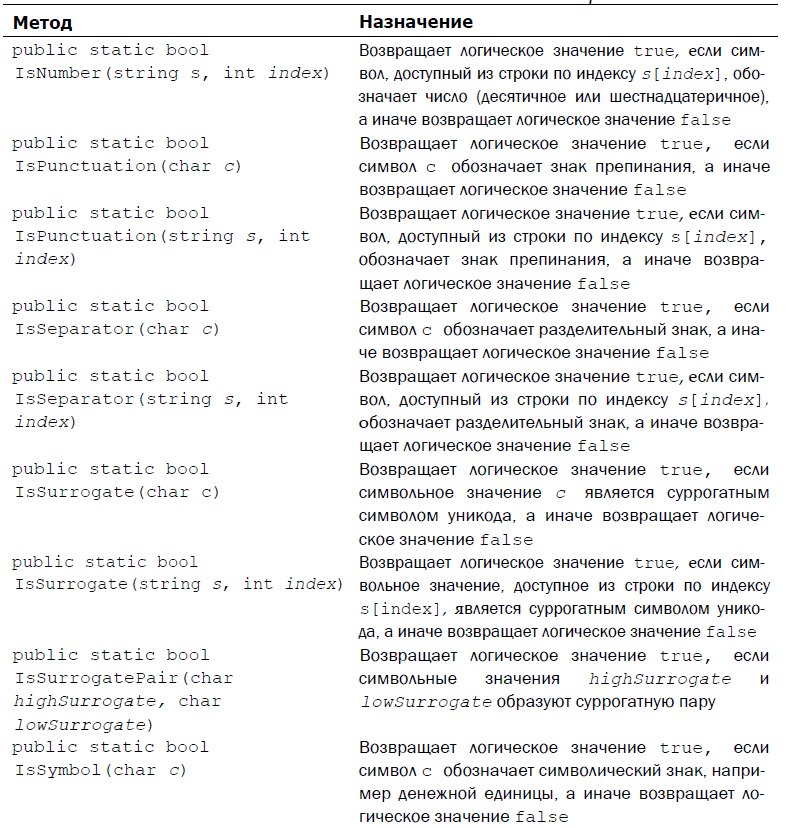
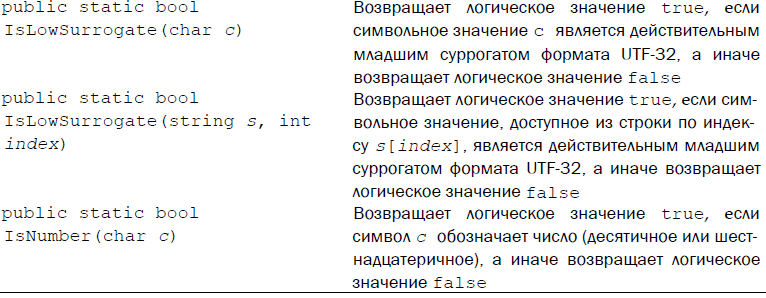
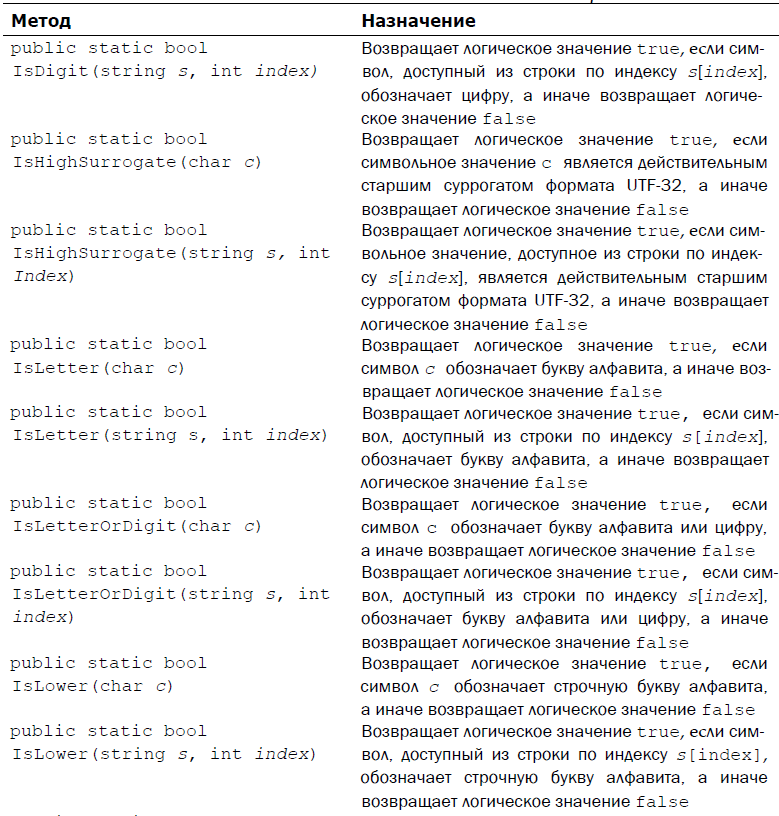
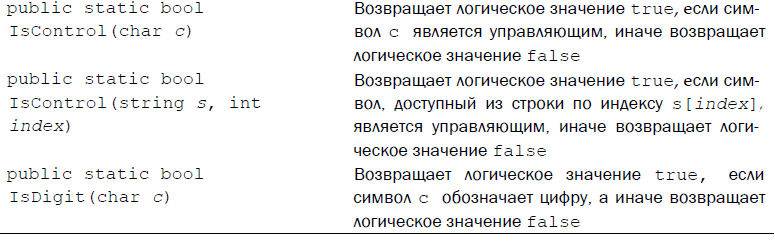
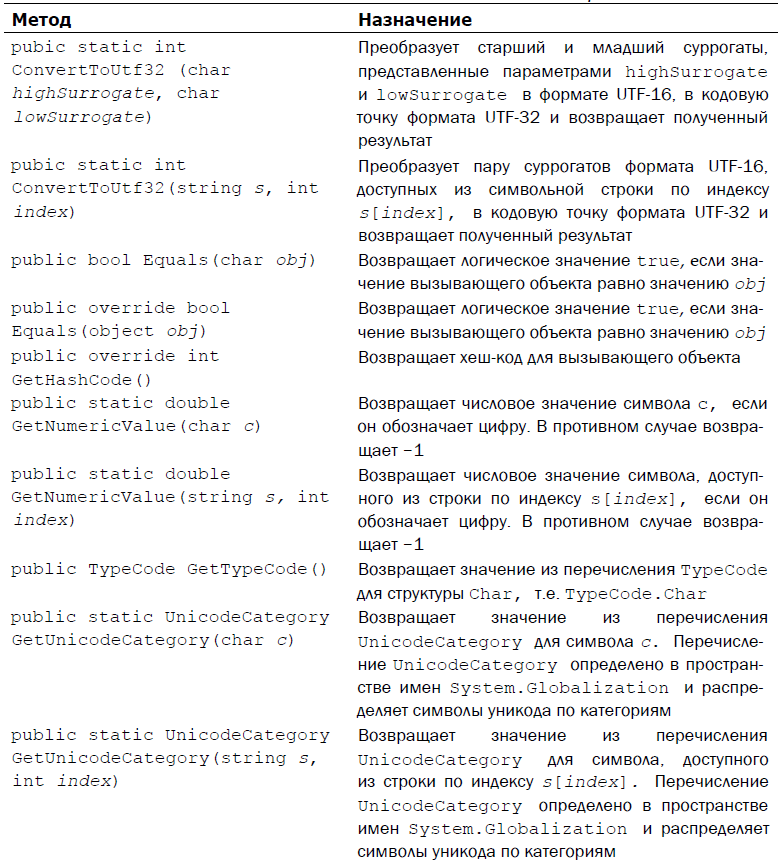
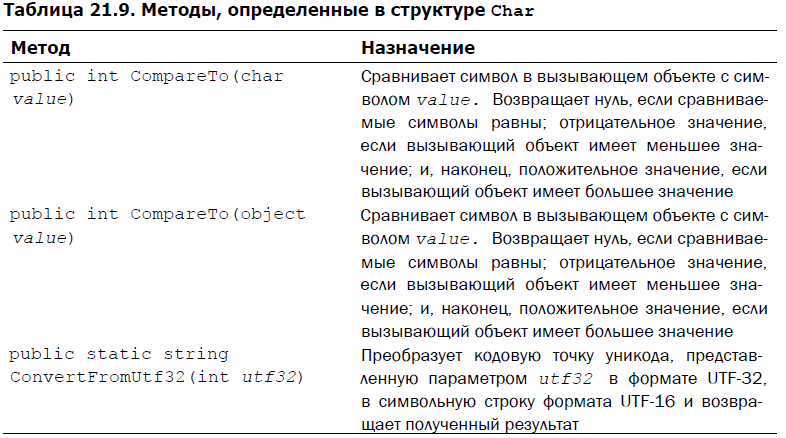
В структуре Char определены также следующие поля.

public const char MaxValue

public const char MinValue

Кроме того, в структуре Char реализуются следующие интерфейсы: IComparable,

IComparable<char>, IConvertible и IEquatable<char>.



Ниже приведен пример программы, в которой демонстрируется применение нескольких

методов, определенных в структуре Char.

(***glava20\_2***)

class CharDemo

{

static void Main()

{

string str = "This is a test. $23";

int i;

for(i = 0; i< str.Length; i++)

{

Console.Write(str[i] + " beeing");

if (Char.IsDigit(str[i]))

Console.Write(" number");

if (Char.IsLetter(str[i]))

Console.Write(" letter");

if (Char.IsLower(str[i]))

Console.Write(" lower");

if (Char.IsUpper(str[i]))

Console.Write(" upper");

if(Char.IsSymbol(str[i]))

Console.Write(" symbol");

if (Char.IsSeparator(str[i]))

Console.Write(" separator");

if (Char.IsWhiteSpace(str[i]))

Console.Write(" space");

if (Char.IsPunctuation(str[i]))

Console.Write(" mark of punctuation");

Console.WriteLine();

}

Console.WriteLine("String str: " + str);

//to upper

string newstr = "";

for (i = 0; i < str.Length; i++)

newstr += Char.ToUpper(str[i], CultureInfo.CurrentCulture);

Console.WriteLine("After changes: " + newstr);

}

}