Оператор foreach служит для циклического обращения к элементам *коллекции,*

представляющей собой группу объектов. В C# определено несколько видов коллекций,

каждая из которых является массивом. Ниже приведена общая форма оператора

цикла foreach.

***foreach*** (***тип*** *имя\_переменной\_цикла* ***in*** *коллекция*) *оператор;*

Здесь *тип имя\_переменной\_цикла* обозначает тип и имя переменной управления

циклом, которая получает значение следующего элемента коллекции на каждом

шаге выполнения цикла foreach. А *коллекция* обозначает циклически опрашиваемую

коллекцию, которая здесь и далее представляет собой массив. Следовательно, *тип*

переменной цикла должен соответствовать типу элемента массива. Кроме того, *тип*

может обозначаться ключевым словом ***var***. В этом случае компилятор определяет тип

переменной цикла, исходя из типа элемента массива. Но, как правило, тип указывается явным образом.

Оператор цикла ***foreach*** действует следующим образом. Когда цикл начинается,

первый элемент массива выбирается и присваивается переменной цикла. На каждом

последующем шаге итерации выбирается следующий элемент массива, который сохраняется

в переменной цикла. Цикл завершается, когда все элементы массива окажутся

выбранными. Следовательно, оператор foreach циклически опрашивает массив по

отдельным его элементам от начала и до конца.

Следует, однако, иметь в виду, что переменная цикла в операторе ***foreach*** служит

только для чтения. Это означает, что, присваивая этой переменной новое значение,

нельзя изменить содержимое массива. (***glava7\_10***)

class ForeachDemo

{

static void Main()

{

int sum = 0;

int [] nums = new int [10];

//assign array

for (int i = 0; i < 10; i++)

nums[i] = i;

//use foreach - output

//elements of array and sum

foreach (int x in nums)//читает все эл-ты в массиве и суммирует их

{

Console.WriteLine("Element = " + x);

sum += x;

}

Console.WriteLine("Sum = " + sum);

}

}

Как видите, оператор foreach циклически опрашивает массив по порядку индексирования

от самого первого до самого последнего его элемента.

Несмотря на то что цикл ***foreach*** повторяется до тех пор, пока не будут опрошены

все элементы массива, его можно завершить преждевременно, воспользовавшись оператором

***break***. (***glava7\_10.1***)

foreach (int x in nums)

{

Console.WriteLine("Element = " + x);

sum += x;

if (x == 4) break; //break the cycle if array = 4

}

Оператор цикла ***foreach*** можно также использовать для циклического обращения

к элементам многомерного массива. В этом случае элементы многомерного массива

возвращаются по порядку следования строк от первой до последней, как демонстрирует

приведенный ниже пример программы. (***glava7\_10.2***)

int [,] nums = new int[3, 5];

//assign array

for (int i = 0; i < 3; i++)

for (int j = 0; j < 5; j++)

nums [i, j] = (i + 1) \* (j + 1);

//use foreach - output

//elements of array and sum

foreach (int x in nums)

{

Console.WriteLine("Element = " + x);

sum += x;

}

Оператор ***foreach*** допускает циклическое обращение к массиву только в определенном

порядке: от начала и до конца массива, поэтому его применение кажется, на

первый взгляд, ограниченным. Но на самом деле это не так. В большом числе алгоритмов,

самым распространенным из которых является алгоритм поиска, требуется

именно такой механизм. В качестве примера ниже приведена программа, в которой

цикл ***foreach*** используется для поиска в массиве определенного значения. Как только

это значение будет найдено, цикл прервется. (***glava7\_10.3***)

int [] nums = new int[10];

foreach (int x in nums)

{

if (x == 5)

{

found = true;

break;

}

}

Оператор цикла foreach отлично подходит для такого применения, поскольку

при поиске в массиве приходится анализировать каждый его элемент. К другим примерам

применения оператора цикла foreach относится вычисление среднего, поиск

минимального или максимального значения среди ряда заданных значений, обнаружение

дубликатов и т.д.