**Класс Queue<T>**

Класс Queue<T> является обобщенным эквивалентом класса необобщенной коллекции

Queue. В нем поддерживается очередь в виде списка, действующего по принципу

"первым пришел — первым обслужен". В этом классе реализуются интерфейсы

ICollection, IEnumerable и IEnumerable<T>. Кроме того, в классе Queue<T> непосредственно

реализуются методы Clear(), Contains() и CopyTo(), определенные

в интерфейсе ICollection<T>. А методы Add() и Remove() в этом классе не

поддерживаются, как, впрочем, и свойство IsReadOnly. Коллекция класса Queue<T>

имеет динамический характер, расширяясь по мере необходимости, чтобы вместить

все элементы, которые должны храниться в ней. В классе Queue<T> определяются следующие

конструкторы.

public Queue()

public Queue(int capacity)

public Queue(IEnumerable<T> collection)

В первой форме конструктора создается пустая очередь с выбираемой по умолчанию

первоначальной емкостью, а во второй форме — пустая очередь, первоначальный

размер которой определяет параметр *capacity.* И в третьей форме создается очередь,

содержащая элементы коллекции, определяемой параметром *collection.* Ее первоначальная

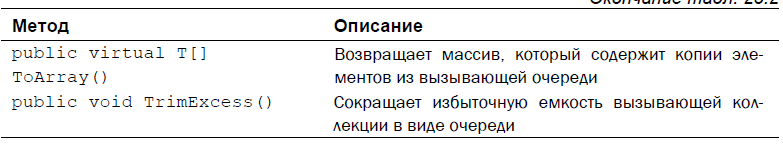
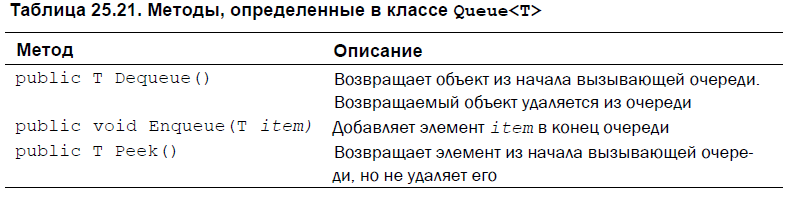
емкость равна количеству указанных элементов.

В классе Queue<T> определяется ряд собственных методов, помимо тех, что

уже объявлены в интерфейсах, которые в нем реализуются, а также в интерфейсе

ICollection<T>. Некоторые из наиболее часто используемых методов этого

класса перечислены в табл. 25.21.



(***glava25\_15***)

class GenQueueDemo

{

static void Main()

{

Queue<double> q = new Queue<double>();

q.Enqueue(98.6);

q.Enqueue(212.0);

q.Enqueue(32.0);

q.Enqueue(3.1416);

double sum = 0.0;

Console.Write("Queue content: ");

while (q.Count > 0)

{

double val = q.Dequeue();

Console.Write(val + " ");

sum += val;

}

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

}

}