**Класс Stack**

Как должно быть известно большинству читателей, *стек* представляет собой список,

действующий по принципу "первым пришел — последним обслужен".

Стек относится к одним из самых важных структур данных в вычислительной технике.

Он нередко применяется, среди прочего, в системном программном обеспечении,

компиляторах, а также в программах отслеживания в обратном порядке на основе искусственного

интеллекта

Класс коллекции, поддерживающий стек, носит название Stack. В нем реализуются

интерфейсы ICollection, IEnumerable и ICloneable. Этот класс создает динамическую

коллекцию, которая расширяется по мере потребности хранить в ней вводимые

элементы. Всякий раз, когда требуется расширить такую коллекцию, ее емкость

увеличивается вдвое.

В классе Stack определяются следующие конструкторы.

public Stack()

public Stack(int initialCapacity)

public Stack(ICollection col)

В первой форме конструктора создается пустой стек, во второй форме — пустой

стек, первоначальный размер которого определяет первоначальная емкость, задаваемая

параметром *initialCapacity,* и в третьей форме — стек, содержащий элементы

указываемой коллекции *col.* Его первоначальная емкость равна количеству указанных

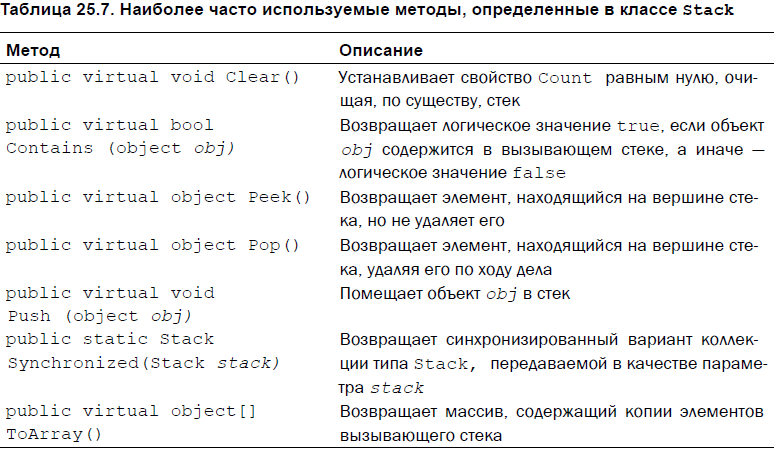
элементов.

В классе Stack определяется ряд собственных методов, помимо тех, что уже объявлены

в интерфейсах, которые в нем реализуются. Некоторые из наиболее часто используемых

методов этого класса приведены в табл. 25.7.

А если вызвать метод Pop() или Рееk(), когда вызывающий стек пуст, то сгенерируется исключение InvalidOperationException.



В приведенном ниже примере программы создается стек, в который помещается

несколько целых значений, а затем они извлекаются обратно из стека.

(***glava25\_6***)

class StackDemo

{

static void ShowPush(Stack st, int a)

{

st.Push(a);

Console.WriteLine("Place in stack: Push(" + a + ")");

Console.Write("Content of stack: ");

foreach (var i in st)

Console.Write(i + " ");

Console.WriteLine();

}

static void ShowPop(Stack st)

{

Console.Write("Pop out of stack: Pop -> ");

int a = (int)st.Pop();

Console.WriteLine(a);

Console.Write("Content of stack: ");

foreach (var i in st)

Console.Write(i + " ");

Console.WriteLine();

}

static void Main()

{

Stack st = new Stack();

foreach (var i in st)

Console.Write(i + " ");

Console.WriteLine();

ShowPush(st, 22);

ShowPush(st, 65);

ShowPush(st, 91);

ShowPop(st);

ShowPop(st);

ShowPop(st);

try

{

ShowPop(st);

}

catch(InvalidOperationException)

{

Console.WriteLine("Stack is empty.");

}

}

}