**Класс HashSet<T>**

В классе HashSet<T> поддерживается коллекция, реализующая множество.

Для хранения элементов этого множества в нем используется хеш-таблица. В классе

HashSet<T> реализуются интерфейсы ICollection<T>, ISet<T>, IEnumerable,

IEnumerable<T>, ISerializable, а также IDeserializationCallback. В коллекции

типа HashSet<T> реализуется множество, все элементы которого являются уникальными.

Иными словами, дубликаты в таком множестве не допускаются. Порядок

следования элементов во множестве не указывается. В классе HashSet<T> определяется

полный набор операций с множеством, определенных в интерфейсе ISet<T>, включая

пересечение, объединение и разноименность. Благодаря этому класс HashSet<T>

оказывается идеальным средством для работы с множествами объектов, когда порядок

расположения элементов во множестве особого значения не имеет. Коллекция типа

HashSet<T> имеет динамический характер и расширяется по мере необходимости,

чтобы вместить все элементы, которые должны в ней храниться.

Ниже перечислены наиболее употребительные конструкторы, определенные в

классе HashSet<T>.

public HashSet()

public HashSet(IEnumerable<T> collection)

public HashSet(IEqualityCompare comparer)

public HashSet(IEnumerable<T> collection, IEqualityCompare comparer)

В первой форме конструктора создается пустое множество, а во второй форме —

множество, состоящее из элементов указываемой коллекции *collection.* В третьей

форме конструктора допускается указывать способ сравнения с помощью параметра

*comparer. А* в четвертой форме создается множество, состоящее из элементов указываемой

коллекции *collection,* и используется заданный способ сравнения *comparer.*

Имеется также пятая форма конструктора данного класса, в которой допускается инициализировать

множество последовательно упорядоченными данными.

В классе HashSet<T> реализуется интерфейс ISet<T>, а следовательно, в нем предоставляется

полный набор операций со множествами. В этом классе предоставляется

также метод RemoveWhere(), удаляющий из множества элементы, не удовлетворяющие

заданному условию, или предикату.

Помимо свойств, определенных в интерфейсах, которые реализуются в классе

HashSet<T>, в него введено дополнительное свойство Comparer, приведенное ниже.

public IEqualityComparer<T> Comparer { get; }

Оно позволяет получать метод сравнения для вызывающего хеш-множества.

Ниже приведен конкретный пример применения класса HashSet<T>.

(***glava25\_18***)

class HshSetDemo

{

static void Show(string msg, HashSet<char> set)

{

Console.Write(msg);

foreach (var ch in set)

Console.Write(ch + " ");

Console.WriteLine();

}

static void Main()

{

HashSet<char> setA = new HashSet<char>();

HashSet<char> setB = new HashSet<char>();

setA.Add('A');

setA.Add('B');

setA.Add('C');

setB.Add('C');

setB.Add('D');

setB.Add('E');

Show("Content of setA: ", setA);

Show("Content of setB: ", setB);

setA.SymmetricExceptWith(setB);

Show("Content of setA after Symmetric with setB: ", setA);

setA.UnionWith(setB);

Show("Content of setA after UnionWith with setB: ", setA);

setA.ExceptWith(setB);

Show("Content of setA after ExceptWith with setB: ", setA);

Console.WriteLine();

}

}