**Класс SortedSet<T>**

Класс SortedSet<T> представляет собой новую разновидность коллекции,

введенную в версию 4.0 среды .NET Framework. В нем поддерживается коллекция,

реализующая отсортированное множество. В классе SortedSet<T> реализуются

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

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

типа SortedSet<T> реализуется множество, все элементы которого являются

уникальными. Иными словами, дубликаты в таком множестве не допускаются. В классе

SortedSet<T> определяется полный набор операций с множеством, определенных

в интерфейсе ISet<T>, включая пересечение, объединение и разноименность. Благодаря

тому что все элементы коллекции типа SortedSet<T> сохраняются в отсортированном

порядке, класс SortedSet<T> оказывается идеальным средством для работы

с отсортированными множествами объектов. Коллекция типа SortedSet<T> имеет

динамический характер и расширяется по мере необходимости, чтобы вместить все

элементы, которые должны в ней храниться.

Ниже перечислены четыре наиболее часто используемые конструктора, определенных

в классе SortedSet<T>.

public SortedSet()

public SortedSet(IEnumerable<T> collection)

public SortedSet(IComparer comparer)

public SortedSet(IEnumerable<T> collection, IComparer 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 SortedSet<T> реализуется интерфейс ISet<T>, а следовательно, в нем

предоставляется полный набор операций со множествами. В этом классе предоставляется

также метод GetViewBetween(), возвращающий часть множества в форме объекта

типа SortedSet<T>, метод RemoveWhere(), удаляющий из множества элементы,

не удовлетворяющие заданному условию, или предикату, а также метод Reverse(),

возвращающий объект типа IEnumerable<T>, который циклически проходит множество

в обратном порядке.

Помимо свойств, определенных в интерфейсах, которые реализуются в классе

SortedSet<T>, в него введены дополнительные свойства, приведенные ниже.

public IComparer<T> Comparer { get; }

public T Max { get; }

public T Min { get; }

Свойство Comparer получает способ сравнения для вызывающего множества. Свойство

Мах получает наибольшее значение во множестве, а свойство Min — наименьшее

значение во множестве.

В качестве примера применения класса SortedSet<T> на практике просто замените

обозначение HashSet на SortedSet в исходном коде программы из предыдущего

подраздела, посвященного коллекциям типа HashSet<T>.