عنوان: **SortedSet در دات نت 4**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۲۶ ۱۸:۱۶:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: #C

SortedSet
قرار گرفته در فضای نام System.Collections.Generic دات نت 4، لیستی از اشیاء به صورت خودکار مرتب شده را ارائه می‌دهد. SortedSet نیز همانند HashSet از اعضای منحصربفردی تشکیل خواهد شد اما اینبار به شکلی مرتب شده. برای پیاده سازی آن از red-black tree data structure استفاده شده است که مهم‌ترین مزیت آن امکان افزودن و یا حذف اشیاء به آن بدون کاهش قابل توجه کارآیی برنامه است.

مثال اول:

```
using System;
using System.Collections.Generic;

namespace SortedSetTest
{
   class Program
   {
       static void sample1()
       {
           var setRange = new SortedSet<int> { 2, 5, 6, 2, 1, 4, 8 };

           foreach (var i in setRange)
           {
               Console.WriteLine(i);
           }
       }

       static void Main()
       {
           sample1();
       }
   }
}
```

در این مثال با نحوه‌ی ایجاد این لیست جنریک خود مرتب شونده‌ی تکراری نپذیر (!) آشنا می‌شوید. اگر این مثال را اجرا نمائید، خروجی آن مرتب شده است و همچنین تنها شامل یک عدد 2 است (اعضای تکراری را حذف می‌کند).

مثال دوم:

```
using System;
using System.Collections.Generic;

namespace SortedSetTest
{
   class Program
   {
       static void sample2()
       {
           var setRange = new SortedSet<int>();
           var random = new Random();

           for (int counter = 0; counter < 100; counter++)
           {
               var rnd = random.Next(-180, 181);
               if (!setRange.Add(rnd))
               {
                   Console.WriteLine("Couldn't add {0}", rnd);
               }
           }

           Console.WriteLine("Result set:");
           foreach (var item in setRange)
           {
```

```
                Console.WriteLine(item);
            }
        }

        static void Main()
        {
            sample2();
        }
    }
}
```

در این مثال نحوه‌ی افزودن اعضای مختلف به این لیست ویژه، توسط متد Add آن بیان شده است. اگر آیتمی در این لیست موجود باشد، مجددا اضافه نشده و حاصل متد Add آن، False خواهد بود.

مثال سوم:

اگر از سایر انواع سفارشی تعریف شده استفاده نمائید، باید روش مقایسه‌ی آن‌ها را نیز با پیاده سازی اینترفیس استاندارد IComparable ارائه دهید؛ در غیر اینصورت با خطای At least one object must implement IComparable متوقف خواهید شد.

```
using System;
using System.Collections;
using System.Collections.Generic;

namespace SortedSetTest
{
    class FileInfo
    {
        public string Name { set; get; }
        public long Size { set; get; }
    }

    class FileInfoComparer : IComparer<FileInfo>
    {
        public int Compare(FileInfo x, FileInfo y)
        {
            var caseiComp = new CaseInsensitiveComparer();
            return caseiComp.Compare(x.Name, y.Name);
        }
    }


    class Program
    {

        static void sample3()
        {
            var setRange = new SortedSet<FileInfo>(new FileInfoComparer())
                                {
                                    new FileInfo
                                        {
                                            Name = "file1.txt",
                                            Size = 100
                                        },
                                    new FileInfo
                                        {
                                            Name = "file2.txt",
                                            Size = 10
                                        },
                                        new FileInfo
                                            {
                                            Name = "file3.txt",
                                            Size = 300
                                            }
                                };

            foreach (var item in setRange)
            {
                Console.WriteLine(item.Name);
            }
        }

        static void Main()
        {
            sample3();
```

```
            Console.WriteLine("Press a key...");
            Console.ReadKey();
        }
    }
}
```

در این مثال اشیایی از نوع کلاس FileInfo به لیست ویژه‌ی ما اضافه شده‌اند. برای اینکه امکان مقایسه‌ی آن‌ها فراهم باشد ، کلاس FileInfoComparer با پیاده سازی اینترفیس IComparer ، روش مقایسه دو شیء از این دست را ارائه می‌دهد.