

```

1  自定义格式字符串 IFormattable, IFormatProvider, ICustomFormatter 三接口的实现
2
3  -----
4
5  public interface IFormattable
6  {
7      string ToString(string format, IFormatProvider formatProvider);
8  }
9
10 public interface IFormatProvider
11 {
12     object GetFormat(Type formatType);
13 }
14
15 public interface ICustomFormatter
16 {
17     string Format(string format, object arg, IFormatProvider formatProvider);
18 }
19
20 -----
21
22 定义一个类
23 using System;
24 namespace MyFormat
25 {
26     public class MyClass : System.IFormattable
27     {
28         Double d;
29
30         public MyClass(Double d)
31         {
32             this.d=d;
33         }
34
35         public string ToString(string format, IFormatProvider formatProvider)
36         {
37             return (format=="MyFormat")?"***"+d.ToString(formatProvider):d.ToString(format,formatProvider);
38         }
39     }
40 }
41
42 再到一控制台中
43 System.Globalization.CultureInfo culture=null;
44
45     MyClass myClass=new MyClass(5);
46     //当IFormatProvider为空时,调用的是当前线程关联的文化信息
47     Console.WriteLine("显示中国货币格式:{0}",myClass.ToString("C",null));
48
49     culture=System.Globalization.CultureInfo.CurrentCulture;
50     Console.WriteLine("显示当前系统默认货币格式:{0}",myClass.ToString("C",culture));
51
52     culture=new System.Globalization.CultureInfo("zh-HK");
53     Console.WriteLine("显示香港特别行政区货币格式:{0}",myClass.ToString("C",culture));
54
55     Console.WriteLine("显示我自己定义的货币格式:{0}",myClass.ToString("MyFormat",null));
56
57     Console.ReadLine();
58
59  -----
60
61 如果希望自定义格式化能在多个不同类使用, 那么实现我们应该实现ICustomFormatter接口
62
63 定义一个类
64 using System;
65
66 namespace MyFormat
67 {
68     public class MyBaseFormat : System.ICustomFormatter, System.IFormatProvider
69     {
70         //如果format Type与当前实例类型相同, 则为当前实例, 否则为空引用
71         public object GetFormat(Type format)

```

```

72     {
73         if (format == typeof (ICustomFormatter))
74             return this;
75         return null;
76     }
77
78     //实现Format方法说明:
79     //如果您的格式方法不支持格式, 则确定正在设置格式的对象是否实现 IFormattable 接口。
80     //如果实现, 请调用该接口的IFormattable.ToString 方法。
81     //否则, 调用基础对象的默认 Object.ToString 方法。
82     public string Format (string format, object arg, IFormatProvider provider)
83     {
84         if (format == null)
85         {
86             if (arg is IFormattable)
87                 return ((IFormattable)arg).ToString(format, provider);
88             return arg.ToString();
89         }
90         else
91         {
92             if (format=="MyBaseFormat")
93             {
94                 return "****"+arg.ToString();
95             }
96             else
97             {
98                 if (arg is IFormattable)
99                     return ((IFormattable)arg).ToString(format, provider);
100                 return arg.ToString();
101             }
102         }
103     }
104 }
105
106
107 到一控制台中
108     string printString=String.Empty;
109     int i=100;
110     MyBaseFormat myBaseFormat=new MyBaseFormat();
111
112     printString=string.Format(myBaseFormat,"显示正常格式:{0}",i);
113     Console.WriteLine(printString);
114     printString=string.Format(myBaseFormat,"显示正常格式:{0:C}",i);
115     Console.WriteLine(printString);
116     printString=string.Format(myBaseFormat,"显示自定义格式{0:MyBaseFormat}",i);
117     Console.WriteLine(printString);
118
119     Console.ReadLine();
120
121     -----
122
123 小总结:
124 1. 如果需要您自己的格式化包含在某个类上, 在该类上实现IFormattable接口
125 2. 如果希望自定义格式化并使它可供多个不同类使用, 那么实现 ICustomFormatter接口
126
127

```