

HashTable

一、HashTable使用

HashTable是一个键值对集合，键(key)对值(value)映射，每一个key对应一个value，key唯一不能重复，value可以重复，输入一个key，则返回他对应的value

1.赋值方法

```
//注：因为键额唯一性，因此在添加前要判断是否已存在同名键
//添加数据的两种方式：
//1. Add()方法      不能添加同名键
Hashtable ht = new Hashtable();
ht.Add(1, "张三");
ht.Add(2, true);
ht.Add(3, '男');
ht.Add(false, "错误的");
ht.Add('上', "上山");

//2. 直接赋值法      添加同名键时，新值(value)会覆盖老值(value)
ht["6"] = "新来的";
ht[2.5] = "我看看输出时能不能插进2和3中间";
ht[1] = "张三变张四"; //因为已有 1 这个键，所以新的值会覆盖掉原本的值
```

2.遍历方法

```
//在键值对集合中，是根据键去找值的
//键可以是各种类型，因此不适合用for循环进行遍历

//错误示例：
for (int i = 0; i < ht.Count; i++)
{
    System.Console.WriteLine(ht[i]);
}
//遍历结果为
//张三、True、男
//因为i永远取不到false和'上'，因此无法打印出"错误的"和"上山"
```

```
/**
 * 遍历集合时使用foreach(var item in collection){}
 * var : 用于声明弱类型变量，声明时必须给变量赋初值，程序根据这个初值自动判断变量的类
型
 * item : 要循环的集合中的每一项
 * collection : 要循环的集合
```

```

*/

//错误示例：遍历集合本身，输出的结果为集合的命名空间：
System.Collections.DictionaryEntry
foreach (var item in ht){
    System.Console.WriteLine(ht);
}

//正确示例： 遍历集合的键(key),输出结果为集合的值(value)
foreach(var item in ht.key){
    System.Console.WriteLine(ht[item]);
}

//输出测试：
foreach (var item in ht.Keys){
    System.Console.WriteLine("键是: " + item + "\t\t值是: " + ht[item]);
}
/*结果为：          注意输出结果与输入顺序不同
    键是： 6          值是： 新来的
    键是： 上          值是： 上山
    键是： 2.5          值是： 我看看输出时能不能插进2和3中间
    键是： 3            值是： 男
    键是： 2            值是： True
    键是： 1            值是： 将张三修改为张四
    键是： False        值是： 错误的
*/

```

```

//var的补充说明：
//定义var型时必须给变量赋初值，如：
var n1 = 3.14;
var n2 = 5000m;
var n3 = true;
var n4 = "男";
//但很多时候不能提前知道需要的类型，如接收用户输入时，因此不常用。

```

3.HashTable常用方法

1. contains方法判断是否存在

```

ht.Add(2.6,null);
//contains和containsKey相同，用来判断某个键是否存在
System.Console.WriteLine(ht.Contains(2.6));           //true
System.Console.WriteLine(ht.ContainsKey(2.6));         //true
//containsValue用来判断某个值是否存在
System.Console.WriteLine(ht.ContainsValue("新来的")); //true

```

2. 删除

```
ht.Clear();           //移除集合中所有元素(键值对)  
ht.Remove(2.5);       //Remove(object key),根据给定的键移除该元素(键值对)
```

二、HashTable练习