# C# 集合（Collection）

集合（Collection）类是专门用于数据存储和检索的类。这些类提供了对栈（stack）、队列（queue）、列表（list）和哈希表（hash table）的支持。大多数集合类实现了相同的接口。

**各种集合类和它们的用法**

下面是各种常用的 **System.Collection** 命名空间的类。点击下面的链接查看细节。

|  |  |
| --- | --- |
| **类** | **描述和用法** |
| [动态数组（ArrayList）](http://www.runoob.com/csharp/csharp-arraylist.html" \o "C# ArrayList) | 它代表了可被单独**索引**的对象的有序集合。  它基本上可以替代一个数组。但是，与数组不同的是，您可以使用**索引**在指定的位置添加和移除项目，动态数组会自动重新调整它的大小。它也允许在列表中进行动态内存分配、增加、搜索、排序各项。 |
| [哈希表（Hashtable）](http://www.runoob.com/csharp/csharp-hashtable.html" \o "C# Hashtable) | 它使用**键**来访问集合中的元素。  当您使用键访问元素时，则使用哈希表，而且您可以识别一个有用的键值。哈希表中的每一项都有一个**键/值**对。键用于访问集合中的项目。 |
| [排序列表（SortedList）](http://www.runoob.com/csharp/csharp-sortedlist.html" \o "C# SortedList) | 它可以使用**键**和**索引**来访问列表中的项。  排序列表是数组和哈希表的组合。它包含一个可使用键或索引访问各项的列表。如果您使用索引访问各项，则它是一个动态数组（ArrayList），如果您使用键访问各项，则它是一个哈希表（Hashtable）。集合中的各项总是按键值排序。 |
| [堆栈（Stack）](http://www.runoob.com/csharp/csharp-stack.html" \o "C# Stack) | 它代表了一个**后进先出**的对象集合。  当您需要对各项进行后进先出的访问时，则使用堆栈。当您在列表中添加一项，称为**推入**元素，当您从列表中移除一项时，称为**弹出**元素。 |
| [队列（Queue）](http://www.runoob.com/csharp/csharp-queue.html" \o "C# Queue) | 它代表了一个**先进先出**的对象集合。  当您需要对各项进行先进先出的访问时，则使用队列。当您在列表中添加一项，称为**入队**，当您从列表中移除一项时，称为**出队**。 |
| [点阵列（BitArray）](http://www.runoob.com/csharp/csharp-bitarray.html" \o "C# BitArray) | 它代表了一个使用值 1 和 0 来表示的**二进制**数组。  当您需要存储位，但是事先不知道位数时，则使用点阵列。您可以使用**整型索引**从点阵列集合中访问各项，索引从零开始。 |

**Hashtable哈希表**

用于处理key/value的键值对，其中key通常可用来快速查找，Hashtable中key/value键值对均为object类型，所以Hashtable可以支持任何类型的key/value键值对.

哈希表的简单操作

添加一个key/value键值对：HashtableObject.Add(key,value);

去除某个key/value键值对：HashtableObject.Remove(key);

移除所有元素：HashtableObject.Clear();

是否包含特定键key：HashtableObject.Contains(key);

遍历哈希表

for(DictionaryEntry de in ht) //ht为一个Hashtable实例

{

Console.WriteLine(de.Key);//de.Key对应于key/value键值对key

Console.WriteLine(de.Value);//de.Key对应于key/value键值对value

}

对哈希表进行排序

我们无法直接在Hashtable进行对key进行重新排列，如果需要Hashtable提供某种规则的输出，可以采用一种变通的做法：

ArrayList akeys=new ArrayList(ht.Keys);

akeys.Sort(); //按字母顺序进行排序

for(string skey in akeys)

{

Console.Write(skey + ":");

Console.WriteLine(ht[skey]);//排序后输出

}