

学习 Excel VBA 数据结构 • 集合详解

完美Excel出品
微信公众号: [excelperfect](#)



内容提要

集合是一组元素的集，可用来存储对象和数据。VBA 提供了集合对象，是标准 VBA 库的一部分。

本电子书《[学习 Excel VBA 数据结构: 集合详解](#)》详细讲解了集合的概念，在 Excel VBA 中如何创建和使用集合的基本知识。



目录

主要内容.....	I
1. 什么是集合.....	1
Excel 对象模型中的集合.....	2
2. 如何创建集合.....	5
集合的作用域.....	6
集合名称命名约定.....	7
3. 基本操作— 在集合中添加元素.....	9
示例 1：添加元素.....	10
示例 2：添加元素并指定键.....	10
示例 3：使用数字作为键.....	11
示例 4：创建唯一元素列表.....	11
示例 5：将元素添加到指定位置.....	12
4. 基本操作——搜索并获取集合中的元素.....	15
示例 1：使用索引值或者键值获取集合中的元素.....	16
示例 2：创建集合和子集合并显示元素值.....	17
示例 3：判断元素在集合中是否存在.....	19
5. 基本操作—— 删除集合中的元素.....	21
示例 1：通过索引值或键值删除指定的元素.....	21



示例 2：删除指定内容的元素	22
示例 3：删除集合中的所有元素	23
6. 示例——对集合中的元素排序	25
关于完美 Excel	31
完美 Excel 微信公众号使用指南	32



主要内容

集合是 Excel VBA 编程中很重要的一种数据结构。我们在编程时，会不自觉地使用到集合，例如 Workbooks 集合、Worksheets 集合，等等。我们也可以创建自己的集合，将数据存储在集合中，方便各种操作。

本书详细讲解了集合的概念，以及在 Excel VBA 中如何创建和使用集合的基本知识。

1.什么是集合

集合的概念，集合的基本属性和方法，Excel 对象模型中的集合。

2.如何创建集合

VBA 中创建集合的两种方法，集合的作用域，集合的命名约定。

3.基本操作——在集合中添加元素

详解在集合中添加元素的 **Add 方法**，以及获取集合中元素数量的 **Count 属性**。文中的示例：①添加元素；②添加元素并指定键；③使用数字作为键；④创建唯一元素列表；⑤将元素添加到指定位置。

4.基本操作——搜索并获取集合中的元素

详解使用 **Item 方法** 获取集合中的元素。文中的示例：①使用索引值或者键值获取集合中的元素；②创建集合和子集合并显示元素值；③判断元素在集合中是否存在。



5.基本操作——删除集合中的元素

详解使用 `Remove` 方法从集合中删除元素：①通过索引值或键值删除指定的元素；②删除指定内容的元素；③删除集合中的所有元素。

6.示例——对集合中的元素排序

演示排序集合中的元素的代码。

温馨提示：

在完美 Excel 微信公众号中发送消息：**VBA 集合**，即可获得本电子书文档下载链接和密码。



如果您对本书有什么建议或者还有什么好的示例，欢迎前往完美 Excel 微信公众号沟通交流。

欢迎分享本电子书，让更多的人方便地得到所需要的知识。



1. 什么是集合

集合是一组元素的集，可用来存储对象和数据。VBA 提供了集合对象，是标准 VBA 库的一部分。

在创建集合对象后，可以在集合中存储任意类型的数据，可以在同一集合中存储不同类型的数据。因此，集合中的元素可以是字符串、数字、日期、数组、对象、类的实例，等等，甚至可以嵌套另一个集合，可以是每种数据类型的组合，并且可以使用索引值或者唯一的键来检索集合中的元素。

集合对象有 4 种基本属性/方法：

- Add
- Count
- Item
- Remove

使用这些方法/属性可以实现对集合的操作，包括写入、读取、修改、替换、删除，等等。

其中，**Add 方法**在集合中添加元素；**Count 属性**返回集合中元素的数量；**Item 方法**通过参数指定的索引值或键从集合中返回对象，集合中的第 1 个元素存储在第 1 个位置，即序数为 1；**Remove 方法**删除集合中的元素。

Excel 在内存中存储集合，因此可以非常快速地对集合进行操作。



Excel 对象模型中的集合

通常，在 Excel 对象模型中，大多数对象都有相关联的集合对象，很多情形下对象的复数形式表示该类相关对象的集合。例如，Workbooks 集合是当前打开的所有 Workbook 对象的集合，Worksheets 集合是某 Workbook 对象中所有 Worksheet 对象的集合。

可以处理对象所属的整个集合或者对象集合中的某个对象。如果在集合名称后面的括号中放置该对象的名称或索引号，则可以引用该对象，例如：

```
Worksheets("MySheet")
```

引用名为 MySheet 的工作表。

如果工作表 MySheet 是集合中的第 2 个工作表，可以使用下面的语句来引用：

```
Worksheets(2)
```

在 Excel 对象模型的层次结构中，上层的对象会包含下层对象的集合，如下图 1.1 所示。



图 1.1

可以看出，集合本身也是对象，该对象包含其他紧密相关的对象。

除上文中集合的基本属性/方法外，对象模型中的集合还具有下面的通用属性：

- Application
- Creator



- Parent

其中，`Application` 属性返回当前对象所在的应用程序，`Creator` 属性返回一个描述是否在应用程序中创建了该对象的长整型数值，`Parent` 属性返回该集合对象的上层对象。





本章内容 2018 年 4 月 10 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[VBA 进阶 | 集合详解 2: 如何创建集合](#)

2.如何创建集合

VBA 提供了创建集合的两种方法。

第一种方法：在集合声明时使用 **NEW** 关键字

语法为：

```
Dim 变量名称 As New Collection
```

说明：

- Dim 语句中的 New 关键字隐含着使用 Set 语句来实例化 Collection 对象。

下面的代码创建名为 colMyCollection 的集合对象，并添加集合元素。

```
Dim colMyCollection As New Collection  
colMyCollection.Add Item:="完美 Excel",  
Key:="excelperfect"
```

第二种方法：声明后使用 **Set** 语句

语法为：

```
Dim 变量名称 As Collection  
Set 变量名称 = New Collection
```

说明：

- 使用 Set 语句来实例化集合。
- 在需要时创建集合，从而节省内存空间。



下面的代码实现与第一方法同样的效果：

```
Dim colMyCollection As Collection
Set colMyCollection = New Collection
colMyCollection.Add Item:="完美 Excel",
Key:="excelperfect"
```

下面的代码在变量 str 的值为“完美 Excel”时才创建集合对象并添加相应的元素。

```
Dim colMyCollection As Collection
Dim str As String

If str = "完美 Excel" Then
    Set colMyCollection = New Collection
    colMyCollection.Add Item:="完美 Excel",
Key:="excelperfect"
Else
    Exit Sub
End If
```

此外，还可以使用 **With ... End With** 创建集合，无须事先声明变量。

```
With New Collection
    .Add "完美 Excel"
End With
```

这个集合只在 With ... End With 块内起作用。

集合的作用域

声明集合变量的位置不同，集合的作用域也不同。

- 如果在过程的内部声明集合变量，那么只能在该过程内部使用这个集合。
- 如果在代码模块的声明区域中声明集合变量，那么可以在该代码模块的所有



过程中使用这个集合。

- 如果在代码模块的声明区域中使用 `Public` 声明集合变量，那么可以在所有代码模块的所有过程中使用这个集合。

集合名称命名约定

对于集合变量使用一致的命名约定有助于代码的理解，可以在变量名中使用 `col` 开头。

避免使用 VBA 保留字或预定义的名称。





3. 基本操作— 在集合中添加元素

使用 Add 方法在集合中添加元素。其语法为:

```
集合对象.Add Item, [Key], [Before], [After]
```

说明:

- 参数 Item, 必需, 可以是任何数据类型, 用来指定加入到集合中的元素。
- 参数 Key, 可选, String 类型, 指定集合元素的键, 为唯一的字符串。如果忽略该参数, 则不能使用参数 Key 来访问集合中的元素。
- 参数 Before, 可选, String 类型或 Numeric 类型, 指定添加的元素在集合中的相对位置。添加的元素移动到集合中则参数 Before 指定的元素之前。
- 参数 After, 可选, String 类型或 Numeric 类型, 指定添加的元素在集合中的相对位置。添加的元素移动到集合中参数 After 指定的元素之后。
- 不能同时指定参数 Before 或 After 的值。如果不指定参数 Before 或 After 的值, 那么元素将添加到集合的最后面。无论参数 Before 或 After 是字符串表达式还是数字表达式, 其值必须是集合中已经存在的元素所在的位置, 否则将导致错误。如果是字符串值, 则必须存在键。如果是数字值, 则该值必须位于 1 和所有元素的总数值之间。
- 按添加的顺序, 自动添加索引值。



使用 Count 属性可以获取集合中的元素数量。其语法为：

`集合对象.Count`

说明：

- 集合以 1 为基数，集合的第 1 个元素的索引值是 1。
- 对于空集合，.Count 返回 0。
- 可以在循环中使用从 1 到.Count 的值来依次获取集合中的元素。

示例 1：添加元素

代码如下：

```
Dim colMyCollection As New Collection

colMyCollection.Add "完美 Excel"
colMyCollection.Add 1001
colMyCollection.Add Array("excelperfect", "Excel")
```

示例 2：添加元素并指定键

代码如下：

```
Dim colMyCollection As New Collection

colMyCollection.Add "完美 Excel", "Key1"
colMyCollection.Add 1001, "Key2"
colMyCollection.Add Array("excelperfect", "Excel"), "Key3"
```



正如前面已提到的，键必须是唯一字符串。如果在添加元素时使用集合中已存在的键，将会产生错误。

在添加元素时，可以将索引值链接到键。代码如下：

```
Dim colMyCollection As New Collection

colMyCollection.Add "完美 Excel", "Key" & colMyCollection.Count
colMyCollection.Add 1001, "Key" & colMyCollection.Count
colMyCollection.Add Array("excelperfect", "Excel"), "Key" & colMyCollection.Count
```

示例 3：使用数字作为键

由于键不能包含数字，因此可以将数字转换成字符串来作为键。代码如下：

```
Dim colMyCollection As New Collection
Dim i As Long

For i = 1 To 10
    colMyCollection.Add i, CStr(i)
Next i
```

示例 4：创建唯一元素列表

使用要添加的元素的内容作为键，创建唯一元素列表。代码如下：

```
Sub CreateUniqueValue()
    Dim sn, col, i, str
    Dim colMyCollection As New Collection
```



```
On Error Resume Next

    sn = Array("aa", "bb", "cc", "dd", "aa", "bb", "cc", "dd")

    For Each col In sn
        colMyCollection.Add col, col
    Next col

    For i = 1 To colMyCollection.Count
        str = str & colMyCollection(i) & vbCrLf
    Next i

    MsgBox str
End Sub
```

运行后的结果如图 3.1 所示。

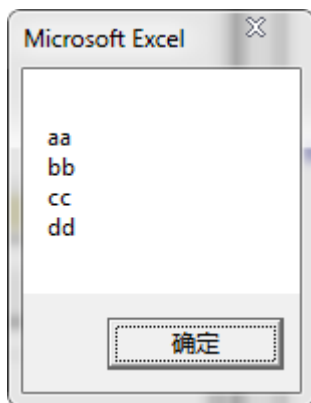


图 3.1

示例 5：将元素添加到指定位置

上文已经提到，集合中的每个元素都有唯一的索引值，添加到集合中的元素顺序就确定了索引值，第 1 个元素的索引值是 1，最后 1 个索引值等于.Count 属性的结果。



使用参数 Before 或 After 可以指定元素添加的位置。如下面所示的代码：

```
Sub AddItemToSpecifyPos()  
    With New Collection  
        .Add 111  
        .Add 222  
        .Add 333, , 2  
        .Add 444, , , 1  
  
        MsgBox .Item(1) & vbCrLf & .Item(2) & vbCrLf & _  
                .Item(3) & vbCrLf & .Item(4)  
    End With  
End Sub
```

运行后的结果如图 3.2 所示。



图 3.2

当然，也可以使用键作为参数 Before 或 After 的值，将数据添加到指定的键之前或之后。





本章内容 2018 年 4 月 26 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[VBA 进阶 | 集合详解 4: 基本操作——搜索并获取集合中的元素](#)

4. 基本操作——搜索并获取集合中的元素

在集合中，每个元素都具有唯一的索引值和对应的元素内容。并且，集合中的元素可以对应唯一的键。

可以使用 Item 方法来获取集合中的元素。其语法为：

```
集合对象.Item(Index)
```

说明：

- 参数 Index，必需，为 Numeric 类型或 String 类型。如果是数字，则该参数为索引值，返回该索引值对应的元素；如果是字符串，则该参数是键，返回键所在的元素。
- 指定的字符串必须是集合中已经存在的键，同样指定的索引值必须处于 1 和集合元素总数之间的数字，否则会产生错误。
- Item 方法是集合对象的默认方法，也就是说，在搜索并获取集合元素时，不必显式调用 Item 方法。语句 colMyCollection.Item(2) 与 colMyCollection(2) 相同。



示例 1：使用索引值或者键值获取集合中的元素

下面的代码使用索引值或者键值来获取集合中指定的元素：

```
Sub GetColValue()  
    Dim colMyCollection As New Collection  
  
    colMyCollection.Add "完美 Excel", "excelperfect"  
    colMyCollection.Add "Excel VBA 宝典", "ExcelVBA"  
    colMyCollection.Add "Office 大全", "Office"  
  
    MsgBox "使用索引值或键从集合中获取数据:" & vbCrLf & _  
        colMyCollection.Item(2) & vbCrLf & _  
        colMyCollection(2) & vbCrLf & _  
        colMyCollection.Item("excelperfect") & vbCrLf & _  
        colMyCollection("excelperfect")  
End Sub
```

运行代码的结果如图 4.1 所示。

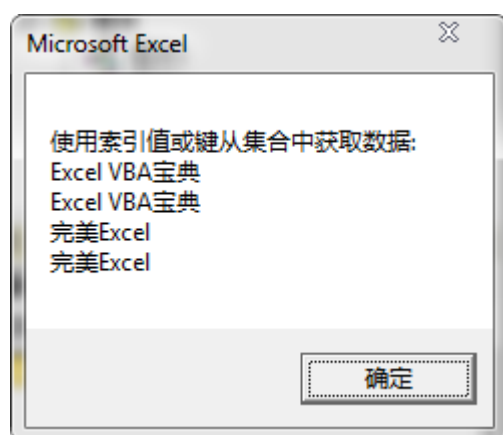


图 4.1

注意，由于不可能获取集合中的所有键，因此只有在知道集合中存在有哪些键时才能更好地利用键值来获取数据。



示例 2：创建集合和子集合并显示元素值

下面的代码演示如何在一个集合中嵌入另一个集合。代码开始创建了子集合 colSubCollection 的 10 个实例，每个实例添加了 2 个元素，然后获取并显示子集合中的值。

```
Sub testCollection()  
    Dim colMainCollection As New Collection  
    Dim colSubCollection As Collection  
    Dim i As Long  
    Dim strEN As String, strCN As String  
  
    For i = 1 To 10  
        Set colSubCollection = New Collection  
        colSubCollection.Add Item:="excelperfect" & i, _  
            Key:="英文版完美 Excel"  
        colSubCollection.Add Item:="完美 Excel" & i, _  
            Key:="中文版完美 Excel"  
        colMainCollection.Add Item:=colSubCollection, _  
            Key=CStr(i)  
        Set colSubCollection = Nothing  
    Next i  
  
    MsgBox "主集合的元素数: " & colMainCollection.Count  
  
    For i = 1 To colMainCollection.Count  
        Set colSubCollection =  
colMainCollection.Item(CStr(i))  
        strEN = strEN & "英文版完美 Excel" & i & "=" & _  
            colSubCollection.Item("英文版完美 Excel")  
        strCN = strCN & "中文版完美 Excel" & i & "=" & _  
            colSubCollection.Item("中文版完美 Excel")  
    Next i  
End Sub
```



```
Set colSubCollection = Nothing  
strEN = strEN & vbCrLf  
strCN = strCN & vbCrLf  
Next i  
  
MsgBox strEN & vbCrLf & vbCrLf & strCN  
End Sub
```

运行代码后的结果如图 4.2 所示。

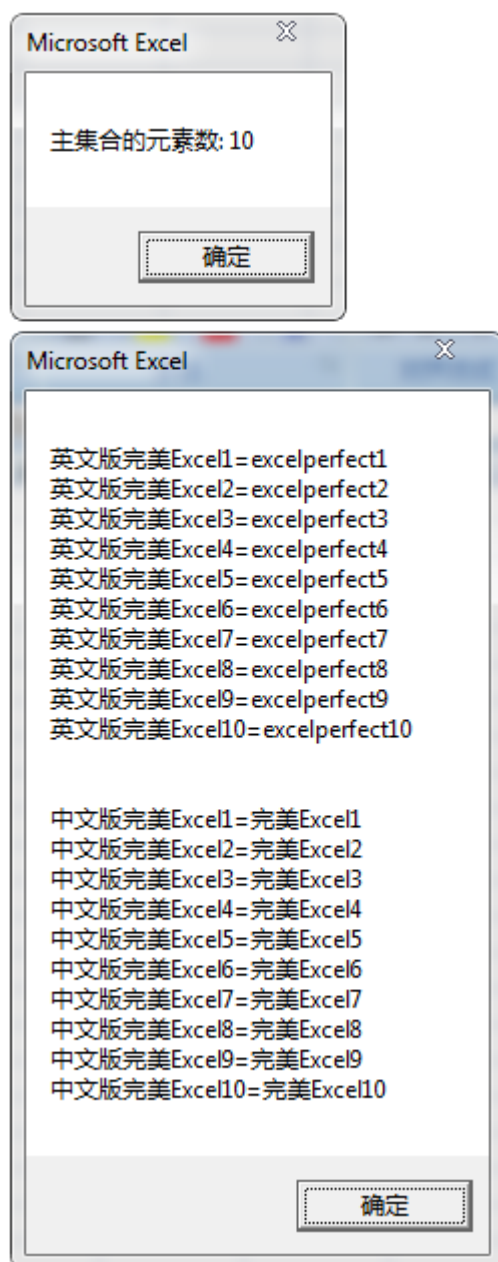


图 4.2



也可以使用 For Each ... Next 循环来遍历并获取集合中的元素，示例代码如下：

```
Sub testCollection1()  
    Dim colMainCollection As New Collection  
    Dim colSubCollection As Collection  
    Dim i As Long  
    Dim strEN As String, strCN As String  
  
    For i = 1 To 10  
        Set colSubCollection = New Collection  
        colSubCollection.Add Item:="excelperfect" & i, _  
            Key:="英文版完美 Excel"  
        colSubCollection.Add Item:="完美 Excel" & i, _  
            Key:="中文版完美 Excel"  
        colMainCollection.Add Item:=colSubCollection, _  
            Key:=CStr(i)  
        Set colSubCollection = Nothing  
    Next i  
  
    For Each colSubCollection In colMainCollection  
        MsgBox colSubCollection.Item("英文版完美 Excel") & _  
            vbCrLf & colSubCollection.Item("中文版完美 Excel")  
    Next colSubCollection  
  
End Sub
```

示例 3：判断元素在集合中是否存在

下面的代码在集合中包含指定键的元素时返回 True。

```
Dim mColMyCollection As Collection
```



```

Function ColIfExists(varKey As Variant) As Boolean
    Dim sValue

    On Error Resume Next

    sValue = mColMyCollection.Item(varKey)

    If Err.Number = 0 Then
        ColIfExists = True
    Else
        ColIfExists = False
    End If
End Function

```

可以通过遍历集合来判断元素是否存在，示例代码如下：

```

Sub ExistSomeValue()
    Dim colMyCollection As New Collection
    Dim i As Integer

    colMyCollection.Add "完美 Excel", "excelperfect"
    colMyCollection.Add "Excel VBA 宝典", "ExcelVBA"
    colMyCollection.Add "Office 大全", "Office"

    For i = 1 To colMyCollection.Count
        If colMyCollection(i) = "完美 Excel" Then
            MsgBox "集合中存在 " & colMyCollection(i)
        End If
    Next i
End Sub

```



本章内容 2018 年 5 月 4 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
VBA 进阶 | 集合详解 5: 基本操作——删除集合中的元素

5. 基本操作—— 删除集合中的元素

使用 Remove 方法从集合中删除元素。其语法为：

```
集合对象.Remove Index
```

说明：

- 参数 Index，必需，为 Numeric 类型或 String 类型。如果是数字，则该参数为索引值；如果是字符串，则该参数是键。
- 如果将参数 Index 为字符串值，则删除该字符串值代表的键相对应的元素；
如果参数 Index 为数字，则该数索引处的元素被删除。
- 如果集合中不存在指定的字符串表示的键，或者指定的数字不在 1 与集合元素总数之间，那么会产生错误。
- 删除元素后，在被删除元素后面的元素将自动上移一个位置。因此，集合中并不会留下空隙。
- 如果使用索引值删除集合中的多个元素，必须从后向前删除，即从最高索引值向最低索引值，因为每次删除后集合会重新进行索引。

示例 1：通过索引值或键值删除指定的元素

下面的代码使用索引值或者键来从集合中删除元素。



```

Sub DeleteSomeValue()
    Dim colMyCollection As New Collection

    colMyCollection.Add "完美 Excel", "excelperfect"
    colMyCollection.Add "Excel VBA 宝典", "ExcelVBA"
    colMyCollection.Add "Office 大全", "Office"

    With colMyCollection
        .Remove 2
        .Remove "Office"
    End With
End Sub

```

示例 2：删除指定内容的元素

下面的代码删除集合中内容为“完美 Excel”的元素。

```

Sub DeleteSomeValueByContent()
    Dim colMyCollection As New Collection
    Dim i As Integer

    colMyCollection.Add "完美 Excel", "excelperfect"
    colMyCollection.Add "Excel VBA 宝典", "ExcelVBA"
    colMyCollection.Add "Office 大全", "Office"

    With colMyCollection
        For i = .Count To 1 Step -1
            If InStr(.Item(i), "完美 Excel") Then
                .Remove i
            End If
        Next i
    End With
End Sub

```



```
        Next i
    End With
End Sub
```

示例 3：删除集合中的所有元素

可以采取从开始到结尾或者从结尾到开始的顺序删除集合中的所有元素。

```
Dim mColMyCollection As Collection

Sub Clear()
    Dim i As Long

    For i = 1 To mColMyCollection.Count
        mColMyCollection.Remove i
    Next i
End Sub
```

或者：

```
Dim mColMyCollection As Collection

Sub Clear()
    Dim i As Long

    For i = mColMyCollection.Count To 1 Step -1
        mColMyCollection.Remove i
    Next i
End Sub
```



还可以将集合再次进行实例化来模拟“清除”集合中的元素：

```
Set colMyCollection = New Collection
```



本章内容 2018 年 5 月 12 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[VBA 进阶 | 集合详解 6: 示例——对集合中的元素排序](#)

6. 示例——对集合中的元素排序

用 Add 方法添加元素时利用参数 Before 或 After 来指定集合中元素的顺序，也可以使用一些排序算法或技巧来实现对集合元素的排序。

下面的示例代码演示了如何排序集合中的元素。

```
Sub SortCollectionSample()  
    Dim colFruit As Collection  
    Dim varItem As Variant  
    Dim varTemp As Variant  
    Dim i As Long, j As Long  
  
    Set colFruit = New Collection  
  
    '添加集合元素  
    colFruit.Add "Mango", "Mango"  
    colFruit.Add "Apple", "Apple"  
    colFruit.Add "Peach", "Peach"  
    colFruit.Add "Kiwi", "Kiwi"  
    colFruit.Add "Lemon", "Lemon"  
  
    '冒泡法排序  
    For i = 1 To colFruit.Count - 1  
        For j = i + 1 To colFruit.Count  
            If colFruit(i) > colFruit(j) Then  
                varTemp = colFruit(j)
```



```

        colFruit.Remove j
        colFruit.Add varTemp, varTemp, i
    End If
Next j
Next i

'显示结果
For Each varItem In colFruit
    Debug.Print varItem
Next varItem
End Sub

```

在代码中，在创建并添加集合元素时，也添加了键。

下面的代码更通用一些，适合于含有键和不含有键的集合排序。

```

Sub SortCollectionSamplePlus(colData As Collection,
Optional Keys = False)
    Dim varItem As Variant
    Dim varTemp As Variant
    Dim i As Long, j As Long

    If Keys = True Then
        '冒泡法排序
        For i = 1 To colData.Count - 1
            For j = i + 1 To colData.Count
                If colData(i) > colData(j) Then
                    varTemp = colData(j)
                    colData.Remove j
                    colData.Add varTemp, varTemp, i
                End If
            Next j
        Next i
    End If
End Sub

```




```

Else
    For i = 1 To colData.Count - 1
        For j = i + 1 To colData.Count
            If colData(i) > colData(j) Then
                varTemp = colData(j)
                colData.Remove j
                colData.Add varTemp, , i
            End If
        Next j
    Next i
End If

'显示结果
For Each varItem In colData
    Debug.Print varItem
Next varItem
End Sub

```

下面的示例先将集合中的元素输入到当前工作表中，然后利用工作表的 Sort 方法对元素进行排序，最后将排序好的元素重新添加到集合中。

```

Sub SortCollectionSampleByWorksheet()
    Dim colFruit As Collection
    Dim varItem As Variant
    Dim varTemp As Variant
    Dim i As Long, j As Long
    Dim lngLast As Long
    Dim rng As Range

    Set colFruit = New Collection

    '添加集合元素
    colFruit.Add "Mango"

```



```

colFruit.Add "Apple"
colFruit.Add "Peach"
colFruit.Add "Kiwi"
colFruit.Add "Lemon"

'在当前工作表中输入集合元素
For i = 1 To colFruit.Count
    Range("A" & i) = colFruit(i)
Next i

lngLast = Range("A" & Rows.Count).End(xlUp).Row

'利用工作表的 Sort 方法排序
With ActiveSheet.Sort
    .SetRange Range("A1:A" & lngLast)
    .Header = xlNo
    .Apply
End With

'清空原集合
Set colFruit = New Collection

'重新添加集合元素
For Each rng In Range("A1:A" & lngLast)
    colFruit.Add rng.Value
Next rng

'显示结果
For Each varItem In colFruit
    Debug.Print varItem
Next varItem
End Sub

```



由于集合没有移动方法，也没有获取键的方法，因此上面的示例介绍的排序方法只适用于没有键的集合。





关于完美 Excel

完美 Excel 是一个坚持分享 Excel 与 VBA 技术知识的微信公众号，自 2017 年 5 月 15 日开始，每天推送一篇 Excel 与 VBA 技术和应用方面的文章。目前，共有 670 余篇实用文章可供广大 Excel 爱好者和使用者学习交流。这本电子书就是根据完美 Excel 上发表的关于集合基础知识的系列文章整理而成的。

每天清晨，完美 Excel 微信公众号：**excelperfect** 都会推送一篇关于 Excel 与 VBA 的相关文章。如果你有兴趣学习 Excel 和 VBA 的相关知识和实战技巧，可以关注完美 Excel 微信公众号，绝对不会让你失望！

可以通过下列方法关注[完美 Excel]微信公众号：

方法 1—在通讯录中搜索“完美 Excel”或者“**excelperfect**”后点击关注。

方法 2—扫一扫下面的二维码



完美 Excel 微信公众号使用指南

下图 1 是完美 Excel 微信公众号的界面。公众号名称显示在屏幕正上方，屏幕底部显示有“菜单栏”，目前设置的菜单为“技术精粹”、“VBA 精选”、“联系 me”。在底部左侧的小键盘图标为消息框入口，单击可进入消息框界面给完美 Excel 公众号发送消息。



图 1

下图 2、图 3、图 4 分别为底部 3 个菜单的子菜单。目前，菜单“技术精粹”中设置有“2018 年文章合集”、“480 篇文章合集”、“又一波 20 个函数”、“快速学会 30 个函数”、“玩转数据验证”等 5 个子菜单；菜单“VBA 精选”中设置有“Excel VBA 编程基础”、“最最基础入门篇”、“VBA 学习经验”等 3 个子菜单；菜单“联系 me”中设置有“投稿须知”、“网站集粹”、“个人声明”、“坚持的美好”、“2019 年目标”等 5 个子菜单。





图 2



图 3





图 4

单击这些子菜单会进入详细的文章页面或者文章整理的入口页面，方便读者浏览或查阅本公众号的文章。同时，这些子菜单会随着完美 Excel 微信公众号内容的增加而适时调整。

可以单击底部左侧的小键盘图标，进入发送消息界面，如图 5 所示。在文本框中输入想要发送的文字，单击底部的“发送”按钮，就可以将消息发送给完美 Excel 微信公众号。

大家应留意完美 Excel 微信公众号推送的文章中的一些信息，例如，我会在百度网盘中存放一些文档资料或者示例工作簿文件，并在文章中给出进入百度网盘下载的文本信息，你只需在发送消息框中输入我给出的文本，单击发送后，就会收到一条关于下载链接和密码的信息。单击链接并按提示输入密码后，即可获得



相关的文档资料或示例工作簿文件了。



图 5

例如，在图 5 所示的界面中输入“Excel 动画图 2”后，会自动收到图 6 所示的信息，根据信息即可获取这个 Excel 动画图表文件。



图 6



希望大家在完美 Excel 微信公众号中能够学习到所需要的知识，获取到所需要的 Excel 应用技巧，提高自己的水平。但愿在今后的日子里，完美 Excel 微信公众号能够真正帮助大家发挥 Excel 的威力，为大家解决问题，提高工作效率。

