

► vJass 系列教程 11

面向对象（六）数组对象

Aeris ▶ NJU ▶ 2009/2/4

vJass 系列教程 11

面向对象（六）数组对象

这是最后一篇了，如果你能够看到这里，那么你已经可以称得上是 vJass 高手了。本篇向大家介绍 vJass 剩下的几个很吸引人的特征，虽然它们使用的频率不是很高。

超大数组

找不到什么合适的词，只能简单称之为“超大数组”了。超大数组解决的是什么问题呢？很简单，我们都知道，Jass 的数组里只能存放 8192 个元素，假如我要更多的，比如说要 10000 个，怎么办呢？vJass 提供了一个特性，使得我们可以声明一个可以具有超过 8192 个元素的数组。很简单，语法如下：

这是正常的数组声明语法（以整数数组为例，其他类型的数组一样）：

```
integer array iar
```

这样就声明了一个容量是 8192 个元素的“正常”数组。如果我要 10000 个元素呢？这样：

```
integer array iar [10000]
```

在数组声明最后加上一对中括号，里面写明容量就 OK 了。

其他类型的数组也一样。关于用法也没什么好多说的，唯一要提醒的是最好不要声明超过一百万个元素的数组，否则你可能会发现 war3map.j 的大小暴涨。

动态数组

vJass 的各种语法是具有很强的针对性的，许多语法都是为了解决 Jass 的缺陷和不足而设置。那么动态数组又是解决的什么问题呢？

我们知道，函数是不能接受数组作为参数的，也不能返回数组，这是 Jass 的一大限制。现在 vJass 的动态数组，打破了 Jass 数组不能作为参数和返回值的限制。而且，动态数组的用法也很简单。

声明一个动态数组类型（注意是数组类型，不是数组）

```
type 数组类型名 extends integer array [数组容量]
```

例如

```
type iar extends integer array [3]
```

声明了一个数组类型叫 iar，每个这种类型的数组有 3 个元素。

注意：数组的容量不能超过数组最大容量的四分之一（ $8192/4=2048$ ）。因为它的实现是使用一个 8192 的全局数组，根据动态数组容量划分成若干个“子数组”。如果动态数组的容量是 1000，那么最多只能分配 $8192/1000=8$ 个数组。如果容量是 100,那么最多分配 80 个。想要更大的容量，请看动态超大数组。

现在有了这个动态数组类型，我们就可以分配动态数组对象了。分配动态数组对象和分配结构对象一样，都是使用 create 方法，例如：

```
local iar a = iar.create()
```

这样就给动态数组 a 分配了一个对象，以后就可以像使用一般数组一样操作 a 了，例如：

```
set a[0] = 1
set a[1] = 2
set a[2] = 3
```

用完了动态数组，必须像一般对象一样使用 destroy 来销毁，否则会有内存泄漏，就像这样：

```
call a.destroy()
```

动态数组最方便的地方是可以作为函数的参数和返回值。例如我们可以写一个函数来给 iar 数组赋值：

```
function SetVal takes iar a returns nothing
    set a[0] = 1
    set a[1] = 2
    set a[2] = 3
endfunction
```

可以看到，动态数组作为函数的参数传入到函数中了。

动态超大数组

前面提到，动态数组的实现是利用一个正常的 8192 大小的数组，分割成若干个小数组的方式来实现的，vJass 编译器限定数组的容量最高只能是总容量的四分之一，所以动态数组的容量最高是 2048。如果需要更大的容量（真的需要这么大么？！），那么可以使用动态超大数组。动态超大数组和普通动态数组唯一的区别就是声明上，这是普通动态数组的声明：

```
type 数组类型名 extends integer array [数组容量]
```

这是动态超大数组的声明，只有最后不同：

```
type 数组类型名 extends integer array [数组容量, 总容量]
```

例如：

```
type bigiar extends integer array [10000, 200000]
```

bigiar 类型的动态数组会具有 10000 个元素，总容量是 200000，也就是说，最多可以创建 $20(200000/10000=20)$ 个 bigiar 类型的动态超大数组。

和超大数组的注意点一样，不要声明过分离谱的大小，否则 war3map.j 的大小会很引人注目。