

# 数组

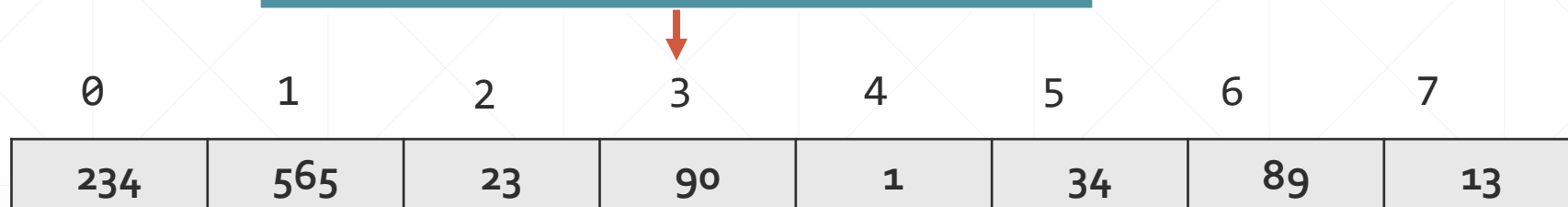
---

北京理工大学计算机学院  
金旭亮

# 数组的定义

我们把一组有顺序的数据所构成的整体，称为“**数组**”。

数组中数据的位置编号，从0开始依次递增，称为“索引（index）”



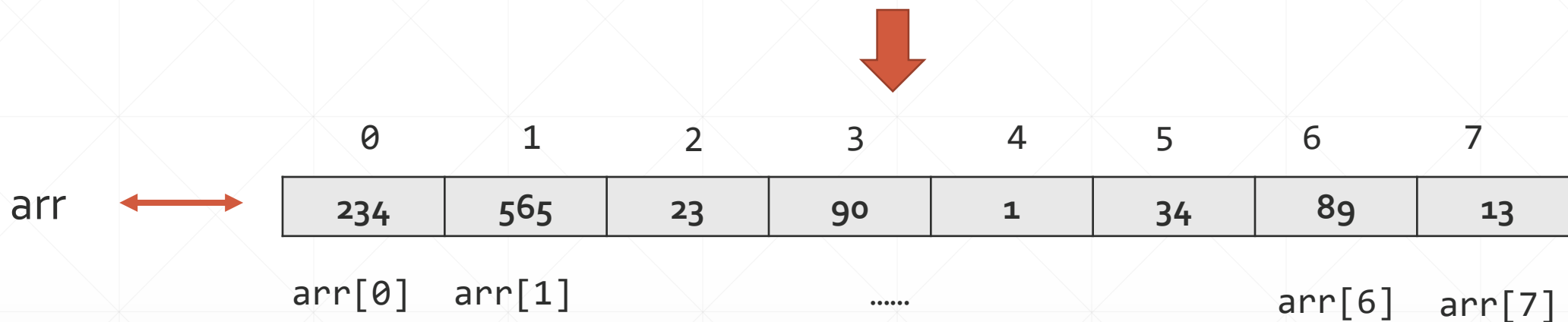
0	1	2	3	4	5	6	7
234	565	23	90	1	34	89	13

数组中的数据

数组中的单个数据，称为“**元素**”，它们具有相同的数据类型。

# 数组的特点

```
int[] arr = new int[8] { 234, 565, 23, 90, 1, 34, 89, 13 };
```



数组一旦创建之后，尺寸保持不变，元素在内存中连续分布。

数组是一个对象，数组变量（`arr`）引用这个数组对象。

通过“数组名[索引值]”访问单个的数组元素。

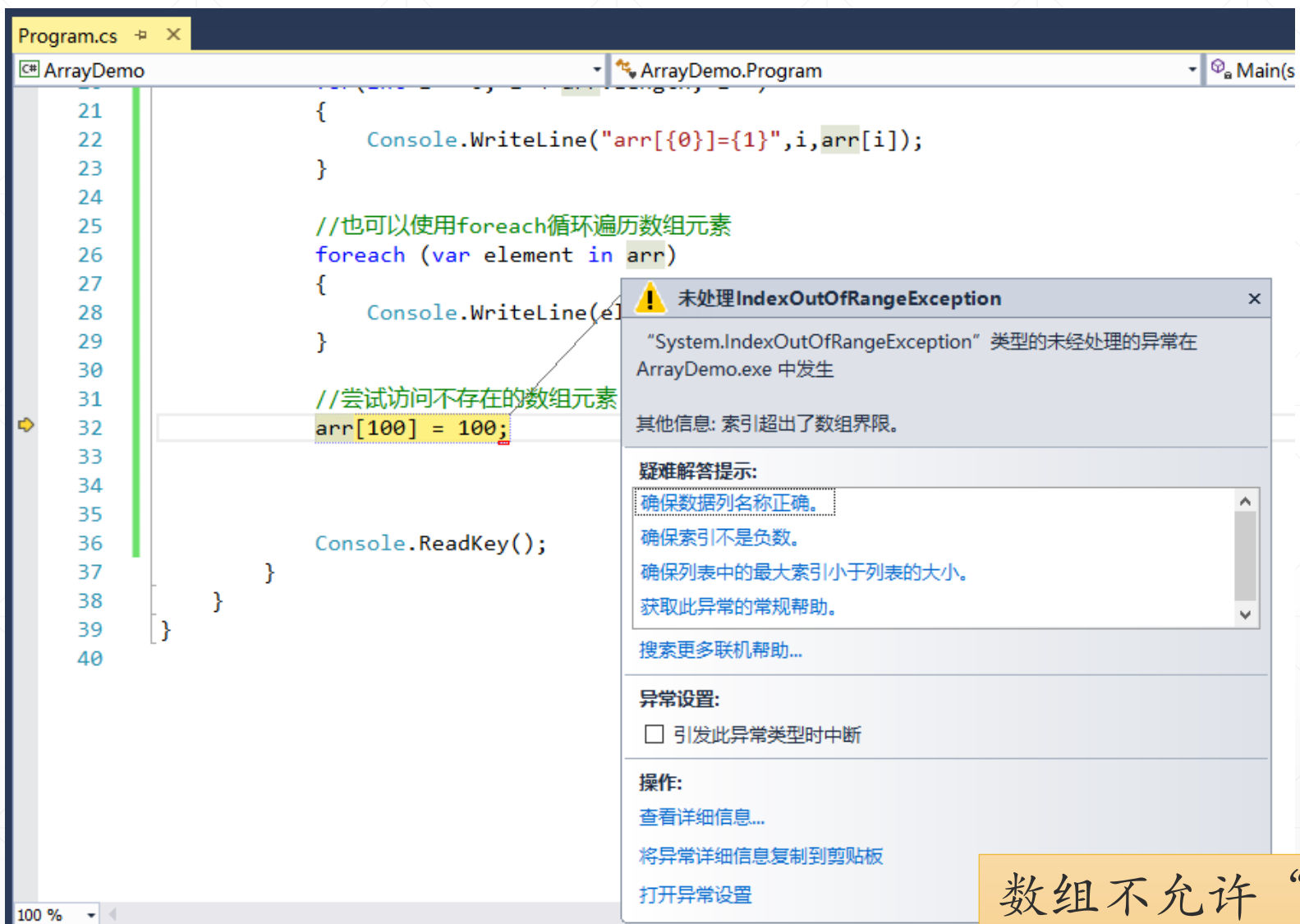
# 数组访问示例

```
// 定义一个包容8个元素的数组，并直接对其初始化
int[] arr = new int[8] { 234, 565, 23, 90, 1, 34, 89, 13 };

// 按照索引访问数组元素
arr[7]++; // 最后一个元素自增1

// 使用for循环遍历数组元素
for (int i = 0; i < arr.Length; i++)
{
    Console.WriteLine("arr[{0}]={1}", i, arr[i]);
}

// 也可以使用foreach循环遍历数组元素
foreach (var element in arr)
{
    Console.WriteLine(element);
}
```



数组不允许“越界访问”，否则，会抛出一个IndexOutOfRangeException

# 编程练习

编写一个控制台程序，程序运行时随机生成10个数，填充一个数组，然后遍历显示数组内容，接着计算数组元素的和，求出其平均值，最大值，最小值，中位数，显示在控制台窗口中。

# 对象数组

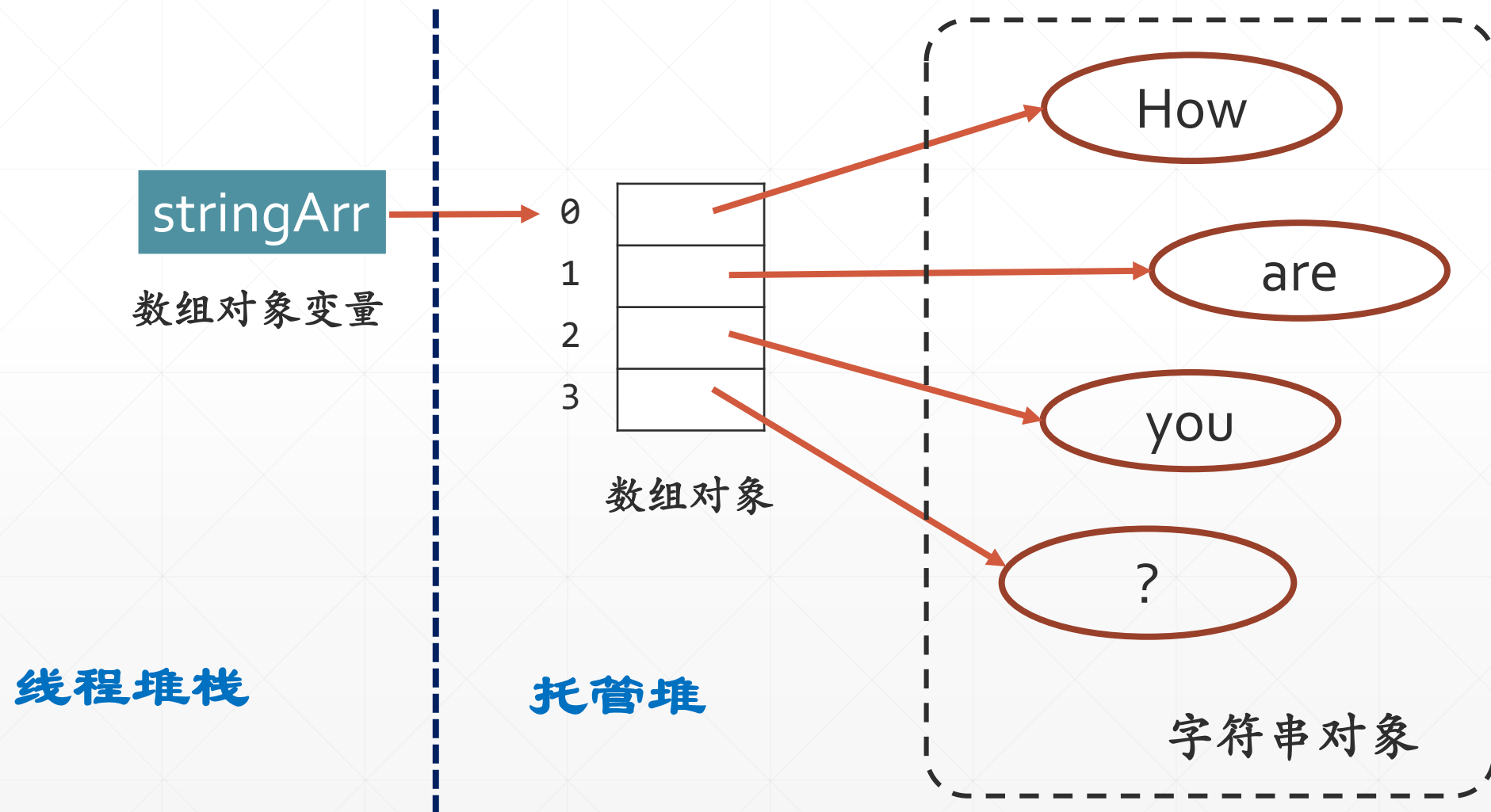
数组中的元素可以是引用类型的，这种数组俗称为“**对象数组**”。

```
var stringArr = new string[] { "How", "are", "you", "?" };
```

变量**stringArr**引用一个包容**4**个元素的数组对象，每个元素又引用一个字符串对象.....

# 对象数组的内存模型

```
var stringArr = new string[] { "How", "are", "you", "?" };
```





# 数组作为方法参数

数组本身是一个引用类型，因此，如果它被传递给一个方法，则方法内部对数组的修改，会直接作用于“原始”的数组对象。

```
//此方法接收一个字符串数组，将其中的所有字符串转换为大写  
1 个引用  
static void ModifyStringArr(string[] arr)  
{  
    for(int i = 0; i < arr.Length; i++)  
    {  
        arr[i] = arr[i].ToUpper();  
    }  
}
```

```
//字符串数组  
var stringArr = new string[] { "How", "are", "you", "?" };  
//将字符串数组传给一个方法  
ModifyStringArr(stringArr);  
//会发现原始数组的内容被方法修改，字符串全都大写了！  
foreach (var str in stringArr)  
{  
    Console.WriteLine(str);  
}
```

# 动手编程

我们已经学习了许多数组的知识，现在请编写一个程序，使用随机数填充一个数组，然后，编写一个方法对数组中的元素进行排序。

# 挑战学霸！

请编写一个程序将一个整数转换为汉字读法字符串。比如“1123”转换为“一千一百二十三”。

更进一步，能否将数字表示的金额改为汉字表达？比如将“¥123.52”转换为“壹佰贰拾叁元伍角贰分”。