**数组的定义和使用：**

1）数组是一组相关数据的集合，数组可分为一维数组、二维数组，多维数组等。

2）要使用数组，必须经过声明数组和分配内存给该数组：

**数据类型[] 数组名称=null;**

**数组名称 =new 数据类型[长度];**

**数据类型[] 数组名称=new 数据类型[长度];**

数组声明后实际上在栈内存中保存了此数组的名称，使用new关键字开辟堆内存 空间给数组。

1. 数组在未开辟堆内存空间前的默认值为null，因为数组属于引用数据类型，而引用 数据类型的默认值为null。
2. 数组的索引从0开始。
3. 要想取得一个数组的长度可以使用”数组名.length”的形式，将返回一个int型数据。

**数组的初始化：**

数组的初始化分为静态初始化和动态初始化。

1. 静态初始化指的是在声明数组变量时就给数组赋值，这样就可以不用new关键字进 行实例化了，如：int[] temp={1,2,3}。
2. 动态初始化指定的是在声明数组并使用new关键字开辟堆内存空间后再给数组赋值。

**二维数组：**

1. 二维数组类似于表格，其声明格式为：

**数据类型 数组名[][];**

**数组名 =new 数据类型[行数][列数];**

**数据类型[][] 数组名=new 数据类型[行数][列数];**

二维数组在分配内存时必须明确指定行数和列数。

1. 对于一维数组如果想要全部输出，需要使用一层循环，而对于二维数组要想全部输 出，则需要使用两层循环。同理，对于N维数组，则需要使用N层循环。
2. 二维数组也分静态初始化和动态初始化，静态初始化如：

**int[][] temp={{},{},{}};**

**数组的引用传递：**

将数组传递进方法之后，如果方法对数组的内容进行了修改，则修改的结果也将保存下 来。

**方法的定义：**

1. 方法是一段可重复调用的代码段，有些书中把方法称为函数。方法是一些操作的行 为。
2. 方法名称首字母全部小写，之后每个单词首字母大写。
3. 根据变量的作用域，在方法中定义的变量只此方法中有效，可以把它们称之为局部 变量。

**方法重载：**

方法重载指的是方法名称相同，但参数的类型或参数的个数不同，通过传递参数的个数 及类型不同以完成不同功能方法的调用。

**return语句：**

1. 使用return语句直接结束方法，往往与if语句进行结合使用。
2. 返回相应返回值类型的数据。

**可变参数：**

可变参数，即方法中的可以接受的参数不再是固定的，而是随着需要传递的。定义格 式如： **访问权限 返回值类型 方法名称(数据类型...参数名称){ };**

**foreach输出：**

利用foreach可以更加方便的对数组和集合进行输出。其语法格式为：

**for(数据类型 变量名称：数组名或集合对象){  
 ...   
 }**