# 普通工具类

## 普通工具类

**public** **class** ArrayTool {

// 1,获取最大值

**public** **int** getMax(**int**[] arr){

**int** max = arr[0]; //记录第一个元素

**for** (**int** i = 1; i < arr.length; i++) { //从第二个元素开始遍历

**if** (max < arr[i]) { //max与数组中其他元素比较

max = arr[i];

}

}

**return** max; //将最大值返回

}

// 2,数组的遍历

**public** **void** print(**int** arr[]){

**for** (**int** i = 0; i < arr.length; i++) {

System.*out*.print(arr[i] + " ");

}

}

// 3,数组的反转

**public** **void** revArray(**int** arr[]){

**for** (**int** i = 0; i < arr.length/2; i++) {

**int** temp = arr[i];

arr[i] = arr[arr.length -1];

arr[arr.length - 1] = temp;

}

}

}

## 测试

调用方式：new一个对象；对象加方法名

**public** **class** TestArrayTool {

**public** **static** **void** main(String[] args) {

ArrayTool at = **new** ArrayTool();

**int** arr[] = {122,4,5,97,-5,45};

//最大

**int** max = at.getMax(arr);

System.*out*.println(max);

//遍历打印

at.print(arr);

//反向

System.*out*.println();

at.revArray(arr);

at.print(arr);

}

}

# 静态工具类

## 静态工具类

**public** **class** ArrayToolStatic {

//私有构造方法

**private** ArrayToolStatic() {

}

// 1,获取最大值

**public** **static** **int** getMax(**int**[] arr){

**int** max = arr[0]; //记录第一个元素

**for** (**int** i = 1; i < arr.length; i++) { //从第二个元素开始遍历

**if** (max < arr[i]) { //max与数组中其他元素比较

max = arr[i];

}

}

**return** max; //将最大值返回

}

// 2,数组的遍历

**public** **static** **void** print(**int** arr[]){

**for** (**int** i = 0; i < arr.length; i++) {

System.*out*.print(arr[i] + " ");

}

}

// 3,数组的反转

**public** **static** **void** revArray(**int** arr[]){

**for** (**int** i = 0; i < arr.length/2; i++) {

**int** temp = arr[i];

arr[i] = arr[arr.length -1];

arr[arr.length - 1] = temp;

}

}

}

## 测试

类名.方法名

**public** **class** TestArrayToolStatic {

**public** **static** **void** main(String[] args) {

**int** arr[] = {122,4,5,97,-5,45};

//最大

**int** max = ArrayToolStatic.*getMax*(arr);

System.*out*.println(max);

//遍历打印

ArrayToolStatic.*print*(arr);

//反向

System.*out*.println();

ArrayToolStatic.*revArray*(arr);

ArrayToolStatic.*print*(arr);

}

}

# 说明书的制作过程

对工具类加入文档注释

通过javadoc命令生成说明书

@author(提取作者内容)

@version(提取版本内容)

javadoc -d 指定的文件目录 -author -version ArrayTool.java

@param 参数名称//形式参数的变量名称

@return 函数运行完返回的数据