1.方法定义：直接在主方法中调用的方法

public static 方法返回值 方法名称（[参数类型 变量...]）{

方法体；

[return 返回值;]

}

void 关键字声明的方法没有返回值

java中只有值传递，交换变量需用到反射。

2.方法重载

定义：方法名称相同，参数的个数，类型，顺序不同，但与方法返回值无关

编码规范：重载函数的返回值类型保持一致

数组的定义与使用

Java数组---引用数据类型

a)数组通过下标索引取得元素，下标默认从0开始，若数组下标超出数组长度，数组越界异常（运行时异常）

b)数组中每个元素都有默认值，默认值是该类型默认值

c)数组名称.length可取得数组长度

JDK1.5提供for-each增强型循环(仅限数组内容读取)

for(临时变量 : 数组名){...}

eg:for(int a : data){...}

d)数组属于引用数据类型，必须在使用前先初始化，否则会报NullPointerException(属于运行时异常)

e)引用传递：声明开辟数组时，数组在栈内存，多个栈内存可以指向同一个堆内存

eg:int[] x = new int[3];

x[2]=2;

int[] **temp = x**; //地址传递

temp[2]=50;

f)匿名数组：

System.out.println(new int[]{1,2,3})--->没有临时变量存储数组（没有栈内存指向堆内存）

1.数组初始化

1.1**动态初始化**（声明并开辟数组）

数据类型[] 数组名称 = new 数据类型 [长度]; //此时在栈内存，初识值为默认值

eg: int[] data = new [5]; //声明并开辟长度为5的整型数组，凡是见new则是开辟了新空间

1.2 数组静态初识化（数组定义时直接用{...}设置内容）

简化格式 数据类型【】 数组名称 = 数据类型{值，值...}

完整格式 数据类型【】 数组名称 = new 数据类型{值，值...}

eg:int[] arr = new int{1,2,3};

此时根据值的个数确定数组长度，并且长度不可修改。

二维数组：

动态初识化： 数据类型【】【】 数组名称 = new 数据类型【行个数】【列个数】

静态初识化：数据类型【】【】 数组名称 = new 数据类型【】【】{{第一行的值}，{第二行的值}...}