

方法

什么是方法？

什么是方法

```
• class HelloImooc{  
•     public static void main(String[] args){  
•         System.out.println( "Hello,imooc!" );  
•     }  
• }
```

什么是方法

```
Scanner sc=new Scanner(System.in);  
sc.nextInt();  
sc.next();
```

```
System.out.println();
```

所谓方法，就是用来解决一类问题的代码的有序组合，是一个功能模块。

主要内容

- 方法的声明和调用
- 方法的重载

方法声明

- 语法格式：
访问修饰符 返回类型 方法名(参数列表){
 方法体
}

方法分类

- 根据方法是否带参数、是否返回值，可分为四类：
 - 无参无返回值方法
 - 无参带返回值方法
 - 带参无返回值方法
 - 带参带返回值方法

无参无返回值方法

- 例：一行打印输出一串星号

无参带返回值方法

- 求长方形的面积

```
public int area(){}  

```

带参无返回值方法

- 定义一个求两个float类型数据最大值的方法，在方法中将最大值打印输出。

```
public void max(float a,float b){}
```

带参有返回值方法

- 定义一个求 $n!$ 的方法，然后再求 $1!+2!+3!+4!+5!$
`public int fac(int n){}`

数组作为方法参数

- 例：定义方法，打印输出数组元素的值。
`public void printArray(int[] arr){}`

数组作为方法参数

- 例：查找数组元素的值

```
public boolean search(int n,int[] arr){}
```

方法重载

- 方法名相同，参数列表不同

方法重载

- 判断下列哪些方法是重载的方法
 - `public void hello(){}`
 - `public int hello(){}`
 - `public void hello(String s){}`
 - `public void hello(int n){}`
 - `public void hello(float f1,float f2){}`
 - `public void hello1(){}`

方法重载

- 定义3个方法，实现int、double和数组类型和的问题

参数的传递问题

- 例：对两个变量的值进行交换并打印输出。

参数传递问题

- 数组作为方法参数的传值问题

可变参数列表

- 例：`public void sum(int... n){}`

可变参数列表总结

- 可变参数一定是方法中的最后一个参数
- 数组可以传递给可变参数的方法，反之不行
- 在重载中，含有可变参数的方法是最后被选中的

方法调试

总结

- 方法声明的语法格式

访问修饰符 返回类型 方法名(参数列表){
 方法体
}

总结

- **方法的重载**
 - **方法名相同，参数列表不同**

总结

- **可变参数列表：**
 - **可变参数一定是方法中的最后一个参数**
 - **数组可以传递给可变参数的方法，反之不行**
 - **在重载中，含有可变参数的方法是最后被选中的**

总结

- 方法的传值问题