



链滴

006 方法

作者: [pzs233](#)

原文链接: <https://ld246.com/article/1537282593419>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

前言

方法 是为完成一个操作而组合在一起的语句组。

6.1 定义方法

语法:

```
修饰符 返回值类型 方法名 (参数列表) {  
    // 方法体  
    语句 (组);  
}
```

方法头 (method header) = 修饰符 (modifier) + 返回值类型 (return valuetype) + 方法名 (method name) + 参数 (parameter)

如果方法有返回值, 则称为 **带返回值的方法 (value-returning method)**, 否则就称这个方法为 **void 方法 (void method)**。

定义在方法头中的变量称为**形式参数 (formal parameter) 或者简称为参 (parameter)**。参数就像占位符。当调用方法时, 就给参数传递一个值, 这个值称为 **实际参数 (actual parameter) 或 实参 (argument)**。**参数列表 (parameter list)** 指明方法中参数的类型顺序和个数。方法名和参数列表一起构成 **方法签名 (method signature)**。参数是可选的, 也就是, 方法可以不包含参数。

在其他某些语言中, 方法称为过程 (procedure) 或函数 (function)。带返回值的方法称为**函数**, 返回值类型为void的方法称为**过程**。

我们经常说“定义方法”和“声明变量”, 这里我们谈谈两者的细微区别。**定义是指被定义的条目是么, 而声明通常是指为被声明的条目分配内存来存储数据。**

6.2 调用方法

方法的调用是执行方法中的代码。

在方法定义中, 定义方法要做什么。为了使用这个方法, 必须 **调用 (call 或 invoke)** 它。根据方法是否有返回值, 调用方法有两种途径:

- 如果返回一个值, 对方法的调用通常就当作一个值处理。例如: `int large = max(3, 4);`
- 如果返回void, 对方法的调用必须是一条语句。例如, println 方法返回void: `System.out.println("Welcome to Java!");`

对带返回值的方法而言, return语句是必需的。

6.3 通过传值进行参数传递

调用方法的时候是通过传值的方式将实参传给形参的。

当调用到参数的方法时, 实参的值传递给形参, 这个过程称为 **按值传递 (pass-by-value)**。如果实

是变量而不是直接量，则将该变量的值传递给形参。无论形参在方法中是否改变，该变量都不受影响。

实参有自己的存储空间，形参在方法中有自己独立于实参的存储空间。所以，形参的改变不影响实参内容。局部变量是在调用方法时分配存储空间的，当方法返回到调用者后它就消失了。

将十六进制数转换为十进制数，运用霍纳算法：

```
int decimalValue = 0;
for (int i = 0; i < hex.length(); i++) {
    char hexChar = hex.charAt(i);
    decimalValue = decimalValue * 16 + hexCharDecimal(hexChar);
}
```

6.4 重载方法

在一个类中有两个方法，它们具有相同的名字，但有不同的参数列表。Java 编译器根据方法签名决定用哪个方法。调用方法时，Java编译器寻找最精确匹配的方法。

6.5 变量的作用域

变量的作用域 (scope of a variable) 是指变量可以在程序中引用的范围。

在方法中定义的变量称为**局部变量 (local variable)**。局部变量的作用域从声明变量的地方开始，到包含该变量的块结束为止。局部变量都必须在使用之前进行声明和赋值。

可以在一个方法中的不同块里声明同名的局部变量，但是，不能在嵌套块中或同一块中两次声明同意部变量。

生成随机的小写字母：

```
(char>('a' + Math.random() * ('z' - 'a' + 1))).
```

生成任意两个字符 ch1 和 ch2 之间的随机字符，其中 ch1 < ch2，如下所示：

```
(char)(ch1 + Math.random() * (ch2 - ch1 + 1))).
```