方法

# 引言

1. 清楚方法的定义原则；
2. 方法的重载原则；
3. 方法的递归调用。

# 具体内容

前提：现在所讲解的方法定义格式，只属于Java方法定义的其中一种组成方式。而完整的一个组成方式将随着接下来逐步渗透。

## 方法的基本定义

方法（Method）在一些书中也将其称为函数（Function）。但是两者的含义是完全相同的，那么什么是方法？为什么需要有方法？实际上所谓的方法指的就是将一段代码封装在一个结构体之中，并且这个结构体可以被开发者随时进行调用。

|  |
| --- |
| public [static] 返回值类型 方法名称([参数类型 变量，参数类型 变量，…]) {  方法中包含的代码；  [return 返回值；]  } |

返回值类型可以设置为两种：

* Java中的数据类型：基本类型、引用类型，如果现在有返回的内容则需要通过return返回；
* 不返回任何的结果：void。

实际上在进行方法编写的时候有其自己的定义格式：要求定义方法的单词第一个单词字母小写而后每个单词的字母大写（驼峰命名法）。

在学习循环语句的时候学习过break于continue语句，这两个语句主要是进行循环控制，合理的利用可以让循环语句之后的代码不在执行，但是必须与if判断一起使用。

而同样的操作在方法中也提供，如果说你的方法上使用了void声明，那么就可以在方法执行过程之中直接利用“return”结束方法调用。

不管什么情况，只要是在return之后的代码都不再执行，但是需要区分的是，如果方法返回的是void，那么久可以直接写return，但是如果方法本身有返回值，那么就必须编写return的同时返回内容。

实际的开发中是否使用方法有两个决定性原则：

* 主方法里的代码是不是太多了，如果多，可以将其拆分到方法之中；
* 在你编写代码的过程之中总是在不断的进行复制和粘贴，就可以考虑将这部分代码封装为方法。

## 方法重载

方法重载指的是方法名称相同，参数的类型或个数不同。

如果说功能相同的方法名字不一样，那么在调用的时候实在太麻烦了。所在这样的情况下是最好的解决方案就是利用重载解决。

调用重载方法时会根据传递的参数以及个数调用不同的方法体执行。

但是在进行方法重载的时候有一项重要的说明——关于返回值问题。在方法重载的时候本身只规定了参数的类型与个数的不同，但是并没有针对于返回值的类型是否相同进行限制。但是从实际来讲，方法重载时返回值一定要相同。

## 方法的递归调用

如果以开发中出现的编写代码为主的话，那么递归操作几乎不会出现，但是如果要想分析一些开发工具原理的话，那么必须使用递归。递归是迈入数据结构的第一步，但是这并不是说一下子就可以理解递归。

所谓的递归指的就是方法自己调用自己的情况。而对于递归操作一般都要求满足于如下形式：

* 递归必须在方法上进行；
* 递归中必须有明确的结束条件；
* 递归的过程之中必须进行结束条件的变更。

范例：利用while循环实现累加操作。

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int x = 1;  int sum = 0;  while (x <= 100) {  sum += x;  x ++;  }  System.out.println(sum);  }  } |

那么下面的代码将改成递归操作

范例：递归调用

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  System.out.println(sum(100));  }  public static int sum(int num) {  if (num == 1) {  return 1;  }  return num + sum(num - 1);  }  } |

对于此代码的调用分析如下：

* 第一次调用（主方法）：return 100 + sum(99);
* 第二次调用（sum(99)方法）：return 99 + sum(98);
* 第三次调用（sum(98)方法）：return 98 + sum(97);
* 倒数第二次调用（sum(2)方法）：return 2 + sum(1);
* 最后一次调用（sum(1)方法）：if判断满足，return 1。

最终变成：100 + 99 + 98 + 97 + … + 2 + 1（if语句）；

实际上递归操作有一件很可怕的影响：由于其操作所占用的时间过长，那么有可能会造成内存溢出导致程序出现问题，所以如果不是必须的情况下，不建议使用递归操作。

# 总结

1. 方法的定义结构。
2. 方法的重载：方法名称相同，参数的类型或个数不同，方法重载对返回值类型没有要求，但是一定要统计返回值类型。