



链滴

005 循环

作者: [pzs233](#)

原文链接: <https://ld246.com/article/1537246593934>

来源网站: [链滴](#)

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

引言

Java 提供了三种类型的循环语句：while 循环、do-while 循环和 for 循环。

5.1 while 循环

while循环在条件为真的情况下，重复地执行语句。

语法：

```
while(循环继续条件) {  
    // 循环体  
    语句 (组) ;  
}
```

要保证循环继续条件最终可以变成 `false`，以便程序能结束。

计数器控制的循环 (counter-controlled loop)：确切的知道循环体需要执行的二次数，使用一个制变量count 来对执行次数计数。

在循环控制中，不要使用浮点值来比较是否相等。因为浮点值都是某些值的近似值，使用它们可能导不精确的循环次数和不准确的结果。

输入和输出重定向

如果要输入大量的数值，那么从键盘上输入时非常繁琐的事。可以将这些数据用空格隔开，保存在一名为input.txt的文本文件中，然后使用下面的命令运行这个程序：`java SentinelValue < input.txt`（入重定向）。此句不是加在源代码中，而是在命令行运行时使用，作用是从指定文件读取数据。`java C assName > output.txt`(输出重定向)

5.2 do-while 循环

do-while循环和while循环基本一样，不同的是它先执行循环一次，然后判断循环继续条件。

语法：

```
do {  
    // 循环体  
    语句 (组) ;  
} while (循环继续条件) ;
```

5.3 for 循环

语法：

```
for (初始操作; 循环继续条件; 每次迭代后的操作){  
    // 循环体  
    语句 (组) ;  
}
```

for循环中的初始动作可以是0个或多个以逗号隔开的语句。

for循环中每次迭代后的动作可以是0个或多个以逗号隔开的语句。

5.4 关键字 break 和 continue

continue 只是跳出了一次迭代，而关键字break是跳出了整个循环。

通常，只有在能够简化代码并使用更容易阅读的情况下，才合适使用 break 和 continue。

END