

计算机二级java循环语句 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_138712.htm Java 的循环语句

有for，while 和 do-while。这些语句创造了我们通常所称的循环（loops）。你可能知道，一个循环重复执行同一套指令直到一个结束条件出现。你将看到，Java 有适合任何编程所需要的循环结构。

5.2.1 while 语句

while 语句是Java 最基本的循环语句。当它的控制表达式是真时，while 语句重复执行一个语句或语句块。它的通用格式如下：`while(condition) { // body of loop }` 条件condition 可以是任何布尔表达式。只要条件表达式为真，循环体就被执行。当条件condition 为假时，程序控制就传递到循环后面紧跟的语句行。如果只有单个语句需要重复，大括号是不必要的。下面的while 循环从10开始进行减计数，打印出10行“tick”。

```
// Demonstrate the while loop.class  
While {public static void main(String args[]) {int n = 10. while(n > 0) {System.out.println("tick " n).n--. }}}
```

当你运行这个程序，它将“tick” 10次：tick 10 tick 9 tick 8 tick 7 tick 6 tick 5 tick 4 tick 3 tick 2 tick 1

因为while 语句在循环一开始就计算条件表达式，若开始时条件为假，则循环体一次也不会执行。例如，下面的程序中，对println() 的调用从未被执行过：`int a = 10, b = 20. while(a > b)System.out.println("This will not be displayed").`

while 循环（或Java 的其他任何循环）的循环体可以为空。这是因为一个空语句（null statement）（仅由一个分号组成的语句）在Java 的语法上是合法的。例如，下面的程序：

```
// The target of  
a loop can be empty.class NoBody {public static void main(String
```

```
args[]) {int i, j. i = 100. j = 200. // find midpoint between i and
jwhile( i System.out.println("Midpoint is " i).}}该程序找出变量i和
变量j的中间点。它产生的输出如下：Midpoint is 150 该程序中的
while 循环是这样执行的。值i自增，而值j自减，然后比较
这两个值。如果新的值i仍比新的值j小，则进行循环。如果i
等于或大于j，则循环停止。在退出循环前，i将保存原始i和j
的中间值（当然，这个程序只有在开始时i比j小的情况下才执
行）。正如你看到的，这里不需要循环体。所有的行为都出
现在条件表达式自身内部。在专业化的Java代码中，一些可
以由控制表达式本身处理的短循环通常都没有循环体。5.2.2
do-while 循环如你刚才所见，如果while 循环一开始条件表达
式就是假的，那么循环体就根本不被执行。然而，有时需要
在开始时条件表达式即使是假的情况下，while 循环至少也要
执行一次。换句话说，有时你需要在一次循环结束后再测试
中止表达式，而不是在循环开始时。幸运的是，Java 就提供
了这样的循环：do-while 循环。do-while 循环总是执行它的循
环体至少一次，因为它的条件表达式在循环的结尾。它的通
用格式如下：do {// body of loop} while (condition). do-while 循
环总是先执行循环体，然后再计算条件表达式。如果表达式
为真，则循环继续。否则，循环结束。对所有的Java 循环都
一样，条件condition 必须是一个布尔表达式。下面是一个重
写的“ tick ”程序，用来演示do-while 循环。它的输出与先前
程序的输出相同。// Demonstrate the do-while loop.class
DoWhile {public static void main(String args[]) {int n = 10. do
{System.out.println("tick " n).n--. } while(n > 0).}}该程序中的循
环虽然在技术上是正确的，但可以像如下这样编写更为高效
```

: do {System.out.println("tick " n). } while(--n > 0). 在本例中，表达式 “ -- n > 0 ” 将n值的递减与测试n是否为0组合在一个表达式中。它的执行过程是这样的。首先，执行-- n 语句，将变量n递减，然后返回n的新值。这个值再与0比较，如果比0大，则循环继续。否则结束。do-while 循环在你编制菜单选择时尤为有用，因为通常都想让菜单循环体至少执行一次。下面的程序是一个实现Java 选择和重复语句的很简单的帮助系统：

```
// Using a do-while to process a menu 0selectionclass
Menu {public static void main(String args[])throws
java.io.IOException {char choice. do {System.out.println("Help
on:").System.out.println(" 1. if").System.out.println(" 2.
switch").System.out.println(" 3. while").System.out.println(" 4.
do-while").System.out.println(" 5.
for\n").System.out.println("Choose one:").choice = (char)
System.in.read().} while( choice 5). System.out.println("\n").
switch(choice) {case 1: System.out.println("The
if:\n").System.out.println("if(condition)
statement.").System.out.println("else statement.").break.case 2:
System.out.println("The
switch:\n").System.out.println("switch(expression)
{").System.out.println(" case constant:"). 100Test 下载频道开通，
各类考试题目直接下载。详细请访问 www.100test.com
```