使用一个控制变量count来对执行次数计数。这种类型的循环称为计数器控制的循环。

另一种控制循环的常用技术是在读取和处理一个集合的值时指派一个特殊值。这个特殊的输入值也称为标记值，用以表明循环的结束。如果一个循环使用标记值来控制它的执行，它就称为标记位控制的循环。

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

**输入和输出重定向**

如果要从键盘上输入大量的数值，可以将这些数据用空格隔开，保存在一个名为input.txt文本文件中，然后使用下面的命令行运行这个程序：

java SentinelValue < input.txt

这个命令称为输入重定向（input redirection）。程序从文件input.txt中读取输入。

输出重定向（output redirection），输出重定向将输出发送给文件，而不是将它们显示在控制台上。

java ClassName > output.txt

可以在同一命令中同事使用输入重定向和输出重定向。例如，下面的命令从文件input.txt中获取输入，并将输出发送给文件output.txt

java SentinelValue <input.txt> output.txt

for循环中的初始动作可以是0个或是多个以逗号隔开的变量声明语句或赋值表达式。例如

for(int i=0,j=0;i+j<10;i++,j++){

}

while循环和for循环都称为前测循环（pretest loop），因为继续条件是在循环体执行之前检测的，do-while循环称为后测循环（posttest loop)，因为循环条件是在循环体执行之后检测的。

在循环继续条件中使用浮点数将导致数值错误。在较大数之前先增加较小数是减少误差的一种方法。即 循环从小到大添加数字。

当程序遇到continue时，它会结束当前的迭代。程序控制转向该循环体的末尾。continue只是跳出了一次迭代，而关键字break是跳出了整个循环。

continue 语句总是在一个循环中。在while和do-while循环中，continue语句之后会马上计算循环继续条件；而在for循环中，continue语句之后会立即先执行每次迭代后的动作，在计算循环急需条件。

循环体执行一次称为循环的一次迭代。

break立即终止包含break的最内层循环。