

- 1 块作用域
- 2 条件语句
- 3 循环
- 4 `switch` 语句
- 5 中断控制流程的语句
  - 5.1 `break` 语句
  - 5.2 `continue` 语句

# 1 块作用域

块（复合语句）是指若干条 Java 语句组成的语句，并用一对大括号括起来。块确定了变量的作用域。

一个块可以嵌套在另一个块中。**不能在嵌套的两个块中声明同名变量。**

# 2 条件语句

`if` 语句：当条件满足时，执行语句体；条件不满足时，不执行语句体。

```
if (condition)
    statement
```

`if-else` 语句：当条件满足时，执行 `if` 语句体；条件不满足时，执行 `else` 语句体。

```
if (condition)
    statement1
else
    statement2
```

其中 `else` 子句与最邻近的未配对的 `if` 构成一组。

可以反复使用 `if...else if...` 构成多分支结构：

```
if (condition1)
    statement1
else if (condition2)
    statement2
// ...
else
    statement_n
```

## 3 循环

`while` 语句：

```
while (condition)
    statement
```

`do-while` 语句：

```
do
    statement
while (condition);
```

`for` 循环：

```
for (expression1; expression2; expression3)
    statement
```

表达式 1 通常是对计数器初始化，表达式 2 给出循环条件，表达式 3 更新计数器。

有一条不成文的规则：`for` 语句的 3 个表达式应该对同一个计数器变量进行初始化、检测和更新。

注意：在循环条件中检测两个浮点数是否相等需要格外小心，否则可能出现死循环。

## 4 `switch` 语句

`switch` 语句的基本结构如下：

```
switch (expression)
{
    case condition1:
        // 操作 1
        break;
    case condition2:
        // 操作 2
        break;
    /* ... */
    case condition_n:
        // 操作 n
        break;
    default:
        // 例外操作
        break;
}
```

`switch` 语句将从与括号内表达式的值相匹配的 `case` 标签开始执行，直到遇到 `break` 语句，或者执行到 `switch` 语句的结束处为止。如果没有匹配的 `case` 标签，而有 `default` 子句，就执行这个子句。

`case` 标签可能是类型为 `char`、`byte`、`short`、`int` 的常量表达式，或者是枚举常量。从 Java 7 开始，`case` 标签还可以是字符串字面量。

## 5 中断控制流程的语句

### 5.1 `break` 语句

`break` 语句可以用来退出 `switch` 语句和循环语句。

Java 还提供了带标签的 `break` 语句，用于跳出多重嵌套的循环语句。标签必须放在希望跳出的最外层循环之前，并且紧跟一个冒号。例如：

```
Scanner in = new Scanner(System.in);
int n;

read_data:
while (...)
{
    for (...)
    {
        n = in.nextInt();
        if (n < 0)
            break read_data; // 退出 read_data 标记的 while 语句
    }
}
```

事实上，可以将标签应用到任何语句，执行带标签的 `break` 语句会跳转到带标签的语句块末尾。

## 5.2 `continue` 语句

`continue` 语句将中断正常的控制流程，转移到最内层循环的首部。

还有一种带标签的 `continue` 语句，将跳到与标签匹配的循环的首部。