程序逻辑控制

# 引言

1. 分支语句；
2. 循环语句。

# 具体内容

程序主要分为三种逻辑：顺序、分支、循环。

## if分支语句

if分支语句最为基础的分支操作，但是其有三种使用形式。

|  |  |  |
| --- | --- | --- |
| if语句： | if…else | if…else if…else if…else |
| if（布尔表达式）{  条件满足时执行；  } | if (布尔表达式) {  条件满足时执行；  } else {  条件不满足时执行；  } | if (布尔表达式) {  条件满足时执行；  } else if(布尔表达式) {  条件不满足时执行；  }…[else {  条件都不满足时执行；  }] |

在整个开发之中，分支语句使用的是最多的，并且往往可能会结合多个判断条件一起判断。

## switch分支语句

利用if…else可以实现多条件的验证，而switch只能够进行多数值的判断，而且switch本身支持的判断数据类型：

* 标准做法：int、char；
* JDK 1.5支持了枚举；
* JDK 1.7支持了字符串。

对于switch语法如下：

|  |
| --- |
| switch(变量 | 内容) {  case 数值 : {  数值满足时执行的语句；  [break ;]  }  case 数值 : {  数值满足时执行的语句；  [break ;]  }  case 数值 : {  数值满足时执行的语句；  [break ;]  }…  [default : {  如果没有break，就是默认执行的语句；  }]  } |

在使用switch语句之中，如果每一个case之后不加上break。那么表示在第一个case语句满足条件之后开始执行，执行遇到break为止，或执行到default为止。所以好的习惯一定要在每一个case中使用break。

## 循环语句

循环的意义本身非常的好理解，就是让一段代码重复执行。但是在循环语法之中，实际上定义有两类循环：while循环、for循环。

### while循环

while循环中分为两种形式：while、do…while。

|  |  |
| --- | --- |
| while循环 | do…while循环 |
| while (判断循环结束条件) {  执行循环语句 ；  修改循环控制 ；  如果不修改会出现死循环；  } | do {  执行循环语句 ；  修改循环控制 ；  如果不修改会出现死循环；  } while (判断循环结束条件) |

### for循环

for循环的语法比while循环简化，其形式如下：

|  |
| --- |
| for(循环初始化条件；循环结束判断；改变循环条件) {  循环语句；  } |

### 循环总结

现在给出了while和for两种循环，那么实际开发中用哪个呢？

* 如果已经明确知道循环次数就使用for循环；
* 如果不知道循环次数，但是知道结束条件，那么使用while循环。

## 循环控制

循环默认情况下肯定要一直执行,知道循环条件满足,但是在循环的执行中提供有两个循环的控制语句:continue（中断一次）、break（跳出循环）。

一般这样的两个语句都会结合if语句来一起使用。

# 总结

先阶段不要求大家编写过于复杂的循环操作。