# 尚马教育 JAVA 基础课程

# Java流程控制语句

文档编号：A03

创建日期： 2017-04-12

最后修改日期：2020-12-09

版 本 号：V3.0

电子版文件名：尚马教育-第一阶段-3.Java流程控制语句专题课程.docx

**文档修改记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| 更新日期 | 更新作者 | 更新说明 | 版本号 |
| 2017-07-30 | 张元林 | 初始版本 | V1.0 |
| 2018-08-01 | 王绍成 | Java基础版本更新 | V2.0 |
| 2019-08-09 | 徐丽莎 | Java基础版本更新 | V3.0 |

**主讲人：**

**武汉-刘蕾**

目录

[尚马教育 JAVA 基础课程](#_Toc13453)

[Java流程控制语句](#_Toc876)

[1. 程序结构](#_Toc281)

[1.1. 顺序结构](#_Toc13604)

[1.2. 分支结构](#_Toc3582)

[1.3. 循环结构](#_Toc28653)

[2. 分支结构](#_Toc25182)

[2.1. 条件判断结构](#_Toc26947)

[2.1.1. 语法](#_Toc1304)

[2.1.2. Scanner](#_Toc32378)

[2.1.3. 案例](#_Toc20575)

[2.2. 条件选择结构](#_Toc29309)

[2.2.1. 语法](#_Toc3822)

[2.2.2. switch和多重if](#_Toc4782)

[2.2.3. 案例](#_Toc6116)

[3. 循环结构](#_Toc17360)

[3.1. while循环](#_Toc1686)

[3.1.1. 语法](#_Toc16148)

[3.1.2. 案例](#_Toc21975)

[3.2. do while循环](#_Toc18631)

[3.2.1. 语法](#_Toc8832)

[3.2.2. 案例](#_Toc13615)

[3.3. for循环](#_Toc14144)

[3.3.1. 语法](#_Toc30109)

[3.3.2. 案例](#_Toc21509)

[4. break](#_Toc30398)

[5. continue](#_Toc30068)

[6. return](#_Toc8870)

[7. 练习](#_Toc17731)

## 程序结构

* 结构化程序设计一共有三种形式: 顺序结构,选择结构，循环结构
* 采用[结构化程序设计方法](https://www.baidu.com/s?wd=%E7%BB%93%E6%9E%84%E5%8C%96%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%E6%96%B9%E6%B3%95&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)，程序结构清晰，易于阅读、测试、排错和修改。由于每个模块执行单一功能，模块间联系较少，使程序编制比过去更简单，程序更可靠，而且增加了可维护性，每个模块可以独立编制、测试。

### 顺序结构

* 顺序结构：顺序结构是一种线性、有序的结构，它依次执行各语句模块。

### 分支结构

* 选择结构是根据条件成立与否选择程序执行的通路

### 循环结构

* 循环结构是重复执行一个或几个模块，直到满足某一条件为止

## 分支结构

* 在我们之前的学习中，我们所编写的java程序都是顺序执行，代码自上而下运行。在实际的功能开发中，其实是要大量运用分支结构或者循环结构的。
* 在java语言里面，分支结构分为以下2种:
  + 条件判断
  + 选择结构

### 条件判断结构

* 也就是我们要学习的 if...else语句

#### 语法

* if条件结构是根据条件判断之后再做处理。

|  |  |
| --- | --- |
| **if**(判断语句1){  执行语句; }  判断语句的返回值必须是boolean型，为true的时候执行if逻辑。 |  |

* if/else语句指的是如果满足某种条件，就执行if代码块；如果不满足，则执行else代码块

|  |  |
| --- | --- |
| **if**(判断语句){  执行语句1; }**else**{  执行语句2; } |  |

* if/else if/else，else if语句可以多个

|  |  |
| --- | --- |
| **if**(判断语句1){  执行语句1; }**else if**(判断语句2){  执行语句2; }  …… **else if**(判断语句n){  执行语句n; }**else**{ } //不满足判断语句1，但是满足2，则执行语句2 |  |

s

#### 案例

* 如果小明的Java考试成绩大于90分，老师就奖励他一部手机

|  |
| --- |
|  |

* 如果小明的Java考试成绩大于90分，老师就奖励他《五年高考三年模拟》

|  |
| --- |
|  |

* 小明语文成绩大于90分，而且历史成绩大于80分，老师奖励他；或者语文成绩等于100分，历史成绩大于70分，老师也可以奖励他

|  |
| --- |
|  |

* 如果李雷考试成绩大于90分，老师就奖励他《五年高考三年模拟》，否则老师就罚他站着听课

|  |
| --- |
|  |

* 对学员的结业考试成绩评测
  + 成绩>=90 ：优秀
  + 成绩>=80 ：良好
  + 成绩>=60 ：中等
  + 成绩<60 ：差

|  |
| --- |
|  |

* 输入李雷的考试成绩，显示所获奖励
  + 成绩==100分，爸爸给他买辆车
  + 成绩>=90分，妈妈给他买电脑
  + 90分>成绩>=60分，妈妈给他买本参考书
  + 成绩<60分，什么都不买

|  |
| --- |
|  |

### 条件选择结构

也就是我们所要学习的switch...case语句。

#### 语法

|  |  |
| --- | --- |
| **switch**(表达式) {  **case** 常量表达式1：  语句1;  **break**;  **case** 常量表达式2：  语句2;  **break**;  ......  **case** 常量表达式n：  语句n;  **break**;  **default**：  语句n+1;  **break**; } | case之间与default没有顺序要求。先执行第一个case，没有匹配的case执行default。  表达式：java 1.6(包括)以前，只支持:byte,short,char,int，enum，1.7加入的新特性可以支持String类型的数据 |

#### switch和多重if

* 相同点
  + 都可以实现多分支结构
* 不同点
  + Switch 只能处理等值的条件判断，且条件是整型变量或字符变量的等值判断
  + 多重if 处理在else部分还包含其它if结构，特别适合某个变量处于某个区间时的情况

#### 案例

* 李雷参加英语比赛
  + 如果获得第一名，奖励1000元
  + 如果获得第二名，奖励500元
  + 如果获得第三名，奖励100元
  + 否则，将被罚款300元

|  |
| --- |
|  |

* 韩梅梅为她的手机设定了自动拨号
  + 按1：拨爸爸的号
  + 按2：拨妈妈的号
  + 按3：拨爷爷的号
  + 按4：拨奶奶的号
* 请分别使用if和switch实现：

自己给定一个月份，判断属于哪个季节。（3-5月为春季、6-8月为夏季、9-11月为秋季、12,1,2月为冬季）

|  |
| --- |
| Scanner input = **new** Scanner(System.***in***);// 导包  System.***out***.println("请录入月份(1,2):");  **int** monthNum = input.nextInt();// 用户动态录入月份数据  // 等值判断操作  **switch** (monthNum) {//int byte short char enum String(jdk1.7)  **case** 12:  **case** 1:  **case** 2:  System.***out***.println(monthNum+"月是冬季");  **break**;//遇见break 结束switch语句  **case** 3:  **case** 4:  **case** 5:  System.***out***.println(monthNum+"月是春季");  **break**;  **case** 6:  **case** 7:  **case** 8:  System.***out***.println(monthNum+"月是夏季");  **break**;  **case** 9:  **case** 10:  **case** 11:  System.***out***.println(monthNum+"月是秋季");  **break**;  **default**:  System.***out***.println("录入数据不合法");  **break**;  } |

## 循环结构

### 循环

* 什么是循环？
  + 《[战国策·燕策二](https://baike.baidu.com/item/%E6%88%98%E5%9B%BD%E7%AD%96%C2%B7%E7%87%95%E7%AD%96%E4%BA%8C/2140406)》：“此必令其言如循环，用兵如刺蜚绣。”往复回旋。指事物周而复始地运动或变化。意思是转了一圈又一圈，一次又一次地循回。
* 生活中的循环有？

### 循环四要素

* 分析4\*400米？跑几圈，什么时候开始？什么时候停止？怎么知道跑了4圈？
  + 一共跑4圈。跑完4圈才停止。
  + 从第1圈开始跑。
  + 要计数1、2、3、4才知道跑了4圈。
  + 因为是接力赛肯定跑的越快越好了。所以要用力跑。
* 通过上述，分析出，循环实际上包含：
  + 什么时候停止？ 🡪 停止条件
  + 什么时候开始？ 🡪 从1开始【起点】
  + 没跑一圈要开始计数。 🡪 循环计数
  + 中间要一直跑. 🡪 循环体
* 循环四要素：
  + 停止条件 🡪 控制条件
  + 从1开始 🡪 循环变量的初始化
  + 循环计数 🡪 循环变量值的改变。【自增/自减】
  + 循环体

### 循环的分类：

* Java语言中的有三种循环语句，分别：
  + while
  + do … while
  + for

### while循环

* 特点：先判断，再执行

#### 语法

|  |  |
| --- | --- |
| while(判断条件语句){  循环体语句块；  控制语句；  }  判断条件语句，返回true，执行循环体，否则跳出循环体；  循环体，每次判断语句返回true，执行一次  控制语句，修改循环变量。执行一次循环体后执行。 |  |

#### 案例

* 输出10次”我能行”

|  |
| --- |
| **int** i = 1;  **while** ( i <= 10 ) {  System.***out***.println("我能行！");  i ++;  } |

* 使用while循环实现以下操作:

为了备战，李雷锲而不舍地练习，韩梅梅严格把关…

“韩梅梅，怎么样，可以了吗？”

“不行，高音部分唱得还不是很好，钢琴还要继续练啊 ！”

没有听到“很棒”的评价，看来革命尚未成功, 李雷并不气馁:

早上5点练声，上午练钢琴，下午到声乐老师家练习唱歌，晚上练习舞蹈基本功

|  |
| --- |
| Scanner input = **new** Scanner(System.***in***);  System.***out***.println("韩梅梅，怎么样，可以了吗? y/n");// y 很棒 n 不行  String answer = input.nextLine();// 动态录入的数据  **while** ("n".equals(answer)) {  System.***out***.println("早上5点练声，上午练钢琴，下午到声乐老师家练习唱歌，晚上练习舞蹈基本功");  System.***out***.println("韩梅梅，我又练习1天了，怎么样，可以了吗? y/n");  answer = input.nextLine();// 为什么使用同一个变量answer?  }  // 不执行循环 证明练习不错了  **if** ("y".equals(answer)) {  System.***out***.println("练习很棒了");  } |

* 动态录入班级人数和学员成绩，计算班级学员的平均成绩

|  |
| --- |
| **public** **static** **void** main(String[] args) {  //动态录入班级人数和学员成绩，计算班级学员的平均成绩  Scanner input = **new** Scanner(System.***in***);  System.***out***.println("请录入班级人数：");  **int** studentCount = input.nextInt();  **int** num = 1;  **double** totalScore = 0;// 总成绩  **while** (num <= studentCount) {  // 动态的录入每个学生的成绩 循环体  System.***out***.println("请录入第" + num + "个学员的成绩:");  **int** studentScore = input.nextInt();  totalScore += studentScore;// totalScore = totalScore+studentScore;  num++;  }  **double** avg = totalScore / studentCount;  System.***out***.println("计算班级学员的平均成绩:" + avg);  input.close();  } |

* 2014年苹果手机售出10万，每年增长45%，请问按此增长速度，到哪一年苹果手机的出货量将达到100万

|  |
| --- |
| **double** num1 = 100000;  **int** year = 2014;  **do** {  num1+= num1\*0.45;//num1= num1+num1\*0.45;  year++;  }**while**(num1<1000000);  System.***out***.println(year); |

### do…while循环

* 特点:先执行循环体，然后再进行条件判断。

#### 语法

|  |  |
| --- | --- |
| do{  循环体语句块；  控制语句；  } while(判断条件语句)  **;** | 循环体，第一次无条件执行，后续每次判断语句返回true，执行一次  控制语句，修改循环变量。  判断条件语句，返回true，执行循环体，否则跳出循环体； |

#### 案例

|  |
| --- |
| **int** c = 0;  **do** {  System.***out***.println("c=" + c);  c++;  } **while** (c < 5); |

### while和do…while有什么区别？

* 相同：
  + while和do…while都可以执行循环操作。
  + while和do…while都有控制条件，只有条件成立才能继续执行循环体中的内容。
* 不同：
  + while是先判断，再执行。只有当控制条件成立，才能执行循环体中共的内容。
  + do…while：先执行一次，再判断。无论条件如何，至少执行一次循环体。

### for循环

#### 语法

|  |
| --- |
| **for**(初始化语句；判断条件语句；控制语句){  循环体语句块； }  初始化语句，给循环变量赋初值；  判断条件语句，返回true，执行循环体，否则跳出循环体；  控制语句，修改循环变量。执行一次循环体后执行。 |
|  |

**for(;;){} 这样的语句会编译通过吗？如果通过，是个什么样的循环？**

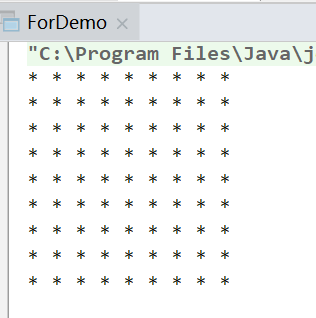
#### 案例

* 使用for循环输出10次”我能行”

|  |
| --- |
| **for**(**int** num=1;num<=10;num++) {  System.***out***.println("苏一航----"+num);  } |

### 嵌套循环

* **什么是嵌套循环？**
  + 把一个循环当作另一个循环的循环体。
  + 换句话说就是：允许在一个循环内使用另外一个循环。
* 打印9\*9的正方形？



|  |
| --- |
| */\*\*  \* 打印9\*9的正方形  \* 横向9行  \* 竖向9列  \*/* **for** (**int** i = 0; i < 9;i++){  **for** (**int** j = 0; j < 9;j++){  *//因为在一横排，所以不换行* System.***out***.print(**"\* "**);  }  *//换行* System.***out***.println(); } |

## break

* 在循环控制语句的循环体中，可以使用break语句，表示终止当前循环，跳出循环体；

#### 案例

* 打印输出0-4中的第一个偶数

|  |
| --- |
|  |

## continue

* 在循环控制语句的循环体中，可以使用continue语句，表示不再继续循环体后面的代码，继续下一次循环

#### 案例

* 使用continue输出0-4中的所有奇数

|  |
| --- |
|  |

* 李雷在说100次“我能行”的过程中,韩梅梅突然在第十次叫停了他

|  |
| --- |
|  |

## return

return关键字不是为了跳转出循环体，更常用的功能是结束一个方法，也就是退出一个方法。跳转到上层调用的方法。这个在方法的使用那里会在详细的讲解