

《Java面向对象程序设计》 第3章

运算符、表达式和语句

导读

主要内容

- 运算符与表达式
- 语句概述
- **if**条件分支语句
- **switch**开关语句
- 循环语句
- **break**和**continue**语句
- 枚举类型与**for**、**switch**语句

难点

- 循环语句
- 枚举类型与**for**、**switch**语句

§ 3.1 运算符与表达式

Java提供了丰富的运算符，如算术运算符、关系运算符、逻辑运算符、位运算符等。

§ 3.1.1 算术运算符与算术表达式

1. 加减运算符：+，-

加减运算符是二目运算符；加减运算符的结合方向是从左到右；加减运算符的操作元是整型或浮点型数据，加减运算符的优先级是4级。

2. 乘、除和求余运算符：*，/，%

以上运算符是二目运算符，结合方向是从左到右，乘、除和求余运算符的操作元是整型或浮点型数据。运算符的优先级是3级。

3. 算术表达式

用算术符号和括号连接起来的符合java语法规则的式子，称为算术表达式。

§ 3.1.2 自增，自减运算符

自增、自减运算符：++，--

是单目运算符，可以放在操作元之前，也可以放在操作元之后。操作元必须是一个整型或浮点型变量。作用是使变量的值增1或减1，如：

`++x` (`--x`) 表示在**使用x之前**，先使x的值增（减）1。

`x++` (`x--`) 表示在**使用x之后**，使x的值增（减）1。

§ 3.1.3 算术混合运算的精度

精度从“低”到“高”排列的顺序是：

byte short char int long float double

Java在计算算术表达式的值时,使用下列计算精度规则:

1. 如果表达式中有双精度浮点数（double型数据），则按双精度进行运算。
2. 如果表达式中最高精度是单精度浮点数（float型数据），则按单精度进行运算。
3. 如果表达式中最高精度是long型整数，则按long精度进行运算。
4. 如果表达式中最高精度低于int型整数，则按int精度进行运算。

§ 3.1.4 关系运算符与关系表达式

关系运算符:

关系运算符是二目运算符，用来比较两个值的关系。关系运算符的**运算结果是boolean型**，当运算符对应的关系成立时，运算结果是**true**，否则是**false**。

表 3.1 关系运算符

运算符	优先级	用法	含义	结合方向
>	6	op1>op2	大于	左到右
<	6	op1<op2	小于	左到右
>=	6	op1>=op2	大于等于	左到右
<=	6	op1<=op2	小于等于	左到右
==	7	op1==op2	等于	左到右
!=	7	op1!=op2	不等于	左到右

§ 3.1.5 逻辑运算符与逻辑表达式

逻辑运算符包括

&&, ||, !

其中**&&**、**||**为二目运算符，实现逻辑与、逻辑或；

!为单目运算符，实现逻辑非。

逻辑运算符的操作元必须是boolean型数据，逻辑运算符可以用来连接关系表达式。

表 3.2 逻辑运算符

运算符	优先级	用法	含义	结合方向
&&	11	op1&&op2	逻辑与	左到右
	12	op1 op2	逻辑或	左到右
!	2	!op	逻辑非	右到左

表 3.3 用逻辑运算符进行逻辑运算

op1	op2	op1&&op2	op1 op2	!op1
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

§ 3.1.6 赋值运算符与赋值表达式

赋值运算符： =

赋值运算符是二目运算符，左面的**操作元必须是变量**，不能是常量或表达式。

赋值运算符的优先级较低，是14级，结合方向右到左。

赋值表达式的值就是“=”左面变量的值。

注意：不要将赋值运算符“=”与等号逻辑运算符“==”混淆。

§ 3.1.7 位运算符

对两个整型数据实施**位运算**，即对两个整型数据对应的**位**进行运算得到一个新的整型数据。

1. “按位与”运算

“按位与”运算符“&”是双目运算符。

2. “按位或”运算

“按位或”运算符：“|”是二目运算符。

3. “按位非”运算

“按位非”运算符：“~”是单目运算符。

4. “按位异或”运算

“按位异或”运算符：“^”是二目运算符。

注意：参与运算的是两个整型数据、结果也是整型数据。

注意：运算法则是什么？例3-1

§ 3.1.8 instanceof 运算符

instanceof 运算符是二目运算符，左面的操作元是一个对象；右面是一个类。当左面的对象是右面的类或子类创建的对象时，该运算符运算的结果是true，否则是false。

§ 3.1.9 运算符综述

- ◆ Java的表达式就是用运算符连接起来的**符合Java规则**的式子。
- ◆ **运算符的优先级**决定了表达式中运算执行的先后顺序。
- ◆ 在编写程序时**尽量的使用括号（）运算符**来实现想要的运算次序，以免产生难以阅读或含糊不清的计算顺序。
- ◆ 运算符的结合性决定了**并列的相同级别运算符**的先后顺序。

§ 3.2 语句概述

Java里的语句可分为以下五类。

1. 方法调用语句。如： `System.out.println(" Hello");`
2. 表达式语句 表示式尾加上分号。比如赋值语句： `x=23;`
3. 复合语句

可以用{ }把一些语句括起来构成复合语句，如：

```
{    z=123+x;  
    System.out.println("How are you");  
}
```

4. 空语句。 一个分号也是一条语句，称做空语句。

5. 控制语句。控制语句分为条件分支语句、开关语句和循环语句。

6. **package**语句和 **import**语句。它们和类、对象有关，将在第4章讲解。

§ 3.3 if条件分支语句

条件分支语句按着语法格式可细分为三种形式，

if语句

if-else语句

if-else if-else 语句

§ 3.3.1 if语句

if语句是单条件分支语句，即根据一个条件来控制程序执行的流程。

if 语句的语法格式：

```
if(表达式){  
    若干语句  
}
```

例3-2

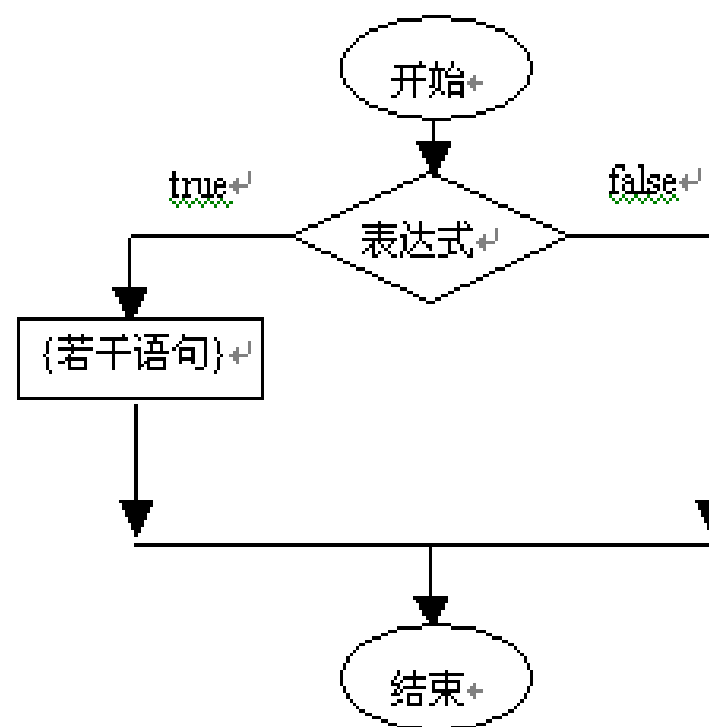


图 3.2 if 条件语句

§ 3.3.2 if-else语句

if-else 语句是双条件分支语句，即根据一个条件来控制程序执行的流程。**if-else 语句的语法格式：**

```
if(表达式) {  
    若干语句  
}  
else {  
    若干语句  
}
```

例3-3

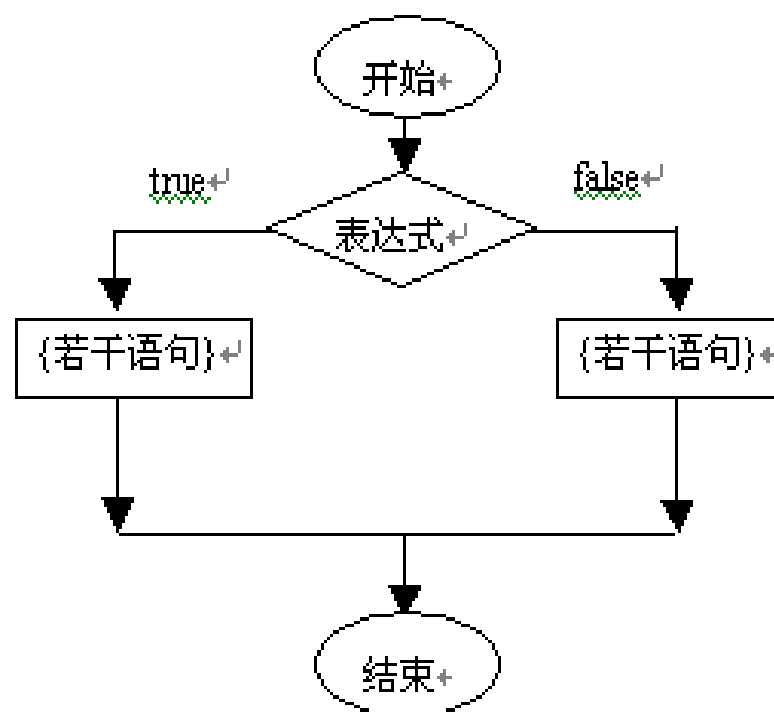


图 3.3· if-else 条件语句

§ 3.3.3 if-else if-else 语句

if-else if-else 语句是多条件分支语句，即根据多个条件来控制程序执行的流程。 **if-else if-else语句的语法格式：**

```
if(表达式) {
    若干语句
}
else if(表达式) {
    若干语句
}
... ..
else {
    若干语句
}
```

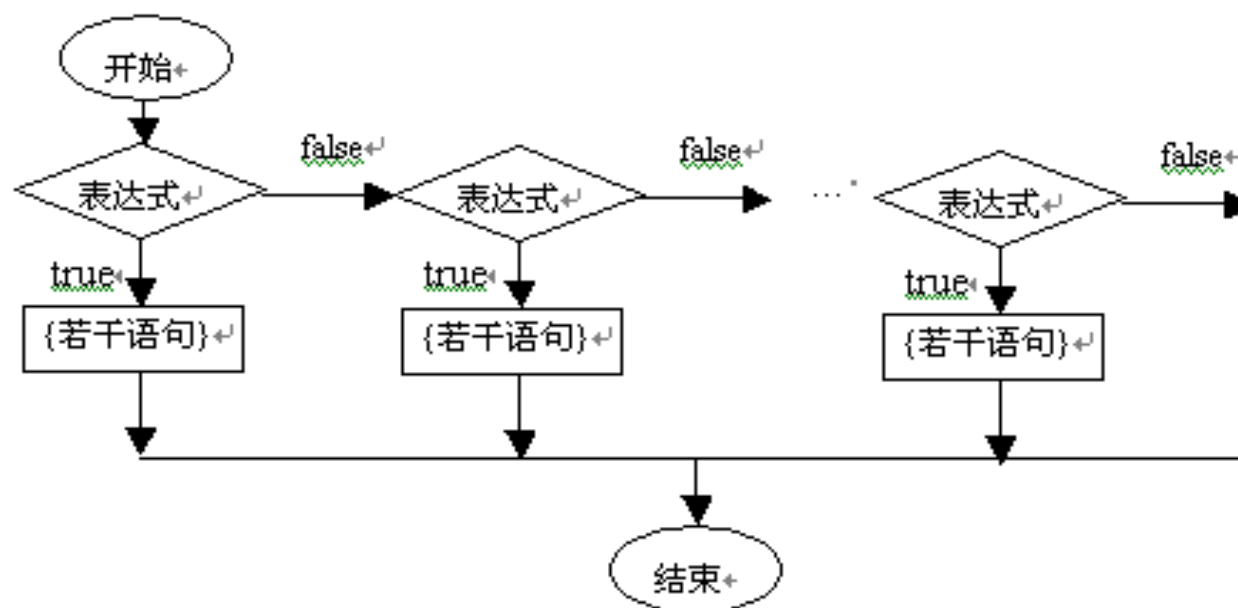


图 3.5 · if-else if-else 多条件语句

§ 3.4 switch开关语句

switch 语句是单条件多分支的开关语句，它的一般格式定义如下（其中break语句是可选的）：

```
switch(表达式)
{
    case 常量值1:
        若干个语句
        break;
    case 常量值2:
        若干个语句
        break;
    ...
    case 常量值n:
        若干个语句
        break;
    default:
        若干语句
}
```

2019-3-21

§ 3.5 循环语句

循环语句是根据条件，要求程序反复执行某些操作，直到程序“满意”为止。

§ 3.5.1 for循环语句

for语句的语法格式:

```
for (表达式1; 表达式2; 表达式3) {  
    若干语句  
}
```

for语句的执行规则是:

(1) 计算“表达式1”，完成必要的初始化工作。

(2) 判断“表达式2”的值，若“表达式2”的值为true，则进行(3)，否则进行(4)。

(3) 执行循环体，然后计算“表达式3”，以便改变循环条件，进行(2)。

(4) 结束for语句的执行。

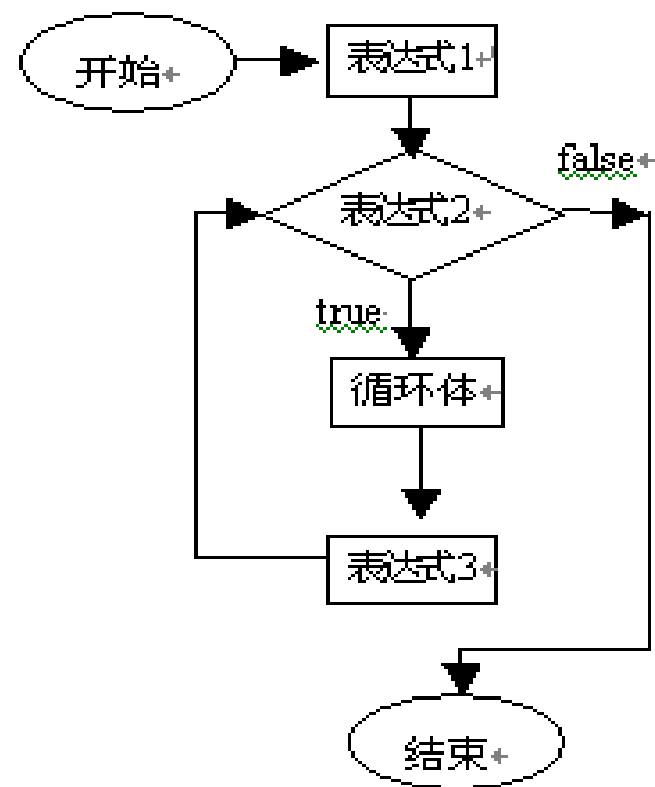


图 3.7 for 循环语句

§ 3.5.2 while 循环

while语句的语法格式:

```
while (表达式) {  
    若干语句  
}
```

while语句的执行规则是:

- (1) 计算表达式的值，如果该值是true时，就进行(2)，否则执行(3)。
- (2) 执行循环体，再进行(1)。
- (3) 结束while语句的执行。

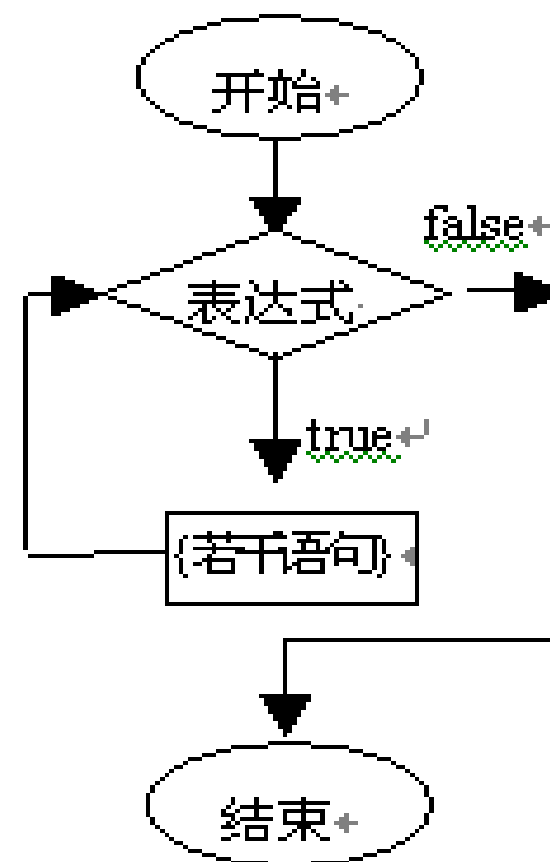


图 3.8 · while 循环语句+

§ 3.5.3 do-while循环

do-while语句的语法格式:

```
do {  
    若干语句  
} while(表达式);
```

do-while语句的执行规则是:

- (1) 执行循环体, 再进行 (2)。
- (2) 计算表达式的值, 如果该值是true时, 就进行 (1), 否则执行 (3)。
- (3) 结束while语句的执行。

例3-6

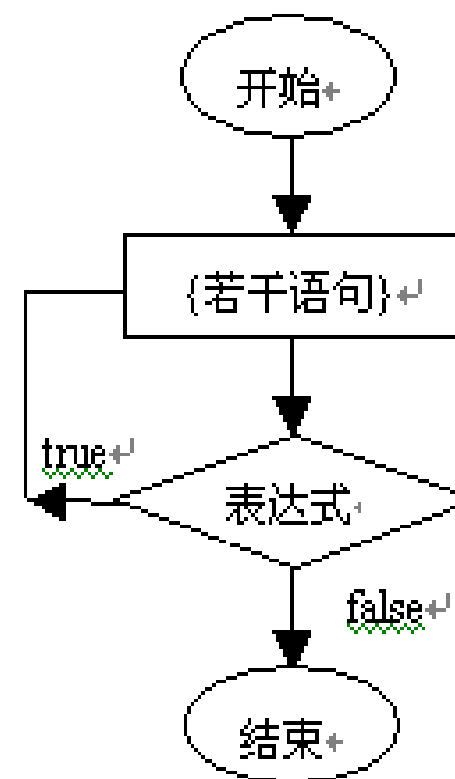


图 3.9 do-while 循环语句

§ 3.6 break和continue语句

break和continue语句是用关键break或continue加上分号构成的语句。

在循环体中可以使用break语句和continue语句。

如果在某次循环中执行了break语句，那么整个循环语句就结束。如果在某次循环中执行了continue语句，那么本次循环就结束，即不再执行本次循环中循环体中continue语句后面的语句，而转入进行下一次循环。

例3-7

2019-3-21

实践篇：

作业：

编写程序，求 $1+3+7+15+31+\dots+(2^{20}-1)$ 的值