

不积跬步，无以至千里
不积小河，无以成江海

昵图网 www.nipic.com 8V: I376077273

NO:20100504102026072060

第三章

运算符、表达式和语句



邹国兵

上海大学

计算机学院

运算符、表达式和语句

- ① 运算符与表达式.....•
- ② Java控制语句.....•

3.1 Java运算符和表达式

基本概念

- 运算符：用来表示各种不同运算的符号
- 操作数：参加运算的数据称为运算对象（操作数）
- 表达式：用运算符和括号将操作数连接起来的式子

3.1 Java运算符和表达式

运算符:

- ❑ 算术运算符: +, -, *, /, %, ++, --
- ❑ 比较运算符: >, <, >=, <=, ==, !=
- ❑ 逻辑运算符: &&, ||, !
- ❑ 位运算符: &, |, ~, ^
- ❑ 赋值运算符: =
- ❑ 扩展赋值运算符: +=, -=, *=, /=, %=

3.1 Java运算符和表达式

算术运算符(++，--)

```
public class Test {  
    public static void main(String arg[]) {  
        int i1 = 10, i2 = 20;  
        int i = (i2++);  
        System.out.print("i=" + i);  
        System.out.println(" i2=" + i2);  
        i = (++i2);  
        System.out.print("i=" + i);  
        System.out.println(" i2=" + i2);  
        i = (--i1);  
        System.out.print("i=" + i);  
        System.out.println(" i1=" + i1);  
        i = (i1--);  
        System.out.print("i=" + i);  
        System.out.println(" i1=" + i1);  
    }  
}
```

输出:
i=20 i2=21
i=22 i2=22
i=9 i1=9
i=9 i1=8

注意:

- ++(--)
- 在前时先运算再取值
- 在后时先取值再运算

Eclipse演示

3.1 Java运算符和表达式

逻辑运算符：&&、||、!

a	b	!a	a&& b	a b
true	true	false	true	true
true	false	false	false	true
false	true	true	false	true
false	false	true	false	false

3.1 Java运算符和表达式

□ 位运算符：&、|、~、^

举例：字符加密与解密。

```
package shu.ces.java.chap3;

public class Example3_1 {
    public static void main(String args[]) {
        char a1='+', a2='点', a3='进', a4='攻';
        char secret='A';
        a1=(char) (a1^secret);
        a2=(char) (a2^secret);
        a3=(char) (a3^secret);
        a4=(char) (a4^secret);
        System.out.println("密文:"+a1+a2+a3+a4);
        a1=(char) (a1^secret);
        a2=(char) (a2^secret);
        a3=(char) (a3^secret);
        a4=(char) (a4^secret);
        System.out.println("原文:"+a1+a2+a3+a4);
    }
}
```

$c=a^b, a=c^b$

^具有可逆性

Eclipse演示

3.1 Java运算符和表达式

注意：位运算符（&、|）和逻辑运算符（&&、||）的区别

- ✓ 利用&、|作位运算时，运算符左右两边的表达式都会执行，最后两表达式的结果再进行与、或运算；
- ✓ 而利用&&、||作逻辑运算时，如果只计算运算符左边的表达式即可确定与、或的结果，则右边的表达式将不会执行。

```
public class Test{
    public static void main(String args[]) {
        int i=1,j=2;
        boolean flag1 = (i>3)&&((i+j)>5);
        //第二个操作数将不再计算
        boolean flag2 = (i<2)||((i+j)<6);
        //第二个操作数将不再计算
    }
}
```


3.1 Java运算符和表达式

扩展赋值运算符

运算符	用法举例	等效的表达式
+=	a += b	a = a+b
-=	a -= b	a = a-b
*=	a *= b	a = a*b
/=	a /= b	a = a/b
%=	a %= b	a = a%b

3.2 Java语言流程控制

流程控制

- 分支语句: if-else if-else, switch
- 循环语句: for, while, do-while
- 与程序转移有关的其它语句: break, continue

3.2 Java语言流程控制

流程控制

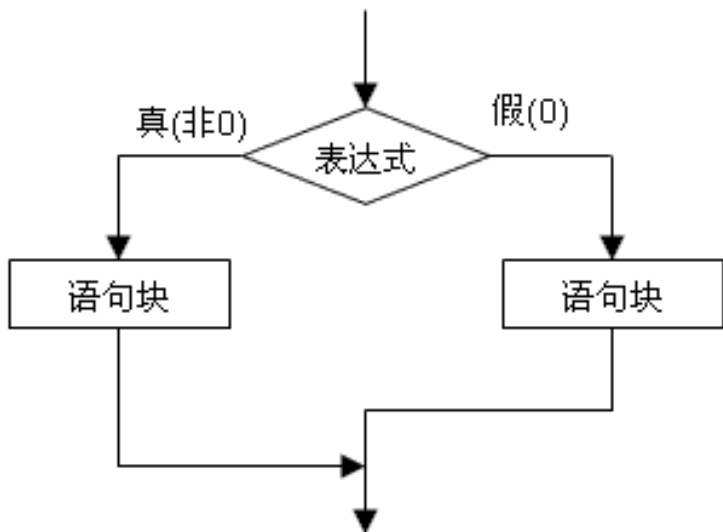
if语句

- ◇ if
- ◇ if .. else ..
- ◇ if .. else if ..
- ◇ if .. else if .. else if ..
- ◇ if .. else if .. else if .. else
- ◇ 只有一句需要执行的语句时，可以省略 {}

3.2 Java语言流程控制

if-else

Eclipse演示



if-else 语句的流程图

```
package shu.ces.java.chap3;

public class Example3_3 {
    public static void main(String args[]) {
        int math=65 ,english=85;
        if(math>60) {
            System.out.println("数学及格了");
        }
        else {
            System.out.println("数学不及格");
        }
        if(english>90) {
            System.out.println("英语是优");
        }
        else {
            System.out.println("英语不是优");
        }
        System.out.println("我在学习if-else语句");
    }
}
```

3.2 Java语言流程控制

switch语句（条件语句补充）

```
◇ switch() {  
    case xx :  
        ....  
    case xx :  
        ....  
    default:  
        ....  
}
```

◇ }

◇ 小心case穿透，推荐使用break语句

◇ 多个case可以合并到一起

◇ default可以省略，但不推荐省略

◇ switch

◇ java中switch语句只能探测int类型值

3.2 Java语言流程控制

中奖举例

```
package shu.ces.java.chap3;

import java.util.Scanner;
public class Example3_4{
    public static void main(String args[]) {
        int number;
        System.out.println("输入正整数(回车确定)");
        Scanner reader=new Scanner(System.in);
        number = reader.nextInt();
        switch(number) {
            case 9 :
            case 131 :|
            case 12 : System.out.println(number+"是三等奖");
                        break;
            case 209 :
            case 596 :
            case 27 : System.out.println(number+"是二等奖");
                        break;
            case 875 :
            case 316 :
            case 59 : System.out.println(number+"是一等奖");
                        break;
            default: System.out.println(number+"未中奖");
        }
    }
}
```

Eclipse演示

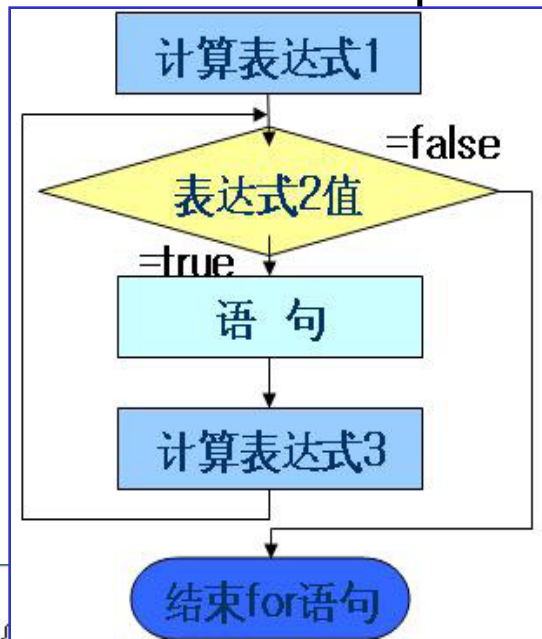
Eclipse演示上机Game

3.2 Java语言流程控制

for 循环语句

- for 语句为如下形式:
for(表达式1; 表达式2; 表达式3){ 语句; ... ; }
- 执行过程
首先计算表达式1, 接着执行表达式2, 若表达式2的值 = true, 则执行语句, 接着计算表达式3, 再判断表达式2的值; 依此重复下去, 直到表达式2的值=false
for语句中三个表达式都可以省略
JDK1.5新补充的for语法 (后面课程中补充, 数组)

```
public class Test {
    public static void main(String args[]) {
        long result = 0;
        long f = 1;
        for (int i = 1; i <= 10; i++) {
            f = f * i;
            result += f;
        }
        System.out.println("result=" + result);
    }
}
```



Eclipse演示

3.2 Java语言流程控制

流程控制

while & do while 语句

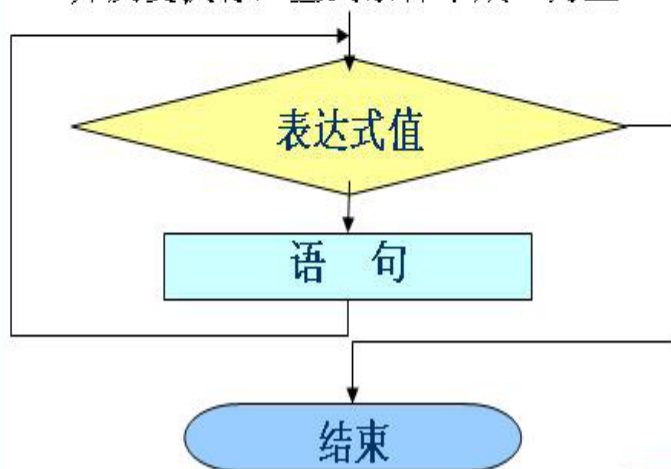
Example3_6演示

- ◇ while 语句为如下形式:

`while(逻辑表达式){ 语句; ... ;}`

- ◇ 执行过程

先判断逻辑表达式的值。若=true, 则执行其后面的语句, 然后再次判断条件并反复执行, 直到条件不成立为止



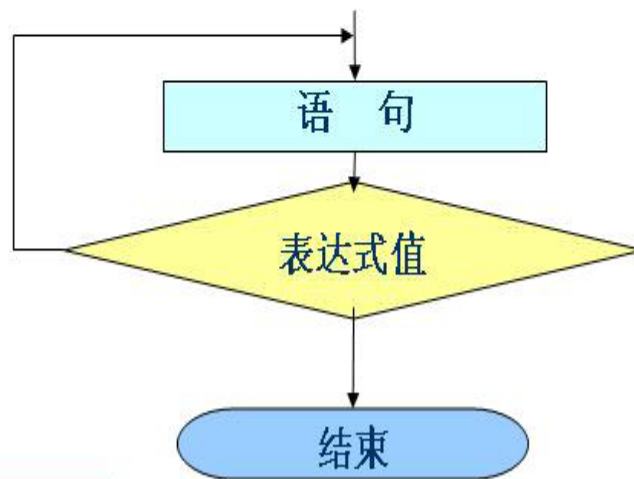
While 语句

- ◇ while 语句为如下形式:

`do { 语句; ... ; } while(逻辑表达式);`

- ◇ 执行过程

先执行语句, 再判断逻辑表达式的值, 若为true, 再执行语句, 否则结束循环。



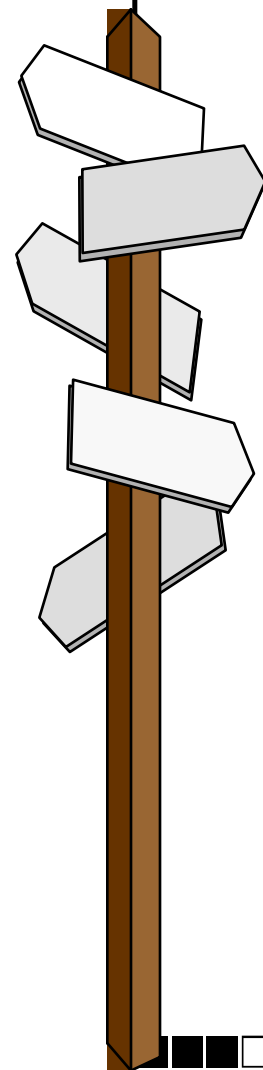
do-While 语句

TestWhile.java



思考:

break VS continue区别?



break & Continue 语句

- break 语句用于终止某个语句块的执行。用在循环语句体中，可以强行退出循环；例如：

```
public class Test {  
    public static void main(String  
        args[]) {  
        int stop = 4;  
        for (int i = 1; i <= 10; i++) {  
            //当i等于stop时，退出循环  
            if (i == stop) break;  
            System.out.println(" i= " + i);  
        }  
    }  
}
```

```
i = 1  
i = 2  
i = 3
```

- continue 语句用在循环语句体中，用于终止某次循环过程，跳过循环体中 continue 语句下面未执行的循环，开始下一次循环过程；例如：

```
public class Test {  
    public static void main(String  
        args[]) {  
        int skip = 4;  
        for (int i = 1; i <= 5; i++) {  
            //当i等于skip时，跳过当次循环  
            if (i == skip) continue;  
            System.out.println("i = " + i);  
        }  
    }  
}
```

```
i = 1  
i = 2  
i = 3  
i = 5
```

思考：

Eclipse演示



本章小结

运算符与表达式

算术、比较、逻辑、位运算符、赋值；

Java控制语句

条件、循环、break和
continue

