运算符和表达式

2018年10月28日　　11:20

1. 算术运算符和表达式
   a. 种类 + - * / %

      int x = 5, y = 4;

      int result1 = x/y;

      int result2 = x%y;

      ★ int x = 4; int y =0;

      int sum = x/y;

      编译没有错误，运行出错（异常， / by zero）

      ★ double r = 3.4 / 0;//这样写是对的

      system.out.prinrln(r);

      结果：Infinity（无穷）

   b. 运算法则：

      ★i. 如果表达式中有变量参与的运算，整个表达式的值最小数据类型为int，具体看最大的数的数据类型
2. 赋值运算符和表达式

   =、 +=、-=、*=、/=、%=

   a. =

      i. int x = 5; int y =5;

      赋值运算符的左边必须是变量

      赋值运算符右边可以是变量，字面量，表达式

      x = x+5;

      x = y;

      赋值方向：从右到左

      ii. 自动提升（byte -> short -> int (char) -> long -> float ->double）

      1）如果赋值运算符右边小于左边的类型，右边的值自动提升为大的数据类型

      iii. 强制转换

      1）如果赋值运算符右边大于左边的类型，那么使用强制类型转换y = (int)f;

   b. +=、-=、*=、/=、%=

      i. byte b2 = 2;

      b2 = b2 + 3;//error

      ★ii. 如果表达式中有变量参与的运算，整个表达式的值最小数据类型为int，具体看最大的数的数据类型

      ！ b2 += 3;//等同于 b2 = b2 + 3　right!

      因为上式相当于：b2 = (byte)(b2 + 3);//自动强制类型转换
3. 自增自减运算符和表达式 ++ --

   int x = 5;

   int r1 = ++x;//++(--)如果放到变量的前边，表示前置运算符

   int r2 = x++;//++(--)如果放到变量的后边，表示后置运算符

   a. 运算法则：

      i. 前置运算：变量先自增（自减），然后再把变量的值赋给r1（整个表达式的值）

      ii. 后置运算：先把变量的值赋给r2（整个表达式的值），然后变量本身再做自增（自减）

```
void test3() {
    int x = 5;
```

```
        int r1 = ++x;//x=6 r1=6
        System.out.println(r1+","+x);
        int r2 = x++;//x=7 r2=6
        System.out.println(r2+","+x);
        int r3 = x--;//x=6  r3=7
        System.out.println(r3+","+x);
        int r4 = --x;//x=5  r4=5
        System.out.println(r4+","+x);
}
```

4. 比较运算符和表达式
   a. 种类：>、<、>=、<=、!=、==
   b. 使用比较运算符连接而成的表达式，表达式返回值类型是boolean

   int x=1;

   int y=2;

   boolean b1 = x>y;//false

```
void test4() {
        int x = 1;
        int y = 1;
        boolean b1 = x == y;
        System.out.println(b1);//true
        }
```

5. 逻辑运算符和表达式
   a. 种类： &&(&)、||(|)、！
   b. &&表达式：表达式的左边和右边必须是能够返回boolean类型的值,表达式的数据类型为：boolean 一假则假

   int x =1;

   iint y = 2;

   boolean b1 = (x>y && x<9);

```
void test5() {
        int x =1;
        int y = 2;
        boolean b = false;
        boolean b1 = (x>y && x<9);//b1 = false
        boolean b2 =true && x>y;//b2 = false
        boolean b3 = x>y && b;//b3 = false
      //boolean b4 = x++ && x>9;//error  x++返回类型不是boolean类型
        }
```

      i. &&运算法则：只有两边的值都为true是，则整个表达式的值为true，否则为false
      ii. &&短路问题：左边的表达式为false时，&&运算处于短路状态（右边的表达式不再做运算）

```
x = 1;y = 2;
boolean b5 = true && x++<--y;//x=2  y=1 b5 =false
x = 1;y = 2;
boolean b6 = ++x==++y && y--<x;//x=2 y=3 b6=false
```

   c. &运算符：

```
void test6() {
        int x = 1;
        int y = 2;
        //&作为二进制的运算符使用
        //可以表示java中的二进制的运算
        //按位与运算为：两位全为1，结果为1，即1&1=1，1&0=0，0&1=0，0&0=0。
        int r1 = x&y;
        //&作为逻辑运算符使用（逻辑与）
        //不同的地方，&作逻辑运算没有短路现象
        boolean r2 = x>y & y++<4;
        System.out.println("r2="+r2+" , "+"y="+y);//r2=false , y=3
        }
```

   d. ||表达式：表达式的左边和右边必须是能够返回boolean类型的值,表达式的数据类型为：boolean 一真则真

```
void test7() {
```

```
    int x =1;
    int y = 2;
    boolean b = false;
    boolean b1 = (x>y || x<9);//b1 = true
    boolean b2 =true || x>y;//b2 = true
    boolean b3 = x>y || b;//b3 = false
  //boolean b4 = x++ && x>9;//error  x++返回类型不是boolean类型
    }
```

i. ||运算法则：只要有一边的值为true是，则整个表达式的值为true，否则为false

ii. ||短路问题：左边的表达式为true时，||运算处于短路状态（右边的表达式不再做运算）

```
x = 1;y = 2;
boolean b5 = true || x++<--y;//x=1  y=2 b5 =true
x = 1;y = 2;
boolean b6 = ++x==++y || y--<x;//x=2 y=2 b6=false
```

e. | 运算符：

```
void test8() {
    int x = 1;
    int y = 2;
    //|作为二进制的运算符使用
    //可以表示Java中的二进制的运算
    //两位只要有一位为1，结果则为1，即1|1=1，1|0=1，0|1=1，0|0=0。
        特殊用法：
        （1）与0相或可保留原值。
        （2）与1相或可将对应位置1。例如，将X=10100000的低四位置1，使X | 00001111 = 10101111即可。
    int r1 = x|y;
    //|作为逻辑运算符使用（逻辑与）
    //不同的地方，|作逻辑运算没有短路现象
    boolean r2 = x>y | y++<4;
    System.out.println("r2="+r2+" , "+"y="+y);//r2=true , y=3
    }
```

f. ！逻辑非（单目运算符）

```
boolean b = true;
boolean b1 = !true;
boolean b2 = !(3>5);
boolean b3 = !b;
```

i. 运算法则：取反

```
void test9() {
    boolean b = true;
    boolean b1 = !true;//b1 = false
    boolean b2 = !(3>5);//b2 = true
    boolean b3 = !b;//b3 = false
    }
```

6. 三目运算符和表达式（等价于：if --- else）

a. 格式：a ? b : c;

表示如果a为true，那么表达式的值为b，否则为c

a必须为返回boolean类型的变量，字面量，表达式

```
String str = true ? "对了" : "错了";
double d = 4 > 5 ? 4 : 4.5;//前面的变量类型取决于后面的最大数据类型
```

```
void test1() {
    boolean b = true;
    String str = true?"对了":"错了";
    double d = b?4:4.5;
    int i = 4<7?4:7;
    System.out.println(str);//对了
```

```
System.out.println(d);//4.0
System.out.println(i);//4
}
```

7. 连接符 （+）
   a. 前面是字符串->""，后面的所有+号当做连接符来用,如果变成运算符使用，后面加（）

```
void test2() {
System.out.println(3+4+" ");//7
System.out.println(" " + 3+4);// 34
System.out.println(3+4+" "+3+4);//7 34
System.out.println(" " + (3+4));// 7
}
```

8. 控制台输入

```
void test3() {
Scanner sc = new Scanner(System.in);

//next方法表示输入字符串
System.out.print("请输入姓名：");
String str = sc.next();

//nextInt 方法表示输入int
System.out.print("请输入i的值：");
int i = sc.nextInt();

//nextDouble方法表示输入doble类型的数据
System.out.print("请输入d:");
double d = sc.nextDouble();

System.out.println(str+","+i+","+d);
}
```