

JAVA Learning class 4

P.R.O. Club

JAVA运算符

按照操作数的多少来分类

单目	双目	三目
<code>a++</code>	<code>a+b</code>	<code>(a>b) ? x : y</code>

按照运算符的功能来进行分类

算术运算

赋值运算

关系运算 (比较)

逻辑运算

位bit运算

算术运算

+ - * / %(取余)

++ (自增) --(自减)

int x = 5; x/2等于几 x%2等于几

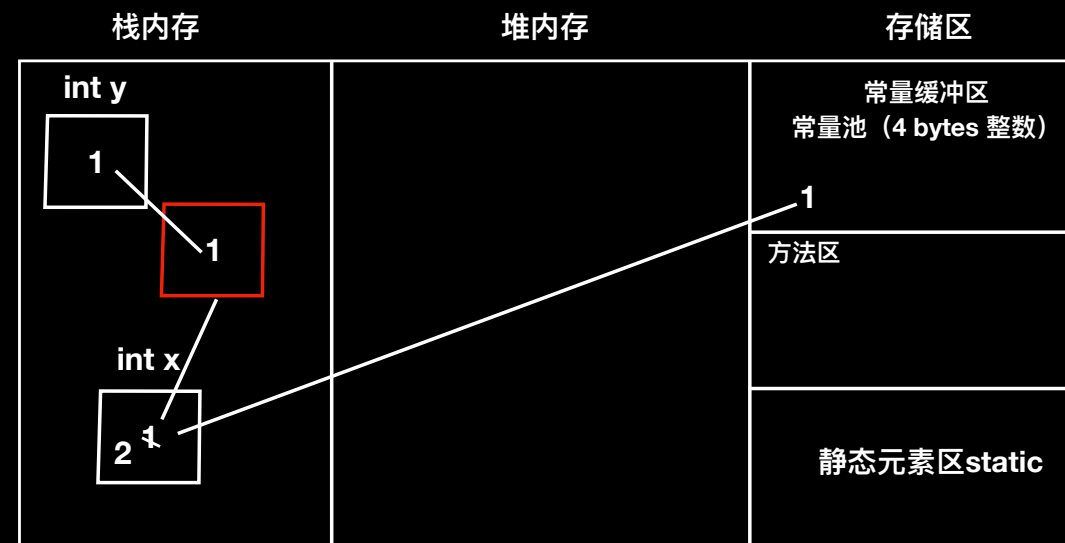
int x = 1;

x = x + 1; // x++ x+=1 ++x

```
int x = 1;  
int y = x++;  
x==?   y==?  
x==2   y==1
```

`int x = 1;`
`int y = x++;`

x 在做计算的时候会产生
副本空间



++在变量前, 先自增后备份
++在变量后, 先备份后自增
赋值赋的是备份中的值

```
int x = 1;  
int y = ++x;  
x==?  y==?  
x==2  y==2
```



```
int a = 1;
```

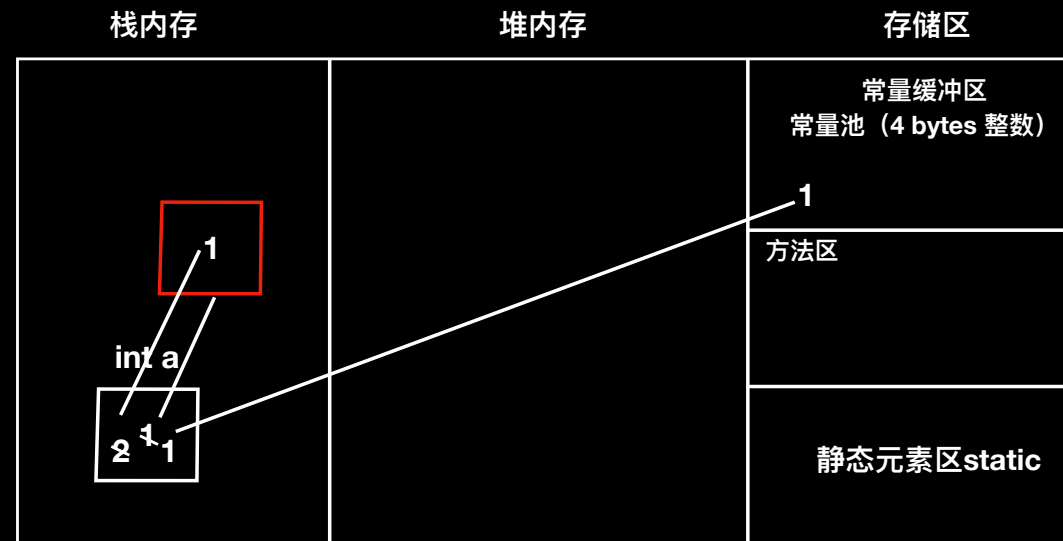
```
a = a++;
```

```
a==?
```

```
a==1
```

`int a = 1;`
`int a = a++;`

a 在做计算的时候会产生
副本空间



++在变量前, 先自增后备份
++在变量后, 先备份后自增
赋值赋的是备份中的值

```
int a = 1;  
int a = a++;  
int a = a++;  
int a = a++;  
int a = a++;  
int a = a++;  
a==1
```

```
int m=1;  
int n=2;
```

```
int sum = m++ + ++n - n - - - -m + n - - - -m;
```

m==? n==? sum==?

m==0 n==1 sum==2

赋值运算

= 赋值符号 将=右边的内容存到=左边的变量空间内

`+=` `-=` `*=` `/=` `%=`

`int x = 1;`

`x+=2;` //x增了2 (3)

`int x = 2;`

`x*=3` //x乘3 (6)

关系运算（结果是`true / false`）

> >= < <= != ==

==比较符号 比较==前面的内容和==后面的内容是否一致

3>4 5<4 3>=2 2!=2 4==4

逻辑运算

& 逻辑与（并且） | 逻辑或 ^ 逻辑异或 ! 逻辑非
&& 短路与 || 短路或

1. 逻辑运算符前后连接的应该是两个boolean结果

- a) &逻辑与 前后两个条件必须都是true 最终才会为true
- b) | 逻辑或 前后只要有一个是true 就是true
- c) ^逻辑异或 前后两个结果不一致 就是true
- d) ! 逻辑非 取反，将原来的结果取反

位(*bit*)运算

& 按位与 | 按位或 ^ 按位异或 ~ 按位取反

<< 按位左位移 >> 按位右位移

位(bit)运算

- 1.将3和5转化成二进制表示形式
- 2.竖着按照对应位置进行& | ^ 计算
- 3.将计算后的二进制结果转化为十进制

3 & 5 = ?

00000011	(3)
00000101	(5)
00000001	(1)

3 | 5 = ?

00000011	(3)
00000101	(5)
00000111	(7)

位(bit)运算

- 1.将3和5转化成二进制表示形式
- 2.竖着按照对应位置进行& | ^ 计算
- 3.将计算后的二进制结果转化为十进制

$$3 \wedge 5 = ?$$

00000011 (3)

00000101 (5)

00000110 (6)

$$\sim 8 = ?$$

-9

-(x+1)

<<按位左位移 $6 \ll 1 = ?$ 12

00000000 00000000 00000000 00000110
00000000 00000000 00000000 00000000

>>按位右位移 $6 \gg 2 = ?$ 1

00000000 00000000 00000000 00000110
00000000 00000000 00000000 00000000