

第二章 数据类型和运算符



基本类型	大小	类型取值
<u>boolean</u>		true, false
byte	1字节8位有符号整数	-128 ~ +127
char	2字节16位Unicode字符	Unicode 0 ~ Unicode $2^{16}-1$
short	2字节16位有符号整数	-32768 (-2^{15}) ~ +32767 ($+2^{15}-1$)
int	4字节32位有符号整数	-2147483648 (-2^{31}) ~ +2147483647 ($2^{31}-1$)
long	8字节64位有符号整数	-2^{63} ~ $+2^{63}-1$
float	4字节32位浮点数	1.4E-45 ~ 3.4E+38, -1.4E-45 ~ -3.4E+38
double	8字节64位浮点数	4.9E-324 ~ 1.7E+308, -4.9E-324 ~ -1.7E+308



byte	1字节8位有符号整数	-128 ~ + 127
short	2字节16位有符号整数	-32768 (-2^{15}) ~ + 32767 ($+2^{15}-1$)
int	4字节32位有符号整数	-2147483648 (-2^{31}) ~ + 2147483647 ($2^{31}-1$)
long	8字节64位有符号整数	-2^{63} ~ + $2^{63}-1$



float	4字节32位浮点数	$1.4\text{E}-45 \sim 3.4\text{E}+38$, $-1.4\text{E}-45 \sim -3.4\text{E}+38$
double	8字节64位浮点数	$4.9\text{E}-324 \sim 1.7\text{E}+308$, $-4.9\text{E}-324 \sim -1.7\text{E}+308$

□ 科学计数法

□ 注意：

➤ 浮点类型注意：

- ✓ 不要在数量级相差很大的浮点类型之间进行运算。
- ✓ 避免进行等量的判断。



char	2字节16位 Unicode 字符	Unicode 0 ~ Unicode $2^{16}-1$
-------------	--------------------------	--------------------------------

□ Unicode编码

boolean



<u>boolean</u>		true, false
-----------------------	--	--------------------



- 语法

- 连接 +

- ## 自动类型转换

byte – short
– int – long – float – double
char



□ 强制类型转换

语法格式为：

(数据类型) 操作数



操作数1类型

byte、short、char

byte、short、char、int

byte、short、char、int、long

byte、short、char、int、long、float

操作数2类型 运算后的类型

int int

long long

float float

double double



□ 实现一个数字加密器，加密规则是：

$$\text{加密结果} = (\text{整数} * 10 + 5) / 2 + 3.14159$$

加密结果仍为一整数，并将结果输出



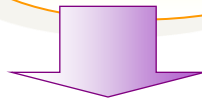
- 括号运算符 ()
- 算数运算符 + - * / %
- 位运算 & | ~
- 关系运算符 > >= < <= == !=
- 逻辑运算符 && || !
- 条件运算符 ?:
- 赋值运算符 = +=



操作数

运算符

操作数



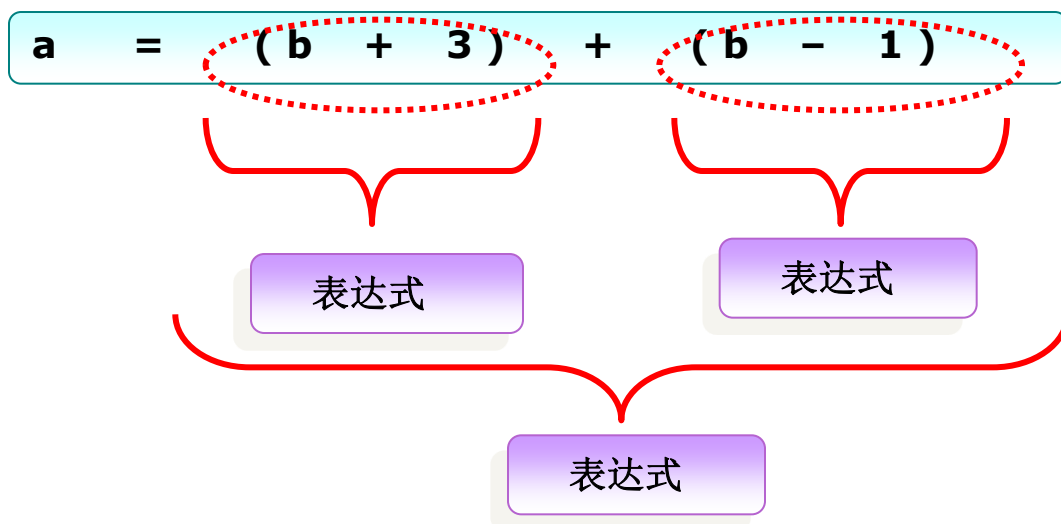
+

-

*

/

%





□ 自增运算符与自减运算符 (++ , --)

➤ 后缀表示法: $x++$

➤ 前缀表示法: $++x$



□ 说出结果？

```
int i = 5;  
int j = 5;  
int m = 5;  
int n = 5;
```

```
i++;  
j = j + 1;  
m--;  
n = n - 1;
```

```
System.out.println(i);  
System.out.println(i++);  
System.out.println(++i);
```




□ 关系运算符

>

<

>=

<=

==

!=



□ 逻辑运算符

➤ && || !

表达式一	表达式二	op1&&op2	op1 op2	!op1
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

说出运算结果：

result = (5>9) && (7==7)

result = (5>9) || (7==7)

result = !((5>9) && (7==7))

result = !((5>9) || (7==7))



□ 扩展赋值运算符

`+=`

`-=`

`*=`

`/=`

`%=`



□ 分析代码：

代码段一：

```
short n3 = 5;  
n3  = n3 + 5;  
System.out.println(n3);
```

代码段二：

```
short n3 = 5;  
n3 += 5;  
System.out.println(n3);
```



□ 语法

表达式一 ? 表达式二 : 表达式三



&

|

^

~

<<

>>

>>>



运算符的优先级

□ 例：

boolean

```
r=true||false||false&&false;
```

```
r = true
```

等价于：

boolean

```
r=true||false||(false&&false);
```



常用运算符优先级：

()

! ++ --

- 算术运算符
- 关系运算符
- 位 运算符
- 逻辑运算符
- 条件
- 赋值



□ 步骤：

- ① `import java.util.Scanner;`
- ② `Scanner superman= new Scanner(System.in);`
- ③ `superman.nextInt();`