Chapter 4. 运算符. 周日 8.11

"/"号的使用

↓

println(10/4) → 输出2.

println(10.0/4) → 2.5

double d = 10/4 → 2.0

%取余

10%3 = 1.

-10%3 = -1

10%-3 = 1

-10%-3 = -1

%本质 → a%b = a - a/b*b

↓

int(a)

++的使用

int i=10

单独{ i++ → i=i+1 ; ++i → i=i+1 } 先增,后进

表达式{ int k=++i → i=i+1, k=i ; int k=i++ → k=i, i=i+1 } 先进,后增

面试题:

int i=1 ; i=i++ ; ① temp=i ; ② i=i+1 ; ③ i=temp=1.

int i=1 ; i=++i ; ① i=i+1 ; ② temp=i ; ③ i=temp

4.3 关系运算符. → 用来比较运算 P69

*结果均为boolean型.

== 相等 ; != 不等

4.4 逻辑运算 P70-P75

| → 或(全假为假) ; & → 与(全真为真) ; ^ → 异或

|| → 短路或 → 若条件1为真则条件2则不判断

&& → 短路与 → 若条件1为false后面条件不判断

! → 取反

a^b → a与b不同,则取true.

常用,效率高.

4.5 赋值运算 P76-P77

"=" 赋值.

a+=b → a=a+b

同理, -=, *=, /=, %=, ...

赋值运算特点:

* 运算顺序右→左, int num=a+b

* 左边为变量,右边为变量、表达式或常量.

* 复合赋值时会进行类型转换

byte b=2

b+=3; // 等价于 b=(byte)(b+3)

b=b+3 → 为整型,需转型

4.6 三元运算符 P79－80

"三元"构成：条件表达式① ? 表达式②：表达式③；

① = { true → 执行②
　　 { false → 执行③

例. int n1 = 55;
　　int n2 = 33;
　　int max = n1 > n2 ? n1 : n2;

★4.7 运算符优先级 P81

梳理：

① ( ) 行，等
② 单目运行 ++，--
③ 算术运算符 + - * %
④ 位移运算符.
⑤ 比较运算符
⑥ 逻辑
⑦ 三元
⑧ 赋值

4.8 标识符命名规则与规范 P82.－P85

↓

变量、方法、类，等需自主命名的字符序列.

命名规则 {
- 由26字母，0-9，_ or $组成.
- 数字不开头 int 3ab X
- 不含关键字
- 严格区分大小写
- 不含空格.

}

命名规范
- 包名：均小写
- 类名、接口名：Xxx Yyy Zzz（大驼峰）
- 变量名、方法名：xxx Yyy Zzz（小驼峰）
- 常量名：大写字母，下划线连接 XXX_YYY

4.12. 进制. P87 – 97.

二进制：以0b or 0B开头. → 0b1010

八进制：以0开头. → 0108

十六进制：A(10) – F(15)，以0x or 0X开头 → 0X1010

进制转换：

二 → 十：$0b1011 = 1\times2^0 + 1\times2^1 + 0\times2^2 + 1\times2^3 = 1+2+0+8 = 11$（3210）

八 → 十：$0234 = 4\times8^0 + 3\times8^1 + 2\times8^2 = 4+24+128 = 156$（210）

十六 → 十：$0X23A = 10\times16^0 + 3\times16^1 + 2\times16^2 = 10+48+512 = 570$（210）

十 → 二

2|134 0
2|67 1
2|33 0
2|16 0
2|8 0
2|4
1

→ 0b10000110

十 → 八

8|131 3
8|16 0
2

→ 0203

十 → 十六

16|237 13→D
14→E

→ 0XED

二 → 八.（每三位对应八进制的一位）

= 0b11010101

八 3 2 5

二 → 十六.（每四位对应一位）

二 → 0b11010101

13 5

13→D

∴ 十六 → 0x D5

八 → 二（每1位专变成二进制的3位）

八 → 0 2 3 7

010 011 111

二 → 0b10011111

十六 → 二（每1位转成二进制的4位）

十六 → 0x 2 3 B

二 → 0b. 0010 0011 1011

8421

4.27 原码，反码，补码. (重点!!!) P19.

相关规则. (背!!!)

1. 二进制最高位是符号位：0表正，1表负，(0→0，1→-) 最左边位

2. 正数 原，补，反码均一样 (三码合一)

3. 负数反码 = 原码符号位不变，其它位取反(0→1，1→0)

4. 负数补码 = 反码+1，负数反码 = 补码 -1

5. 0的反码，补码都为0

6. java中的数，都是有符号的

7. 计算机运算时，均以补码方式运算，重要!!

8. 看运算结果时，看原码!(重点)

为什么设置补码？
补码将正、负数统一起来了

4.28 位运算符 P100 学进制是为了学位运算

位与&：两位全为1，结果为1.否则为0.

位或|：两位有1为1，结果为1.否则为0.

位异或^：两位分别为0和1，则为1.

位取反~：0→1，1→0.

算术右移 >>：低位溢出，符号不变，符号位补溢出位.

```
int a = 1>>2; // 1 → 00000001 → 00000000 本质→ 1/2/2=0
```

算术左移：符号位不变，低位补0.

```
int c = 1<<2; // 1 → 00000001 → 00000100 本质→ 1*2*2.
```