Java语言基础

【Day03】

Java 运算符及表达式

operator expression

算术运算符

- 加（+）、减（-）、乘（*）、除（/）、 取余（%）
- 整数相除，只能取整数部分，小数部分被舍弃。
- 整数运算时，0不能做除数；浮点运算时，0.0可以，但结果是无穷。
- 算数运算看起来简单，其实有难度。

```
int a = 121;
int b = 15;
int c = a / b;
```

结果是8，整数除法会取整

```
int a = 5;
System.out.println(a % 2);
```

计算a除以2取余数，结果是1。

项目案例

- 提示用户输入整数类型的秒数，输出x小时x分x秒。
  如：输入7199，输出1小时59分59秒。

字符串连接运算符

- + 可以实现字符串的连接。同时可以实现字符串与其他数据类型“相连”。

```
int a = 100;
String msg = "a=" + a;
System.out.println(msg);结果为：a=100

msg = "" + 100 + 200;
System.out.println(msg);结果为：100200

msg = 100 + 200 + "";
System.out.println(msg);结果为：300
```

关系运算符

- Java关系运算符用于判断数据之间的大小关系。
  > 大于　< 小于　>= 大于等于　<= 小于等于　== 等于　!= 不等于
- 关系表达式的结果为boolean类型（true或false）

```
int a = 100;
boolean b1 = a > 100;          false

boolean b2 = (a + 1) >= 100;   true
```

自增减运算符

- 自增（++）、自减（--）
- 只能用于变量，常数不可以

```
int i = 10;
int m = i++;
System.out.println(m + "," + i);

int n = ++i;
System.out.println(n + "," + i);
```

10,11

12,12

逻辑运算符

- 逻辑运算符的操作数均为boolean表达式。
- 逻辑运算符：&& 与 || 或 ！ 非

b1	b2	b1 && b2	b1 || b2	!b1
false	false	false	false	true
false	true	false	true	
true	false	false	true	false
true	true	true	true	

逻辑运算符

```
double income = 4000;
boolean b1 = (income >= 3500) && (income < 5000);
System.out.println(b1); true

int num = 30;
boolean b2 = num < 0 || num > 100;
System.out.println(b2); false
```

“&&”、“||” 具备“短路”的特性：如果通过第一个表达式的值即可得出最后的结果，则不计算第二个表达式。

条件运算符

- 条件运算符又称“三目”运算符，其结构为：
  条件表达式 ？ 表达式1： 表达式2
- 先计算boolean表达式的值，如果为true，则整个表达式的值为表达式1的值；如果为false，则整个表达式的值为表达式2的值。

```
int a = 100, b = 200;
int flag = a > b ? 1 : -1;
```

flag的值为-1

```
int a = 100, b = 200;
int max = a >= b ? a : b;
```

max的值为200

赋值运算符

- = 称为赋值运算符，用于对变量赋值。
  赋值表达式本身也有值，其本身之值即为所赋之值。
- 可以使用扩展赋值表达式（+=、-=、*=、/=…）。

```
int num = 91;
int index = num % 5; 1
```

```
int a, b, c;
a = b = c = 100;
```

整个表达式的值为100

```
int sum = 100;
sum += 10;
```

相当于 sum = sum + 10

常用运算符的优先级

- 常用运算符的优先级如下：

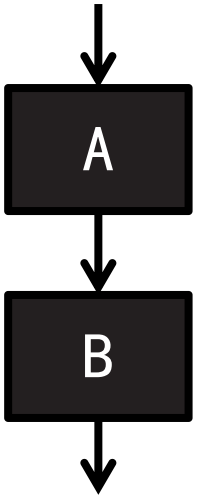
运算符	结合性
[] . ( ) (方法调用)	从左向右
! ~ ++ -- +(一元运算) -(一元运算)	从右向左
* / %	从左向右
+ -	从左向右
<< >> >>>	从左向右
< <= > >= instanceof	从左向右
== !=	从左向右
&	从左向右
^	从左向右
|	从左向右
&&	从左向右
||	从左向右
?:	从右向左
=	从右向左

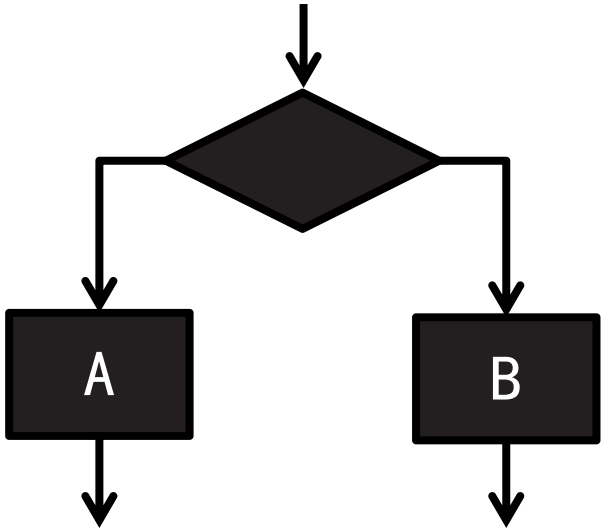
Java 分支结构

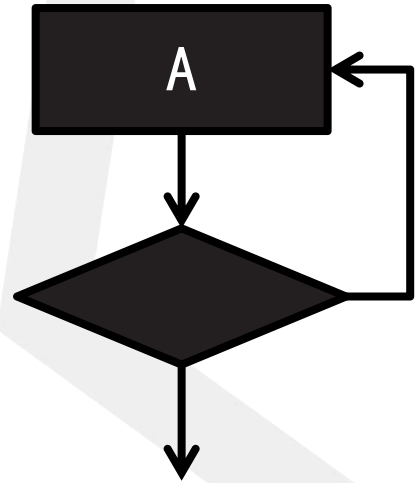
Branch Structure

三种程序结构

- 任何复杂的程序逻辑都可以通过 顺序、分支、循环 三种基本的程序结构实现







if分支结构

- if（条件表达式） {
  语句块;
  }
- 当条件表达式为true时，执行语句块，否则不执行

```
if(max<b)   max = b;
```

当语句块只有一条语句时也省略语句块

项目案例

- 提示用户输入4个整数并记录到a、b、c、d中，找出4个整数中的最大值打印出来。

if-else分支结构

- if（条件表达式） {
  语句块1;
  } else {
  语句块 2
  }
- 当条件表达式为true时，执行语句块1，否则执行语句块2

```
if(score>=60) {
    System.out.println("Pass");
} else {
    System.out.println("Fail");
}
```

if-else if-else分支结构

- if（条件表达式1） {
  语句块1;
  } else if（条件表达式2）{
  语句块 2;
  }
  … …
  else{
  语句块n;
  }
- 当条件表达式为true时，执行对应的语句块，否则执行语句块n

项目案例

- 判断用户输入的整数是正数、负数还是零并打印。

项目案例

- 根据用户输入的薪水计算个人所得税并打印出来，其中个税起征点为：5000元，具体规则如下：

全月应纳税所得额	税率	速算扣除数（元）
全月应纳税额不超过3000元（8000）	3%	0
全月应纳税额超过3000元至12000元（17000）	10%	210
全月应纳税额超过12000元至25000元（30000）	20%	1410
全月应纳税额超过25000元至35000元（40000）	25%	2660
全月应纳税额超过35000元至55000元（60000）	30%	4410
全月应纳税额超过55000元至80000元（85000）	35%	7160
全月应纳税额超过80000元	45%	15160

项目案例

- 出租车计费方式：由里程钱数和等候时间钱数相加得出。
- 里程数前3公里10元，超过3公里到15公里部分每公里2元，15公里以上部分每公里3元。
- 等候时间每2分半1元，不足部分不要钱。
- 输入公里数和等候秒数，输出车费。

比如：16公里，等候290秒，车费 = 10 + （15-3）*2 + （16-15）*3 + 1 = 38

总结与答疑

技术资料

IT兄弟连

ITXDL.CN

变态严管　　让学习成为一种习惯