

# Java 语言基础

DAY03

# 逻辑运算

- 逻辑运算的操作数均为BOOLEAN表达式。
- JAVA提供如下逻辑运算符：“&&”（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（续）

```
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
```

```
boolean b3 = (str != null) && (str.length()  
> 0);
```

“&&”、“||”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：

**BOOLEAN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;
```

# 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 = 90;  
int index = num % 5;
```

```
int a, b, c;  
a = b = c = 100; “c=100” 整个表达式的值为100
```

```
int sum = 100;  
sum += 10; 相当于 “sum = sum + 10”
```

# 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的“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);
```

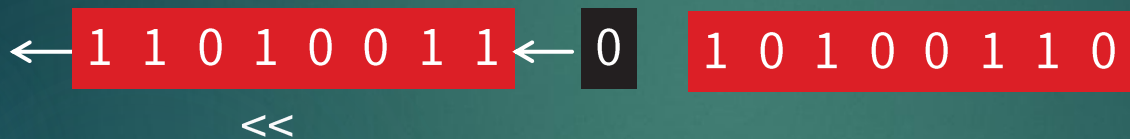
结果为: 100200

```
msg = 100 + 200 + "";  
System.out.println(msg);
```

结果为: 300

# 移位运算符

- 左移 (<<)、算数右移 (>>)、逻辑右移 (>>>)。
- 移位运算针对于整数做二进制位移动。



# 移位运算符（续）

```
int a = 8;
a = a >> 1;
System.out.println(a); 结果为4，相当于除2。
a = a << 2;
System.out.println(a); 结果为16，相当于乘4。
int b = -8;
System.out.println(b >> 2); 结果为-2，相当于除4。
System.out.println(b >>> 2); 结果为1073741822，相当于将原数
                                看成无符号数 4294967288
                                （ ffffffff ），再除4。
int c = 8;
System.out.println(c << 33); 结果为16，移位运算时，移位值将对
                                32取模。此处的移位值为 $33\%32 = 1$ 
```



# 位运算符

按位逻辑运算（& 按位与、| 按位或、  
^ 按位异或、~ 按位非），对整数类型  
（包括char）按照二进制位逐位进行  
逻辑运算。

操作数1	操作数2	&		^
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

# 位运算符(续)

```
int b1 = 1;           //00000001
int b2 = 2;           //00000010
int b3 = 4;           //00000100
System.out.println(b1 | b2 | b3); //输出7 等于合并为00000111
```

```
int b4 = 257;         // 1 00000001
int b5 = 0xff & b4;   // 消去高位, 保留后8位 00000001
System.out.println(b5); //输出1
```

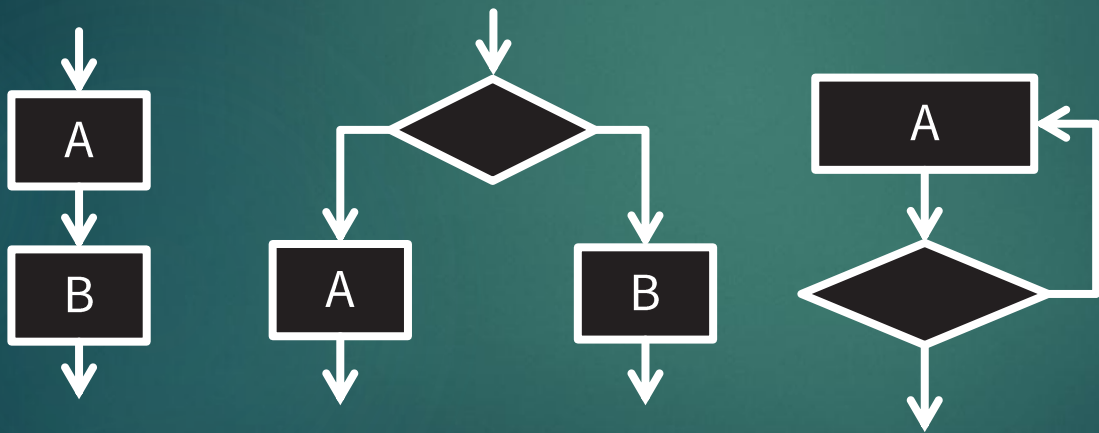


# 分支 结构

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结构

- `if (关系表达式) {`  
    `// 语句块`  
`}`
- 当关系表达式为TRUE时，执行语句块，否则不执行。

```
if (a > b) {  
    temp = a;  
    a = b;  
    b = temp;  
}
```

当a大于b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。

# 求四个整数中的最大值。

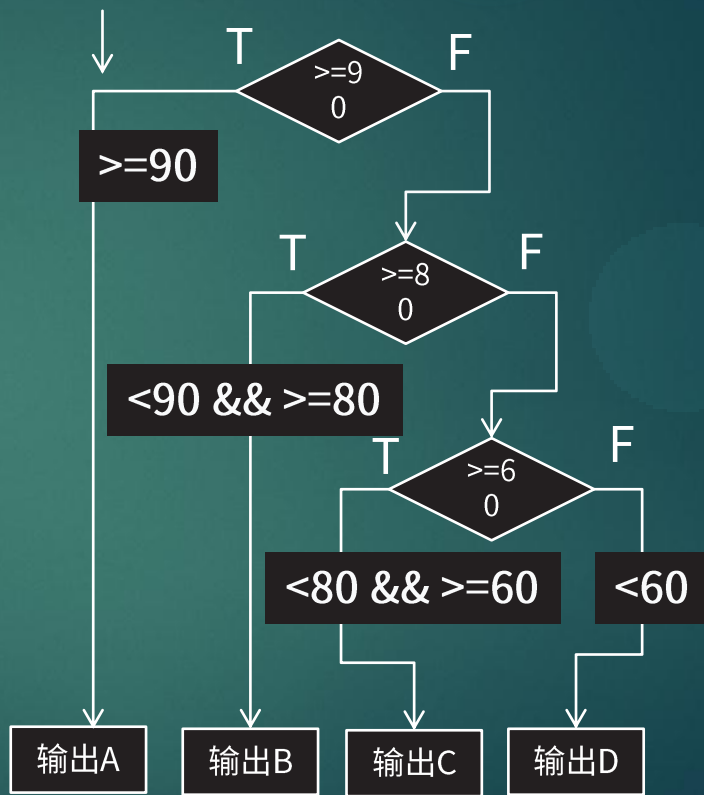
将A、B、C、D四个变量中的最大值输出。

# if-else结构

- `if (关系表达式) {`  
    `// 语句块 1`  
`} else {`  
    `// 语句块 2`  
`}`
- 当关系表达式为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嵌套

```
if(score>=90) {  
    System.out.println("A");  
} else if(score>=80){  
    System.out.println("B");  
} else if(score>=60) {  
    System.out.println("C");  
} else {  
    System.out.println("D");  
}
```





判断一个数是正数、零还是负数。

# 个人所得税计算器。

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1500元(5000)	3%	0
全月应纳税额超过1500元至4500元(8000)	10%	105
全月应纳税额超过4500元至9000元(12500)	20%	555
全月应纳税额超过9000元至35000元(38500)	25%	1005
全月应纳税额超过35000元至55000元(58500)	30%	2755
全月应纳税额超过55000元至80000元(83500)	35%	5505
全月应纳税额超过80000元	45%	13505

# 出租车计费软件。

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。

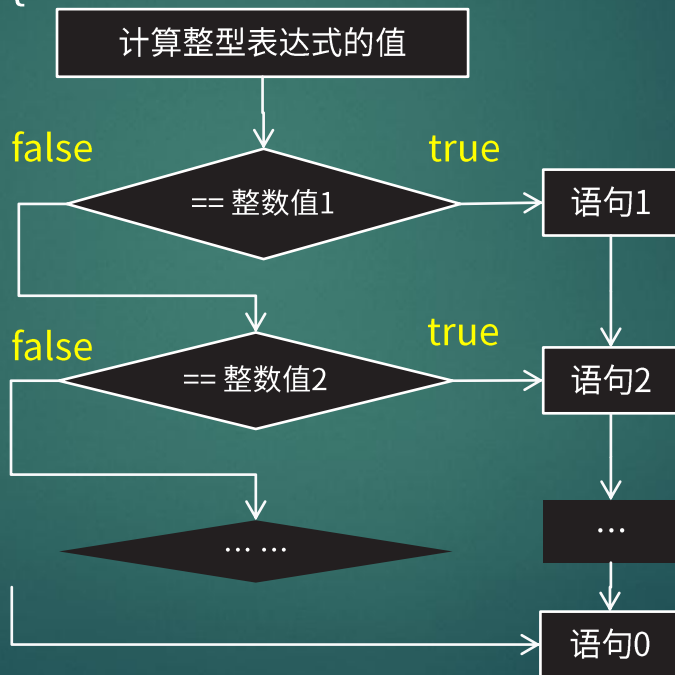
等候时间每2分半1元，不足部分不要钱。

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 =  $10 + (15-3) * 2 + (16-15) * 3 + 1 = 38$

# switch-case结构

```
switch (整型表达式) {  
  case 整数值1:  
    语句1  
  case 整数值2:  
    语句2  
    ... ..  
  default:  
    语句0  
}
```



# switch-case结构 (续)

```
switch (command) {  
  case 1:  
    System.out.println("显示全部记录  
");  
    break;  
  case 2:  
    System.out.println("查询登录记录  
");  
    break;  
  case 0:  
    System.out.println("欢迎使用");  
}
```

通常case1、case2、…、caseN对应完全不同的操作，可以和break语句配合使用，执行完相应语句后即退出switch块，不继续执行下面的语句。

# 命令解析器。

使用SWITCH-CASE结构实现输入1打印显示所有用户，输入2显示增加新用户，输入3显示修改用户信息，输入4显示删除用户，输入5显示退出。

- 1、运算符中的算数运算符、逻辑运算符、赋值运算符。
- 2、if代码(所有代码)。