

多路分支

switch-case

```
if ( type==1 )  
    System.out.println("你好");  
else if ( type==2 )  
    System.out.println("早");  
else if ( type==3 )  
    System.out.println("晚");  
else if ( type==4 )  
    System.out.println("再见");  
else  
    System.out.println("啊， 什么啊? ");
```

```
switch ( type ) {  
    case 1:  
        System.out.println("你好");  
        break;  
    case 2:  
        System.out.println("早上好");  
        break;  
    case 3:  
        System.out.println("晚上好");  
        break;  
    case 4:  
        System.out.println("再见");  
        break;  
    default:  
        System.out.println("啊， 什么啊? ");  
}
```

switch-case

```
switch ( 控制表达式 ) {  
case 常量:  
    语句  
    .....  
case 常量:  
    语句  
    .....  
default:  
    语句  
    .....  
}
```

- 控制表达式只能是整数型的结果
- 常量可以是常数，也可以是常数计算的表达式
- 根据表达式的结果，寻找匹配的case，并执行case后面的语句，一直到break为止
- 如果所有的case都不匹配，那么就执行default后面的语句；如果没有default，那么就什么都不做

```
switch ( type )
{
case 1:
case 2:
    System.out.println("你好");
    break;
case 3:
    System.out.println("晚上好");
case 4:
    System.out.println("再见");
    break;
default:
    System.out.println("啊，什么  
啊? ");
    break;
}
```

break

- switch语句可以看作是一种基于计算的跳转，计算控制表达式的值后，程序会跳转到相匹配的case（分支标号）处。分支标号只是说明switch内部位置的路标，在执行完分支中的最后一条语句后，如果后面没有break，就会顺序执行到下面的case里去，直到遇到一个break，或者switch结束为止。

try

$$\begin{aligned} f(x) = & -1; x < 0 \\ & 0; x = 0 \\ & 2x; x > 0 \end{aligned}$$

- 这个分段函数能否用switch-case?