

Uso de switch

Instrucción switch

➤ Evalúa una expresión cuyo resultado debe ser entero (int). Se ejecutarán diferentes bloques de sentencias en función de los posibles resultados

```
switch(expresion){  
    case valor1:  
        //sentencias  
        break;  
    case valor2:  
        //sentencias  
        break;  
    :  
    default:  
        //sentencias  
}
```

```
switch(a*2){  
    case 0:  
        System.out.println("nada");  
        break;  
    case 4:  
        System.out.println("cerca");  
        break;  
    case 8:  
        System.out.println("acierto");  
        break;  
    default:  
        System.out.println("error");  
}
```

Funcionamiento de switch

- Si el resultado de la expresión coincide con uno de los valores indicados en los case, ejecutará el bloque correspondiente de sentencias, sino, entrará en el bloque default (opcional).
- La instrucción break al final de cada bloque case es opcional. Si no se indica, el programa entrará en el siguiente bloque:

```
int a=10;  
switch(a){  
    case 10:  
        System.out.println("Es 10");  
    default:  
        System.out.println("Sin valor");
```

```
}
```

El programa muestra:
Es 10
Sin valor

Valores de los *case*

➤ Los valores de los *case* deben ser literales o constantes enteras *int*, o convertibles implícitamente en *int*:

```
int p=5;
```

```
final int k=30;
```

```
int n=3;
```

```
switch(p){
```

```
    case 10: //ok, es un literal int
```

```
    case k:  //ok, es una constante
```

```
    case n: //error de compilación, no es una constante
```

```
    case '@': //ok, char convertible implícitamente a int
```

```
}
```

Bloque default

➤ El bloque default es opcional y no tiene que aparecer necesariamente al final:

```
int p=5;  
switch(p){  
    case 10:  
        System.out.println("Es 10");  
    default:  
        System.out.println("Default");  
    case 2:  
        System.out.println("Es 2");  
}
```

El programa muestra:
Default
Es 2

switch con valores String

- Desde Java 7 es posible evaluar en un switch expresiones cuyo resultado sea una cadena de caracteres:

```
String data="prueba";  
final String s="hello";  
switch(data){  
    case "uno": //ok  
  
    case s: //ok, es una constante  
  
    case 10: //error de compilación  
}
```

- Si la expresión es evaluada como String, no se admiten valores enteros en los case y viceversa

Instrucción switch mejorada

- Permite incluir más de un valor en los case.
- Los bloques de instrucciones delimitadas por llaves
- Utilización de *yield* para devolución de resultado:

Si se usa en expresiones, debe devolver valor para **todos** los posibles resultados

```
double factura=30;
int envio=5;
int descuento=0;
descuento=switch(envio) {
    case 1,2,3->{
        int op=envio*2;
        yield op;
    }
    case 4,5->{
        int op=envio*2+1;
        yield op;
    }
    default->5;
};
System.out.println(factura*100/(100+descuento));
```