

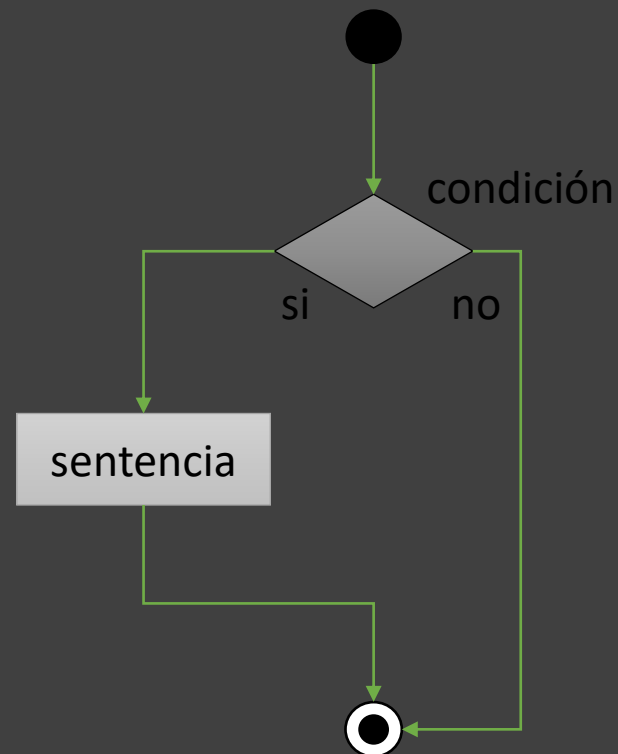
Flujo de control en Java



Andrés Guzmán F

Sentencia if/else

- Ejecuta un bloque de código si alguna condición es verdadera



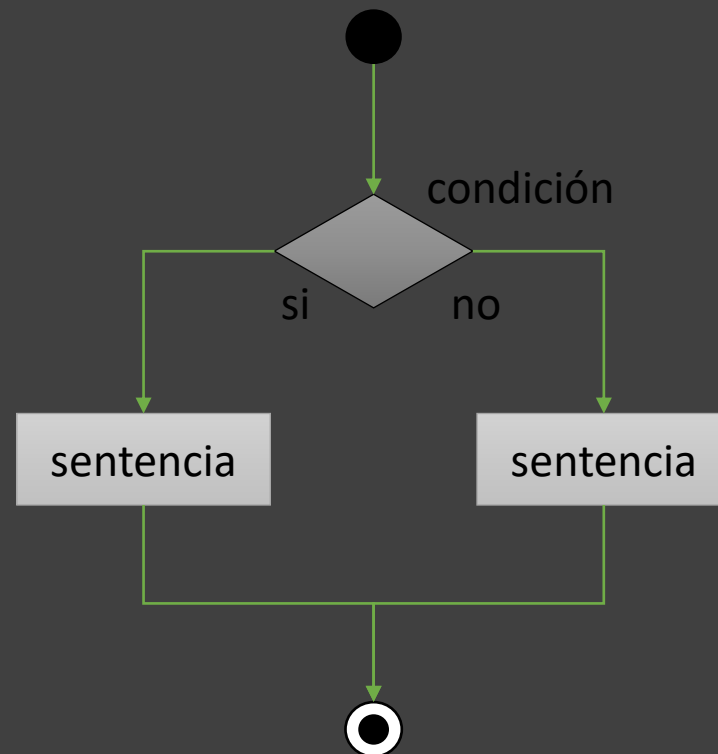
Sentencia if/else

- Ejecuta un bloque de código si alguna condición es verdadera

```
if(expresion) {  
    // ejecuta el bloque de código si cumple la condición  
}
```

Sentencia if/else

- Si la expresión es falsa podemos hacer uso de else



Sentencia if/else

- Si la expresión es falsa podemos hacer uso de else

```
if(expresion) {  
    // ejecuta el bloque de código si cumple la condición  
} else {  
    // sentencia a ejecutar si NO se cumple la condición  
}
```

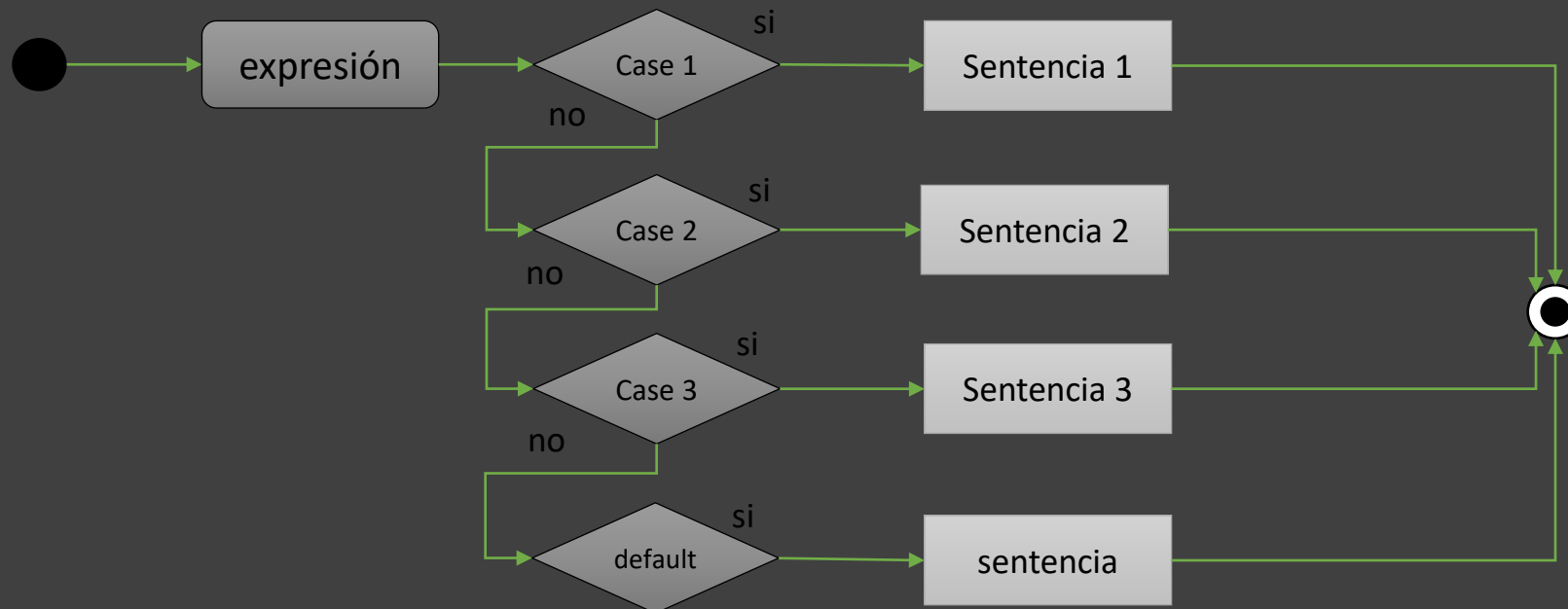
Sentencia if/else

- O bien, una serie de comparaciones else if y una sentencia else

```
if(expresion) {  
    // bloque a ejecutar si cumple la primera condición  
} else if(expresion) {  
    // sentencia a ejecutar si cumple segunda condición  
} else {  
    // sentencia a ejecutar si NO cumple ninguna condición  
}
```

Sentencia switch case

- La sentencia switch o interruptor se utiliza para realizar sentencias condicionalmente basadas en alguna expresión
- De acuerdo al valor de la expresión se ejecuta la sentencia case apropiada



Sentencia switch case

- La sentencia **break** hace que el control salga de la sentencia **switch** y continúe con la siguiente línea

```
switch(variable){  
  case valor1:  
    // ejecuta sentencia 1  
    break;  
  case valor2:  
    // ejecuta sentencia 3  
    break;  
  case valor3:  
    // ejecuta sentencia 3  
    break;  
  default:  
    // Si no existe coincidencia, se ejecuta la sentencia default  
}
```


Separadores

- Existen otros caracteres en Java con significado especial, son los separadores:

Separador	Descripción
()	Se utilizan para contener condiciones en sentencias de flujo de control. Contienen listas de parámetros, tanto en la definición de un método como en la llamada al mismo. También se utilizan para modificar la precedencia en una expresión. También para realizar conversiones de tipo
{ }	Se utilizan para definir bloques de código, definir ámbitos También para contener los valores iniciales de los arreglos.
[]	Se utiliza tanto para declarar arreglos o matrices como para referenciar valores dentro de los mismos.
;	Separa sentencias. También se utiliza para encadenar sentencias dentro de una estructura for.
,	Separa identificadores consecutivos en la declaración de variables y en las listas de parámetros de un método.
.	Separa un nombre de atributo o método de una variable de referencia o objeto. También separa nombre de paquete de los de un subpaquete o una clase.