

Uso de operadores y construcciones de decisión

fundacion@proydesa.org) has a
this Student Guide.

Importancia

- Cuando tiene que tomar una decisión para la que existan varios caminos diferentes, ¿cómo selecciona en última instancia un camino en lugar de los otros?
- Por ejemplo, ¿en qué cosas piensa cuando va a comprar un artículo?

(fundacion@proydesa.org) has a
to use this Student Guide.

Importancia

- Cuando tiene que tomar una decisión para la que existan varios caminos diferentes, ¿cómo selecciona en última instancia un camino en lugar de los otros?
- Por ejemplo, ¿en qué cosas piensa cuando va a comprar un artículo?

(fundacion@proydesa.org) has a
to use this Student Guide.

Uso de operadores relacionales y condicionales



adacion@proydesa.org) has a
this Student Guide.

Ejemplo de ascensor

```
public class Elevator {  
  
    public boolean doorOpen=false; // Doors are closed by default  
    public int currentFloor = 1; // All elevators start on first floor  
    public final int TOP_FLOOR = 10;  
    public final int MIN_FLOORS = 1;  
  
    public void openDoor() {  
        System.out.println("Opening door.");  
        doorOpen = true;  
        System.out.println("Door is open.");  
    }  
  
    public void closeDoor() {  
        System.out.println("Closing door.");  
        doorOpen = false;  
        System.out.println("Door is closed.");  
    }  
    ...  
}
```

Abrir la puerta.



Cerrar la puerta.



Archivo ElevatorTest.java

```
public class ElevatorTest {  
    public static void main(String args[]) {  
  
        Elevator myElevator = new Elevator();  
  
        myElevator.openDoor();  
        myElevator.closeDoor();  
        myElevator.goDown();  
        myElevator.goUp();  
        myElevator.goUp();  
        myElevator.goUp();  
        myElevator.openDoor();  
        myElevator.closeDoor();  
        myElevator.goDown();  
        myElevator.openDoor();  
        myElevator.goDown();  
        myElevator.openDoor();  
    }  
}
```



Operadores relacionales

Condición	Operador	Ejemplo
Es igual a	<code>==</code>	<code>int i=1;</code> <code>(i == 1)</code>
Es distinto de	<code>!=</code>	<code>int i=2;</code> <code>(i != 1)</code>
Es menor que	<code><</code>	<code>int i=0;</code> <code>(i < 1)</code>
Es menor o igual que	<code><=</code>	<code>int i=1;</code> <code>(i <= 1)</code>
Es mayor que	<code>></code>	<code>int i=2;</code> <code>(i > 1)</code>
Es mayor o igual que	<code>>=</code>	<code>int i=1;</code> <code>(i >= 1)</code>

Prueba de la igualdad entre cadenas

Ejemplo:

```
public class Employees {  
  
    public String name1 = "Fred Smith";  
    public String name2 = "Joseph Smith";  
  
    public void areNamesEqual() {  
  
        if (name1.equals(name2)) {  
            System.out.println("Same name.");  
        }  
        else {  
            System.out.println("Different name.");  
        }  
    }  
}
```



Operadores condicionales comunes

Operación	Operador	Ejemplo
Si una condición AND otra condición	&&	<pre>int i = 2; int j = 8; ((i < 1) && (j > 6))</pre>
Si una condición OR otra condición		<pre>int i = 2; int j = 8; ((i < 1) (j > 10))</pre>
NOT	!	<pre>int i = 2; (!(i < 3))</pre>

Operador condicional ternario

Operación	Operador	Ejemplo
Si someCondition es true, asigne el valor de value1 al resultado. En caso contrario, asigne el valor de value2 al resultado.	?:	<code>someCondition ? value1 : value2</code>

Creación de construcciones `if` e `if/else`

Una sentencia `if`, o una construcción `if`, ejecuta un bloque de código si una expresión es *true*.

fundacion@proydesa.org) has a
to use this Student Guide.

Construcción if

- Sintaxis:

```
if (boolean_expression) {  
    code_block;  
} // end of if construct  
// program continues here
```

- Ejemplo de posible salida:

```
Opening door.  
Door is open.  
Closing door.  
Door is closed.  
Going down one floor.  
Floor: 0 ← Se trata de un error en la lógica.  
Going up one floor.  
Floor: 1  
Going up one floor.  
Floor: 2  
...
```

Construcción if: Ejemplo

```
...  
public void goDown() {
```

```
    if (currentFloor == MIN_FLOORS) {  
        System.out.println("Cannot Go down");  
    }
```

El ascensor no puede bajar y se muestra un error.

```
    if (currentFloor > MIN_FLOORS) {  
        System.out.println("Going down one floor.");  
        currentFloor--;  
        System.out.println("Floor: " + currentFloor);  
    }
```

El ascensor puede bajar y se muestra la planta actual más la nueva planta.

```
}  
}
```

Construcción `if`: Salida

Ejemplo de posible salida:

```
Opening door.  
Door is open.  
Closing door.  
Door is closed.  
Cannot Go down ← La lógica de Elevator evita el problema.  
Going up one floor.  
Floor: 2  
Going up one floor.  
Floor: 3  
...
```

Sentencias if anidadas

```
...  
public void goDown() {  
  
    if (currentFloor == MIN_FLOORS) {  
        System.out.println("Cannot Go down");  
    }  
  
    if (currentFloor > MIN_FLOORS) {  
  
        if (!doorOpen) {  
            System.out.println("Going down one floor.");  
            currentFloor--;  
            System.out.println("Floor: " + currentFloor);  
        }  
    }  
}  
}
```

Sentencia
if anidada

Construcción if/else

Sintaxis:

```
if (boolean_expression) {  
    <code_block1>  
}  
// end of if construct  
  
else {  
    <code_block2>  
}  
// end of else construct  
  
// program continues here
```

Construcción if/else: Ejemplo

```
public void goUp() {  
    System.out.println("Going up one floor.");  
    currentFloor++;  
    System.out.println("Floor: " + currentFloor);  
}
```

```
public void goDown() {
```

```
    if (currentFloor == MIN_FLOORS) {  
        System.out.println("Cannot Go down");  
    }
```

```
    else {  
        System.out.println("Going down one floor.");  
        currentFloor--;  
        System.out.println("Floor: " + currentFloor);  
    }
```

```
}
```

```
}
```

Se ejecuta si la
expresión es true.

Se ejecuta si la
expresión es false.

Construcción if/else

Ejemplo de posible salida:

```
Opening door.  
Door is open.  
Closing door.  
Door is closed.  
Cannot Go down ← La lógica de Elevator evita el problema.  
Going up one floor.  
Floor: 2  
Going up one floor.  
Floor: 3  
...
```

Encadenamiento de construcciones if/else

Sintaxis:

```
if (boolean_expression) {  
    <code_block1>  
} // end of if construct  
  
else if (boolean_expression){  
    <code_block2>  
} // end of else if construct  
  
else {  
    <code_block3>  
}  
// program continues here
```

Encadenamiento de construcciones if/else

```
...  
public void calculateNumDays() {
```

```
    if (month == 1 || month == 3 || month == 5 || month == 7 ||  
        month == 8 || month == 10 || month == 12) {
```

```
        System.out.println("There are 31 days in that month.");
```

```
    }
```

```
    else if (month == 2) {
```

```
        System.out.println("There are 28 days in that month.");
```

```
    }
```

```
    else if (month == 4 || month == 6 || month == 9 || month == 11) {
```

```
        System.out.println("There are 30 days in that month.");
```

```
    }
```

```
    else {
```

```
        System.out.println("Invalid month.");
```

```
...
```

1

Se ejecuta cuando la
sentencia if es true.

2

Se ejecuta cuando la
primera sentencia if es
false y la sentencia
else es true.

3

Se ejecuta cuando la primera
sentencia if es false, la
primera sentencia else es
false y esta sentencia else
es true.

4

Se ejecuta cuando todas
las sentencias son false.

Uso de la construcción `switch`

Sintaxis:

```
switch (variable) {  
    case literal_value:  
        <code_block>  
        [break;]  
    case another_literal_value:  
        <code_block>  
        [break;]  
    [default:]  
        <code_block>  
}
```


Uso de la construcción switch: Ejemplo

```
public class SwitchDate {  
  
    public int month = 10;  
  
    public void calculateNumDays() {  
  
        switch(month) {  
        case 1:  
            case 3:  
            case 5:  
            case 7:  
            case 8:  
            case 10:  
            case 12:  
                System.out.println("There are 31 days in that month.");  
                break;  

```

...



Cuándo utilizar construcciones `switch`

- Pruebas de igualdad
- Pruebas en un *único* valor, como `customerStatus`
- Pruebas en el valor del tipo `int`, `short`, `byte`, `0 char` y `String`
- Pruebas en un valor corregido conocido en el momento de la compilación