

8. Sentencias condicionales (II) .

8.1 Operadores lógicos

Los operadores de comparación se pueden combinar con los operadores lógicos. Por ejemplo, si queremos saber si la variable *a* es mayor que *b* y además es menor que *c*, escribiríamos `if ((a > b) && (a < c))`. En la siguiente tabla se muestran los operadores lógicos de Java:

OPERADOR	NOMBRE	EJEMPLO	DEVUELVE VERDADERO CUANDO...
&&	y	<code>(7 > 2) && (2 < 4)</code>	las dos condiciones son verdaderas
	o	<code>(7 > 2) (2 < 4)</code>	al menos una de las condiciones es verdadera
!	no	<code>!(7 > 2)</code>	la condición es falsa

Vamos a ver cómo funcionan los operadores lógicos con un ejemplo. Mediante `if ((n < 1) || (n > 100))` se pueden detectar los números que no están en el rango de 1 a 100; literalmente sería “*si n es menor que 1 o n es mayor que 100*”.

```

import java.util.Scanner;

public class OperadoresLogicos1 {

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);
        String linea;
        int n;

        System.out.println("Adivina el número que estoy pensando.");
        System.out.print("Introduce un número entre el 1 y el 100: ");

        linea = s.nextLine();
        n = Integer.parseInt( linea );

        // Número fuera del rango permitido
        if ((n < 1) || (n > 100)) {
            System.out.println("El número introducido debe estar en el intervalo 1 - 100.");
            System.out.print("Tienes otra oportunidad, introduce un número: ");
            linea = s.nextLine();
            n = Integer.parseInt( linea );
        }

        // Se compara al número pensado, el 27
        if (n == 27) {
            System.out.println("¡Enhorabuena!, ¡has acertado!");
        } else {
            System.out.println("Lo siento, ese no es el número que estoy pensando.");
        }
    }
}

```

El comportamiento de los operadores lógicos se muestra en la siguiente tabla de verdad, donde “V” significa “verdadero” y “F” significa “falso”.

A	B	A && B	A B	!A	!B
V	V	V	V	F	F
V	F	F	V	F	V
F	V	F	V	V	F
F	F	F	F	V	V

En el siguiente programa puedes ver el uso de operadores lógicos combinado con operadores relacionales (operadores de comparación). Intenta adivinar cuál será el resultado mirando el código.

```
public static void main(String[] args) {  
  
    boolean a = true;  
    boolean b = false;  
  
    System.out.println("a = " + a);  
    System.out.println("b = " + b);  
    System.out.println("a && b = " + (a && b));  
    System.out.println("a || b = " + (a || b));  
    System.out.println("!a = " + !a);  
    System.out.println("a || (6 > 10) = " + (a || (6 > 10)));  
    System.out.println("((4 <= 4) || false) && (!a) = " + (((4 <= 4) || false) && (!a)));  
}
```

8.2 La sentencia switch (selección múltiple)

A veces es necesario comparar el valor de una variable con una serie de valores concretos. La selección múltiple es muy parecida (aunque no es exactamente igual) a una secuencia de varias sentencias if.

El formato de switch es el que se muestra a continuación. En lenguaje natural sería algo así como “Si variable vale valor1 entonces entra por case valor1; si variable vale valor2 entonces entra por case valor2;... si variable no vale ninguno de los valores que hay en los distintos case entonces entra por default:.

```
switch(variable) {  
    case valor1:  
        sentencias  
        break;  
    case valor2:  
        sentencias  
        break;  
    .  
    .  
    .  
    default:  
        sentencias  
}
```

A continuación tienes un ejemplo completo en Java. Se pide al usuario un número de mes y el programa da el nombre del mes que corresponde a ese número.

```
String nombreDelMes;
Scanner s;
int mes;

System.out.print("Por favor, introduzca un numero de mes: ");
s = new Scanner(System.in);
mes = s.nextInt();
s.close();

switch (mes) {
case 1:
    nombreDelMes = "enero";
    break;
case 2:
    nombreDelMes = "febrero";
    break;
case 3:
    nombreDelMes = "marzo";
    break;
case 4:
    nombreDelMes = "abril";
    break;
case 5:
    nombreDelMes = "mayo";
    break;
case 6:
    nombreDelMes = "junio";
    break;
case 7:
    nombreDelMes = "julio";
    break;
case 8:
    nombreDelMes = "agosto";
    break;
case 9:
    nombreDelMes = "septiembre";
    break;
case 10:
    nombreDelMes = "octubre";
    break;
case 11:
    nombreDelMes = "noviembre";
}
```

```

        break;
    case 12:
        nombreDelMes = "diciembre";
        break;
    default:
        nombreDelMes = "no existe ese mes";
    }
    System.out.println("Mes " + mes + ": " + nombreDelMes);
}
}

```

Observa que es necesario introducir un *break* después de la asignación de la variable *nombreDelMes*. En caso de no encontrarse el *break*, el programa continúa la ejecución en la línea siguiente.

El bloque que corresponde al *default* se ejecuta cuando la variable no coincide con ninguno de los valores de los *case*. Escribiremos siempre el *default* al final de la sentencia *switch* aunque no sea necesario.

La sentencia *switch* se utiliza con frecuencia para crear menús.

```

/*
 *
 *  Calcula el área de diferentes figuras
 *
 */

double lado;
double base;
double altura;
Scanner s = new Scanner(System.in);
int opcion;

System.out.println(" CÁLCULO DE ÁREAS");
System.out.println(" -----");
System.out.println(" 1. Cuadrado");
System.out.println(" 2. Rectángulo");
System.out.println(" 3. Triángulo");
System.out.print("\n Elija una opción (1-3): ");

opcion = Integer.parseInt(s.nextLine());

switch (opcion) {
case 1:
    System.out.print("\nIntroduzca el lado del cuadrado en cm: ");
    lado = Double.parseDouble(s.nextLine());
    System.out.println("\nEl área del cuadrado es " + (lado * lado) + " cm2");
    break;
case 2:
    System.out.print("\nIntroduzca la base del rectángulo en cm: ");
    base = Double.parseDouble(s.nextLine());
    System.out.print("Introduzca la altura del rectángulo en cm: ");
    altura = Double.parseDouble(s.nextLine());
    System.out.println("El área del rectángulo es " + (base * altura) + " cm2");
    break;
case 3:
    System.out.print("\nIntroduzca la base del triángulo en cm: ");
    base = Double.parseDouble(s.nextLine());
    System.out.print("Introduzca la altura del triángulo en cm: ");
    altura = Double.parseDouble(s.nextLine());

    System.out.println("El área del triángulo es " + ((base * altura) / 2) + " cm2");
    break;
default:
    System.out.print("\nLo siento, la opción elegida no es correcta.");
}

s.close();

```

Ejercicio 8.1

Escribe un programa que pida por teclado un día de la semana y que diga qué asignatura toca a primera hora ese día.

Ejercicio 8.2

Realiza un programa que pida una hora por teclado y que muestre luego buenos días, buenas tardes o buenas noches según la hora. Se utilizarán los tramos de 6 a 12, de 13 a 20 y de 21 a 5. respectivamente. Sólo se tienen en cuenta las horas, los minutos no se deben introducir por teclado.

Ejercicio 8.3

Vamos a ampliar uno de los ejercicios de la relación anterior para considerar las horas extras. Escribe un programa que calcule el salario semanal de un trabajador teniendo en cuenta que las horas ordinarias (40 primeras horas de trabajo) se pagan a 12 euros la hora. A partir de la hora 41, se pagan a 16 euros la hora.

Ejercicio 8.4

Realiza un programa que resuelva una ecuación de segundo grado (del tipo $ax^2 + bx + c = 0$).

Ejercicio 8.5

Escribe un programa que nos diga el horóscopo a partir del día y el mes de nacimiento.

Ejercicio 8.6

Escribe un programa que nos diga el horóscopo a partir del día y el mes de nacimiento.

Ejercicio 8.7

Escribe un programa que diga cuál es la última cifra de un número entero introducido por teclado.

Ejercicio 8.8

Realiza un programa que, dado un día de la semana (de lunes a viernes) y una hora (horas y minutos), calcule cuántos minutos faltan para el fin de semana. Se considerará que el fin de semana comienza el viernes a las 14:15h. Se da por hecho que el usuario introducirá un día y hora correctos, anterior al viernes a las 14:15h.

Ejercicio 8.9

Una pastelería nos ha pedido realizar un programa que haga presupuestos de tartas. El programa preguntará primero de qué sabor quiere el usuario la tarta: manzana, fresa o chocolate. La tarta de manzana vale 18 euros y la de fresa 16. En caso de seleccionar la tarta de chocolate, el programa debe preguntar además si el chocolate es negro o blanco; la primera opción vale 14 euros y la segunda 15. Por último se pregunta si se añade nata y si se personaliza con un nombre; la nata suma 2.50 euros y la escritura del nombre 2.75 euros.

Ejemplo:

```
Elija un sabor (manzana, fresa o chocolate): fresa
```

```
¿Quiere nata? (si o no): si
```

```
¿Quiere ponerle un nombre? (si o no): si
```

```
Tarta de fresa: 16,00 €
```

```
Con nata: 2,50 €
```

```
Con nombre: 2,75 €
```

```
Total: 21,25
```

Implementa el juego piedra, papel y tijera. Primero, el usuario 1 introduce su jugada y luego el usuario 2. Si alguno de los usuarios introduce una opción incorrecta, el programa deberá mostrar un mensaje de error.

```
Turno del jugador 1 (introduzca piedra, papel o tijera): papel
```

```
Turno del jugador 2 (introduzca piedra, papel o tijera): tijera
```

```
Gana el jugador 2
```