

## Unidad 02. Estructuras básicas de control

### 1. Estructura condicional (if)

Ejemplos:

Ejemplo 1:

```
if (true) {  
    System.out.println("Esto siempre se imprimirá");  
}
```

Ejemplo 2:

```
int sueldo=2500, irpf=10;  
if (sueldo>2250) {  
    irpf=15;  
}
```

Ejemplo 3:

```
int nota=6;  
if (nota>=5) {  
    System.out.println("Has aprobado");  
} else {  
    System.out.println("Has suspendido");  
}
```

Ejemplo 4:

```
nota=7;  
if (nota==10) {  
    System.out.println("Matrícula de Honor");  
} else if (nota>=9) {  
    System.out.println("Sobresaliente");  
} else if (nota>=7) {  
    System.out.println("Notable");  
} else if (nota>=6) {  
    System.out.println("Bien");  
} else if (nota>=5) {  
    System.out.println("Suficiente");  
} else {  
    System.out.println("Suspenso");  
}
```

### \* Ejercicios

- 1.1.- Programa que lea un número entero pasado desde línea de comandos y diga si es par o impar
- 1.2.- Programa que lea un número entero pasado desde línea de comandos y diga la nota simbólica (5 -> Suficiente, 6 -> Bien, 7 u 8 -> Notable, 9 -> Sobresaliente, 10 -> Matrícula, Menos de 5 -> Suspenso)
- 1.3.- Programa que lea un sueldo pasado desde línea de comandos y muestre el IRPF que le correspondería pagar en función de la siguiente tabla:

Sueldo entre	IRPF correspondiente
[15000, +Infinito)	25
[10000, 15000)	20
[7500,10000)	15
[5000,7500)	12
[2500,5000)	10
[0,2500)	0

## 2. Switch

```
switch(nota) {  
    case 10: System.out.println("Matrícula");break;  
    case 9: System.out.println("Sobresaliente"); break;  
    case 8:  
    case 7: System.out.println("Notable"); break;  
    case 6: System.out.println("Bien"); break;  
    case 5: System.out.println("Suficiente"); break;  
    default: System.out.println("Suspendido");  
}
```

### \* Ejercicios

2.1.- Programa que reciba un número entre [1,7] pasado desde línea de comandos y diga el día de la semana (p.ej. 1 -> Lunes)

2.2.- Programa que reciba un número entre [1,12] pasado desde línea de comandos y diga el mes correspondiente (p.ej. 6 -> Junio)

## 3. While

```
int i=1;  
int n=7;  
System.out.println("Tabla de Multiplicar del 7");  
while (i<=10) {  
    System.out.println(n+"*"+i+"="+n*i);  
    i++;  
}
```

### \* Ejercicios

3.1.- Programa que reciba un número pasado desde línea de comandos e imprima su tabla de multiplicar

3.2.- Programa que reciba un número pasado desde línea de comandos e imprima todos los números pares comprendidos entre 0 y ese número

3.3.- Programa que reciba un número pasado desde línea de comandos e imprima la suma de todos los números comprendidos entre 1 y ese número

3.4.- Programa que reciba un número pasado desde línea de comandos y muestre todos los números múltiplos de 5 comprendidos entre 0 y ese número

## 4. Do while

```
java.util.Scanner s=new java.util.Scanner(System.in);  
int opt, n1, n2;  
do {  
    System.out.println("Elige una opción de entre las siguientes");  
    System.out.println("1. Suma");  
    System.out.println("2. Resta");  
    System.out.println("3. Salir");  
    opt=s.nextInt();  
    System.out.println("Introduce n1: ");  
    n1=s.nextInt();  
    System.out.println("Introduce n2: ");  
    n2=s.nextInt();  
    switch(opt) {  
        case 1: System.out.println("La suma es: "+(n1+n2));  
        case 2: System.out.println("La resta es: "+(n1-n2));  
        default: System.out.println("Error opción incorrecta: "+opt);  
    }  
} while(opt!=3);
```

### \* Ejercicios

4.1.- Añadir al ejemplo anterior las opciones de multiplicar, dividir y calcular el resto

4.2.- ¿Qué diferencia hay entre un bucle **while** y un bucle **do-while**?

4.3.- Generar 2 números entre [1,49] y pedirle al usuario 2 números en el mismo intervalo y comprobar si el usuario ha acertado los 2 números

## 5. For

```
n=8;
System.out.println("Tabla de Multiplicar del "+n);
for(i=1;i<=10;i++) {
    System.out.println(n+"*"+i+"="+n*i);
}
```

### \* Ejercicios

- 5.1.- Realizar todos los ejercicios del apartado 3. **while** utilizando en su lugar la estructura for
- 5.2.- Programa que genere 6 números aleatorios entre [1,49]
- 5.3.- Programa que simule el lanzamiento de un dado 5 veces consecutivas

## 6. Break y continue

```
//imprime los números impares
int i=0;
while (true) {
    i++;
    if (i>10) { break; }
    if (i%2==0) { continue; }
    System.out.println(i);
}
```

## 7. Introducción a los métodos (funciones)

Siempre que estemos copiando y pegando un trozo de código, deberíamos plantearnos en abstraer ese trozo de código y convertirlo en una función.

Ejemplo: Ejercicio 6 (número primo)

## 8. Try-catch-finally (Excepciones)

```
jshell> int a=5, b=0;
```

```
a ==> 5
```

```
b ==> 0
```

```
jshell> a/b;
```

```
| Exception java.lang.ArithmeticException: / by zero
|   at (#9:1)
```

```
int a=6, b=0;
try {
    System.out.println("La división de a/b es: "+(a/b));
} catch (ArithmeticException e) {
    System.out.println("Se ha Producido la siguiente excepción: ");
    System.out.println(e.getMessage());
} finally {
    //Liberar recursos
}
System.out.println("Pero la ejecución del programa no se corta");
```

## **Boletín Ejercicios del Tema 2**

NOTA. En los ejercicios a continuación cuando se diga "**reciba**" quiere decirse que se pasará como parámetro en línea de comandos, de lo contrario se hará referencia a que el dato se lee del teclado

- 1.- Programa que **reciba** tres números y diga cuál es el menor y el mayor
- 2.- Programa que genere un número aleatorio entre [100,200] e imprima el número y diga si es par o impar
- 3.- Programa que **reciba** un número, n, y nos dé la suma de los números comprendidos entre [1,n]
- 4.- Ídem con el producto
- 5.- Programa que **reciba** un número, n, y muestre las tablas de multiplicar del 2 hasta n
- \*6.- Programa que **reciba** un número e imprima todos los números primos comprendidos entre 1 y ese número (un número es primo si y sólo si es divisible por 1 y él mismo) (repetido en 11)
- 7.- Programa que **reciba** dos números y calcule la potencia del primero elevado al segundo
- \* 9.- Programa que nos diga cuántos dígitos tiene un número introducido por teclado
- 10.- Programa que **reciba** tres números a, b y c, coeficientes de una ecuación de segundo grado y dé sus soluciones (usar try catch)
- \* 11.- Programa que **reciba** un número comprendido entre [1,3999] y lo muestre en números **Romanos** ([https://es.wikipedia.org/wiki/Numeración\\_romana](https://es.wikipedia.org/wiki/Numeración_romana))
- \* 12.- Programa que **reciba** un número y diga si es primo
- \* 13.- Programa que **reciba** un número de 3 cifras y diga si es un número de **Armstrong**
- \* 14.- Programa que **reciba** un número, n, e imprima los n primeros términos de la sucesión de *Fibonacci*
- \* 15.- Programa que genere un número aleatorio entre [1,100] y le dé al usuario la oportunidad de adivinarlo en 7 intentos como máximo...
- \*\* 16.- Programa que imprima el "**lucky number**" de una persona. Éste se consigue reduciendo la fecha de nacimiento a un sólo dígito. P.Ej. si la fecha de nacimiento es 11-02-1973, entonces el sería:  
 $11+2+1973=1986 \rightarrow 1+9+8+6=24 \rightarrow 2+4=6$ . Entonces **6** sería el resultado.
- \*\* 17.- Programa que **reciba** un número impar, n, e imprima un **rombo** de caracteres '\*' de diagonal n
- \*\* 18.- Programa que **reciba** un valor, n, en minutos, diga la hora actual y a partir de ese momento cuando la hora termine en n o múltiplo de n vuelva a decir la hora, y así hasta el infinito
- \*\* 19.- Programa que **reciba** la Entidad, Sucursal y Cuenta Bancaria y calcule los dígitos de control
- \*\* 20.- Programa que **reciba** un número romano y dé su valor en decimal
- \*\* 21.- Programa que **reciba** un día, mes y año y diga si la fecha es correcta (hay que tener en cuenta si el año es bisiesto)
- \*\* 22.- Programa que **reciba** un número entero y lo imprima al revés. P.ej.: si recibe el número 1759 debería imprimir 9571.