

Ejercicios prácticos con estructuras de iteración.

Ejercicio 1.

Escribe un programa que imprima números del 1 al 10 usando un bucle for.

Ejercicio 2.

Usa un bucle while para hacer una cuenta regresiva desde 10 hasta 1.

Ejercicio 3.

Utiliza un bucle for para imprimir la tabla de multiplicar de un número dado por el usuario.

Ejercicio 4.

Escribe un programa que sume números desde 1 hasta un número N proporcionado por el usuario, usando do-while.

Ejercicio 5.

Dada una serie de números ingresados por el usuario, encontrar el número más pequeño. El programa no tendrá en consideración si el usuario introduce un 0 para encontrar el número más pequeño, y se utilizará el valor 0 como condición de terminación.

Ejercicio 6.

Calcula el promedio de una serie de números ingresados por el usuario. El programa termina cuando se introduce un 0.

Ejercicio 7.

Imprime todos los números pares entre 1 y 100.

Ejercicio 8.

Usa un bucle while para imprimir números impares entre 1 y 100, saltándose los pares con continue.

Ejercicio 9.

Crea un programa que solicite un número entero mayor de 10 y sume el primer dígito y el último. El programa no se detendrá hasta que el usuario haya introducido un número mayor igual a 10. Por ejemplo, si se introduce el número 1754, el resultado debe ser 5.

Ejercicio 10.

Utiliza un bucle while para contar cuántas veces aparece un carácter específico en un string. El programa debe solicitar al usuario el texto y el carácter a buscar.

Ejercicio 11.

Calcula el factorial de un número N usando un bucle for. El factorial representado por $n!$, es el producto de todos los números enteros positivos desde 1 hasta n. Por ejemplo, el factorial de 5! es $5 * 4 * 3 * 2 * 1 = 120$

Ejercicio 12.

Implementa la secuencia de Fibonacci hasta un número N usando un bucle while. La secuencia de Fibonacci es una serie de números en la cual cada número es la suma de los dos números anteriores, su fórmula es: $F(n) = F(n-1) + F(n-2)$. Por ejemplo, la secuencia de Fibonacci del número 5 es: 0, 1, 1, 2, 3, 5

Ejercicio 13.

Escribe un programa que simule lanzar una moneda al aire varias veces hasta que salga cara (o se alcance un número máximo de intentos). El número de intentos se solicitará al usuario, si una vez alcanzado el número de intentos y no ha salido cara se debe mostrar un mensaje, indicando que no se ha conseguido. Nota. Para generar aleatoriamente un booleano, se puede utilizar `new Random().nextBoolean()`

Ejercicio 14.

Crea un pequeño juego para adivinar un número secreto entre 1 y 100. El programa generará un número al azar y pedirá al usuario que intente adivinarlo. Después de cada intento, el programa dirá si el número secreto es mayor o menor del elegido por el usuario. El juego continúa hasta que el usuario adivine el número correcto, y el programa indicará cuántos intentos se necesitaron para acertar. Nota. Para generar un número aleatorio se puede utilizar la función `.nextInt(numero)` del objeto `Random`

Ejercicio 15.

Crea un programa que defina un tipo enumerado llamado `DiasSemana`, el cual debe contener los siete días de la semana. Luego, escribe un código que utilice un bucle for y un foreach para mostrar todos los valores del enumerando

Ejercicio 16.

Realiza un programa para solicitar la altura de un triángulo de 90° luego imprimir un patrón de asteriscos que dibuje el triángulo. Por ejemplo, si se indica una altura de 3 se dibujará:

```
*  
**  
***
```

Ejercicio 17.

Realiza un programa que simule un cajero automático de monedas. A partir, de un importe debe indicar el número mínimo de monedas necesarias para dar el cambio específico. Se solicitará al usuario el importe total a retirar. Por ejemplo, 4.35, devolverá 4 euros y 35 céntimos.

Ejercicio 18.

Hacer un programa que muestre un cronometro, indicando las horas, minutos y segundos. No hace falta que pare el cronometro en ningún momento. Nota. Para pausar un tiempo la ejecución de un programa se puede utilizar el siguiente código:

```
try {  
    Thread.sleep(1000);  
} catch (InterruptedException e) {  
    e.printStackTrace();  
}
```