

Programación. Ejercicios Tema 3

- 1) ¿Es apropiada la introducción interactiva en una aplicación que debe leer 1000 números enteros?
- 2) ¿Qué tipo de expresión permite transformar un tipo de datos en otro tipo?
- 3) ¿Qué sentencia utilizarás para convertir una cadena en un entero?
- 4) Escribe la primera instrucción de una sentencia **while** para construir un bucle que itere hasta el valor de una variable de tipo **boolean** sea **true**.
- 5) Cuáles son las cuatro partes de un bucle controlado por contador.
- 6) ¿Cómo se utiliza una variable indicadora para controlar un bucle?
- 7) ¿Cuál es la diferencia entre una operación de incremento y una de suma en el interior de un bucle?
- 8) ¿Cuál es la diferencia entre un bucle controlado por una variable y uno controlado por contador?
- 9) ¿Qué tipo de bucle utilizarías para leer el precio de cierre en Bolsa al final del día, haciéndolo durante 11 semanas?
- 10) Explica la diferencia entre iteración y selección.
- 11) Cuál es la salida del siguiente bucle.

```
numero = 1;
while ( numero < 11 ) {
    numero++;
    System.out.println(numero);
}
```

- 12) Reescribe el bucle anterior para que imprima los números del 1 al 10 pero –sólo- reordenando las sentencias, sin modificar cómo están escritas.
- 13) ¿Cuántas iteraciones se realizan cuando se ejecuta el siguiente bucle?

```
numero = 2;
hecho = false;
while ( ! hecho ) {
    numero = numero * 2;
    if ( numero > 64 )
        hecho = true;
}
```

- 14) Cuál es la salida de estos bucles anidados.

```
i = 4;
while ( i >= 1 ) {
    j = 2;
    while ( j >= 1 ) {
        System.out.println(j + " ");
        j--;
    }
    System.out.println(i);
    i--;
}
```

- 15) El siguiente fragmento de código debería imprimir, supuestamente, los números pares entre 1 y 15 pero no lo hace. ¿Por qué? ¿Qué es lo que se obtiene? ¿Puedes corregirlo?

```
n = 2;
while ( n <= 15 ) {
    n = n + 2;
    System.out.println(n + " ");
}
```

- 16) Qué tipo de valor centinela elegirías para una aplicación que lee números de teléfono como enteros.

- 17) Considera el siguiente fragmento de código y los valores 5, 6, -3, 7, -4, 0, 5, 8 y 9

```
suma = 0;
i = 1;
limite = 8
finalizado = false;
while ( i <= limite && ! finalizado ) {
    numero = teclado.nextInt();
    if ( numero > 0 )
        suma = suma + numero;
    else if ( numero == 0 )
        finalizado = true;
    i++;
}
System.out.println("Fin de la prueba. " + suma + " " + numero);
```

¿Cuáles son los contenidos de **suma** y **numero** tras salir del bucle? ¿Se utilizan todos los datos para comprobar el código? Explica tus respuestas.

- 18)Cuál es la salida del siguiente fragmento de código.

```
i = 1;
while ( i <= 5 ) {
    suma = 0;
    j = 1;
    while ( j <= i ) {
        suma = suma + j;
        j++;
    }
    System.out.println(suma + " ");
    i++;
}
```

- 19) ¿Cuáles son los principales tipos de bucles?

- 20) Qué sucedería si olvidaras incrementar la variable de control de un bucle controlado por contador.

- 21) Distingue entre un contador de bucle y un contador de suceso.

- 22) Cómo es una expresión de asignación.

- 23) Escribe un algoritmo (no código Java) que, mediante bucles anidados, genere la siguiente salida:

```
1
1 2
1 2 3
1 2 3 4
```

- 24) Escribe un algoritmo (no código Java) que lea enteros, cuente cuántos números positivos han sido leídos y cuántos negativos. El proceso terminará cuando sea lea el número cero.
- 25) Escribe un algoritmo (no código Java) que sume los números pares entre 16 y 26, incluidos.
- 26) Escribe una instrucción que incremente en uno un contador y lo ponga de nuevo a cero cada vez que alcance el valor 13.
- 27)Cuál es la principal diferencia entre un bucle **while** y un bucle **do...while**.
- 28) Un problema requiere de un bucle controlado por contador que comienza en 10 y cuenta hasta 1. Escribe la primera línea de una instrucción **for** para controlar dicho bucle.
- 29) Qué tipo de expresión de control Java elegirías para implementar un bucle que es a la vez controlado por contador y por suceso, cuyo cuerpo podría no ejecutarse ni siquiera una vez.
- 30) Convierte un bucle **while** cuya condición de bucle es **delta <= alfa** en un bucle **do...while**. ¿Utilizan la misma condición?
- 31) Es verdadero o falso que una sentencia **do...while** termina siempre con un punto y coma.
- 32) Qué imprime el siguiente fragmento de código.

```
n = 0;
i = 1;
do {
    System.out.print(i);
    i++;
} while ( i <= n );
```

- 33) Qué imprime el siguiente fragmento de código.

```
n = 0;
for ( i = 1; i <= n; i++ )
    System.out.print(i);
```

- 34) Qué imprime el siguiente fragmento de código.

```
for ( i = 4; i >= 1; i-- ) {
    for ( j = i; j >= 1; j-- ) {
        System.out.print(j + " ");
    }
    System.out.println(i);
}
```

- 35) Qué imprime el siguiente fragmento de código.

```
for ( fila = 1; fila <= 10; fila++ ) {
    for ( columna = 1; columna <= 10 - fila; columna++ )
        System.out.print("**");
    for ( columna = 1; columna <= 2 * fila - 1; columna++ )
        System.out.print("**");
    for ( columna = 1; columna <= 10 - fila; columna++ )
        System.out.print("**");
    System.out.println();
}
```

- 36) Escribe un fragmento de código que lea y sume valores hasta que haya sumado diez valores diferentes o lea un número negativo, lo primero que suceda. Resuélvelo tanto con un bucle **while** como con un **do..while**.
- 37) Reescribe el siguiente fragmento de código utilizando un bucle **do...while** en lugar de uno **while**.

```
respuesta = teclado.nextInt();
```

```
while ( respuesta >= 0 && respuesta <= 127 ) {  
    respuesta = teclado.nextInt();  
}
```

38) Reescribe el siguiente fragmento de código utilizando un bucle **while**.

```
if ( unEntero >= 0 )  
    do {  
        System.out.println("" + unEntero);  
        unEntero = teclado.nextInt();  
    } while ( unEntero >= 0 );
```

39) Reescribe el siguiente fragmento de código utilizando un bucle **for**.

```
suma = 0;  
contador = 1;  
while ( contador <= 1000 )  
    suma += contador;  
    contador++;  
}
```

40) Reescribe el siguiente fragmento de código utilizando un bucle **while**.

```
for ( m = 93; m >= 5; m-- )  
    System.out.println(m + " " + m * m);
```

41) Reescribe el siguiente fragmento de código utilizando un bucle **do...while**.

```
for ( k = 9; k <= 21; k++ )  
    System.out.println(k + " " + 3 * k);
```