

Programación

EXAMEN TEMA 2 – ESTRUCTURAS DE CONTROL

(29/10/2024)

1. (0,5p) ¿Qué diferencia hay entre bucle *while* y un bucle *for*?
 - a) El bucle *for* puede no llegar a ejecutarse nunca, pero el *while* siempre se ejecuta al menos una vez.
 - b) El bucle *for* se ejecuta un número determinado de veces, y el *while* un número indeterminado de veces.
 - c) El bucle *for* no puede convertirse en un bucle *while*, pero sí al contrario.
 - d) El bucle *while* permite su inicialización, pero el bucle *for* no.

2. (0,5p) De acuerdo a la sintaxis del bucle *while*:

```
while (condición) {  
    sentencias  
}
```

¿qué es falso en relación a la *condición*?

- a) La condición es una variable booleana.
 - b) La condición sólo se evalúa al principio de la ejecución del bucle.
 - c) Si la condición es verdadera, se ejecuta el bloque de sentencias, y se vuelve al principio del bucle.
 - d) Si la condición es falsa, no se ejecuta el bloque de sentencias.
3. (0,5p) De acuerdo a la sintaxis del bucle *do-while*:

```
do {  
    sentencias  
} while (condición)
```

Señalar cuál de las siguientes afirmaciones es falsa:

- a) Si condición == true, entonces el bucle se sigue ejecutando.
- b) Aunque se cumpla condición == false, el bucle se llega a ejecutar alguna vez.
- c) Si condición == false, el bucle no se llega a ejecutar nunca.
- d) Ninguna de las anteriores es falsa.

4. (0,5p) De acuerdo a la sintaxis del bucle *for*:

```
for (inicialización ; condición ; actualización) {  
    sentencias  
}
```

Señalar cuál de las siguientes afirmaciones es falsa:

- a) La inicialización se realiza con cada iteración antes de entrar al bucle.
- b) La condición se comprueba cada vez antes de entrar al bucle.
- c) La actualización se realiza siempre al terminar de ejecutar la iteración.
- d) La inicialización, condición y actualización son elementos "opcionales".

5. (0,5p) Transforma este bucle *for*:

```
for (i=0; i <= 100; i+=2){  
    print(i + " ");  
}
```

en su equivalente bucle *while*. ¿Qué imprime por pantalla?

6. (1,5p) ¿Cuántas veces se ejecuta el cuerpo de los siguientes bucles?

a)

```
int i = 0;  
while (true) {  
    i++;  
    if (i < 10)  
        continue;  
    i++;  
    if (i==10)  
        break;  
}
```

b)

```
int i = 1;  
int n = 6;  
  
do {  
    if (i % 2 == 0) {  
        i++;  
        continue;  
    }  
  
    if (i > 4) {  
        break;  
    }  
  
    i++;  
} while (i <= n);
```

7. (0,75p) ¿Cuándo es más adecuado utilizar *switch-case* en lugar de *if-else if*? Pon un ejemplo.

8. (0,75p) Explica con tus palabras en qué casos se lanzan los 3 tipos de excepciones vistos hasta ahora (*InputMismatchException*, *ArithmeticException* y *NumberFormatException*).
9. (1p) Obtén el valor de las variables **resultado** y **a** después de ejecutarse las siguientes instrucciones con el operador ternario:

```
int a = 1, b = 2, c = 3;
int resultado = a++ == b && b == c - a ? c -= a :
                a == b && a++ == c ? a :
                c == a ? a-- : b;
System.out.println("resultado = " + resultado);
System.out.println("a = " + a);
```

10. (1p) Traduce la estructura del ejercicio anterior a instrucciones *if-then-else if* múltiples.
11. (1p) Desglosa en una tabla el orden de ejecución del siguiente bucle *for*:

```
for (int j = 6 ; j >= 0 ; j-=2){
    System.out.println(j);
}
```

Orden	Instrucción	j
1		
2		

12. (1,5p) Realiza la traza de los siguientes códigos:

a)

```
int n = 4, s = 0, k;
k = 2 * n - 1;
while(k >= 1){
    s += k;
    System.out.println(k);
    k = k - 2;
}
System.out.println(s);
```

b)

```
int n = 10;
int a = 0, b = 1, c;
System.out.println(a + " " + b + " ");
for (int i = 2; i < n; i++) {
    c = a + b;
    System.out.println(c + " ");
    a = b;
    b = c;
}
```

BONUS (+0,5p). Dado el siguiente bucle: **for (; ;)** explica lo que hace.