

Unidad 3: Bucles - 21/10/25

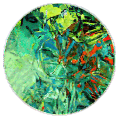
(2,5 puntos) Ejercicio 1: Realiza un programa que pida **10 números enteros** al usuario y muestre al final cuántos son **positivos, negativos y ceros**.

```
package examen;
import java.util.Scanner;
public class Ejercicio1 {
    public static void main(String[] args) {
        // Entrada de datos
        Scanner sc = new Scanner(System.in);
        // Acumuladores
        int positivos = 0, negativos = 0, ceros = 0;
        int numero;
        for (int i = 1; i <= 10; i++) { // Repite 10 veces
            System.out.print("Introduce el número " + i + ": ");
            numero = sc.nextInt();
            if (numero > 0) positivos++;
            else if (numero < 0) negativos++;
            else ceros++;
        }
        System.out.println("Positivos: " + positivos);
        System.out.println("Negativos: " + negativos);
        System.out.println("Ceros: " + ceros);
        sc.close(); // Cierra el scanner
    }
}
```

(2,5 puntos) Diseña un programa que pida números enteros al usuario y los vaya sumando hasta que introduzca el valor **0**, momento en el cual mostrará la suma total.

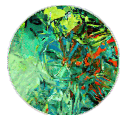
```
package examen;
import java.util.Scanner;
public class Ejercicio2 {
    public static void main(String[] args) {
        // Lectura de datos por teclado
        Scanner sc = new Scanner(System.in);
        int numero;
        // Acumulador de la suma
        int suma = 0;
        do {
            System.out.print("Introduce un número (0 para terminar): ");
            numero = sc.nextInt(); // Leer número
            suma += numero;        // Sumar al total
        } while (numero != 0);     // Repetir mientras no sea 0

        // Mostrar resultado
        System.out.println("La suma total es: " + suma);
        sc.close(); // Cerrar el scanner
    }
}
```



(5 puntos) Ejercicio 3: Selecciona la respuesta correcta

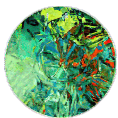
<p>3.1. Un bucle <i>do-while</i> se ejecutará, como mínimo:</p> <p>a) Cero veces.</p> <p>b) Una vez.</p> <p>c) Infinitas veces.</p> <p>d) Ninguna de las opciones anteriores es correcta.</p>	<p>3.2. El uso de llaves para encerrar el bloque de instrucciones de un bucle:</p> <p>a) Es siempre opcional.</p> <p>b) Es opcional si el bloque está formado por una única instrucción.</p> <p>c) En cualquier caso, su uso es obligatorio.</p> <p>d) El programador decide si las utiliza o no.</p>
<p>3.3. La instrucción que permite detener completamente las iteraciones de un bucle es:</p> <p>a) stop.</p> <p>b) break.</p> <p>c) continue.</p> <p>d) finish.</p>	<p>3.4. La instrucción que permite detener la iteración actual de un bucle, continuando con la siguiente, si procede, es:</p> <p>a) stop.</p> <p>b) break.</p> <p>c) continue.</p> <p>d) finish.</p>
<p>3.5. De un bucle <i>do-while</i>, cuya condición depende de una serie de variables que en el bloque de instrucciones no se modifican, se puede afirmar:</p> <p>a) Que su número de iteraciones será siempre una.</p> <p>b) Que el número de iteraciones será siempre par.</p> <p>c) Que las variables cambiarán automáticamente en cualquier momento.</p> <p>d) Ninguna de las opciones anteriores es correcta.</p>	<p>3.6. ¿Cuántas veces se ejecutará el bloque de instrucciones del bucle más interno del siguiente fragmento de código?</p> <pre>for (i=1; i<10; i++) { for (i=1; i<=5; i++) { System.out.println("Hola"); } }</pre> <p>a) 10 veces.</p> <p>b) 5 veces.</p> <p>c) 50 veces.</p> <p>d) Infinitas veces.</p>
<p>3.7. ¿Qué sucede si en un bucle <i>while</i> la condición nunca deja de ser verdadera?</p> <p>a) El programa termina normalmente.</p> <p>b) El bucle se ejecuta solo una vez.</p> <p>c) El bucle se convierte en infinito.</p> <p>d) Se produce un error de compilación.</p>	<p>3.8. En cada iteración, el incremento de un bucle <i>for</i> se ejecuta:</p> <p>a) En primer lugar.</p> <p>b) Después de la inicialización.</p> <p>c) Después de evaluar la condición.</p> <p>d) Justo al finalizar cada iteración.</p>
<p>3.9. Una variable que se declara dentro de su bloque de instrucciones solo se podrá utilizar:</p> <p>a) En cualquier parte del programa.</p> <p>b) En todos los bucles.</p> <p>c) Dentro del bloque de instrucciones donde se ha declarado.</p> <p>d) Todas las opciones anteriores son correctas.</p>	<p>3.10. En un bucle <i>for</i>, la inicialización, condición e incremento son:</p> <p>a) Todos obligatorios.</p> <p>b) Todos opcionales.</p> <p>c) La inicialización siempre es obligatoria.</p> <p>d) La condición siempre es obligatoria.</p>



C.F.G.S. Desarrollo de Aplicaciones Web/Multiplataforma

MÓDULO: Programación

IES CAMAS Curso 2025/26



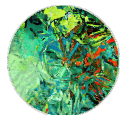
C.F.G.S. Desarrollo de Aplicaciones Web/Multiplataforma

MÓDULO: Programación

IES CAMAS Curso 2025/26

Notas

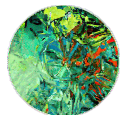
	2'5 puntos							2'5 puntos							5 puntos											
	Ejercicio 1							Ejercicio 2							Ejercicio 3										Total	
	a	b	c	d	e	f	g	a	b	c	d	e	f	g	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	3.10		
ialcrod1802	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	7,0	ialcrod1802
jblagon1804	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	2,5	jblagon1804
jcalgar480	0,75	1	0	1	1	1	1	0,75	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	8,6	jcalgar480
lcascan0310																									0,0	lcascan0310
acobfra1611	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	1	0	8,0	acobfra1611
mdiajim407	0,5	1	1	1	1	1	1	0,5	1	0	1	1	1	0,5	1	1	1	1	0	0	1	1	1	0	8,0	mdiajim407
adorcum696	0,75	1	0	1	1	0,75	1	0,75	1	0	1	1	0,75	1	0	1	1	1	1	0	1	0	1	0	7,2	adorcum696
rfrabar2206	0,75	1	0	1	1	1	1	0,75	1	0	1	1	1	1	1	0	1	1	0	1	1	0	0	0	7,1	rfrabar2206
ggarlob532	0,75	1	1	0,5	1	1	0,5	0,75	1	0	0,5	1	1	0,5	1	1	1	1	1	0	1	1	1	0	8,1	ggarlob532
sherace1405	0,75	1	0	1	1	1	1	0,75	1	0	1	1	1	1	1	0	1	1	1	0	1	1	1	0	8,1	sherace1405
mhuecob081	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	9,0	mhuecob081
gllepen645	1	1	1	1	1	1	1	1	1	1	0,5	1	1	1	1	1	1	1	0	1	1	1	1	0	8,9	gllepen645
amacpac0204	0,5	0,5	0	0	0	0	0	0,5	0,5	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	2,3	amacpac0204
amarloz3112	0,75	1	1	1	1	1	1	0,75	1	1	1	1	1	1	1	0	1	1	0	0	1	1	1	0	8,0	amarloz3112
amarboc	0,75	1	0	0,5	1	1	0,5	0,75	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	0	6,9	amarboc
rrodsan855	0,5	1	0	1	1	0,5	0,75	0,5	1	0	1	1	0,5	0,75	0	0	0	0	1	0	1	0	1	0	5,0	rrodsan855
asanesp2907																									0,0	asanesp2907
jsersan1204	1	1	1	1	1	1	1	1	1	1	1	1	0,75	1	1	0	1	0	1	0	1	0	1	0	7,3	jsersan1204
msieufa855	0,5	1	0	0,5	1	0	0	0,5	1	0	0	0,5	0,5	0,5	1	0	1	1	1	1	1	0	0	0	5,0	msieufa855
ptormar1403	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	0	8,0	ptormar1403



Rúbrica

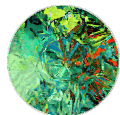
Ejercicio 1 y 2.

Escala de logro					
0% - No logrado 50% - Básico 75% - Competente 100% - Excelente					
RA	Criterio de evaluación	0% No logrado	50% Básico	75% Competente	100% Excelente
RA1	a) Se han identificado los bloques que componen la estructura de un programa informático.	No identifica las partes del programa.	Reconoce solo partes básicas (p. ej., <code>main</code> o variables).	Identifica correctamente las secciones principales.	Explica con precisión la estructura completa y su función.
RA1	d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.	No distingue tipos de variables.	Usa variables sin justificar su tipo.	Usa tipos correctos con leves errores.	Declara y justifica cada variable con precisión.
RA1	i) Se han introducido comentarios en el código.	Sin comentarios o irrelevantes.	Comentarios mínimos o poco claros.	Comentarios adecuados en puntos clave.	Comentarios precisos, claros y útiles.
RA2	b) Se han escrito programas simples.	No plantea solución funcional.	Solución incompleta o con errores lógicos.	Solución correcta con leves fallos.	Programa completo, bien estructurado y correcto.
RA3	a) Se ha escrito código que haga uso de estructuras de selección.	No usa condicionales.	Condicionales básicas con errores.	Condicionales correctas con lógica adecuada.	Condicionales completas y correctamente anidadas.
RA3	b) Se han utilizado estructuras de repetición.	No aplica bucles.	Bucles incorrectos o incompletos.	Bucles correctos y control básico de iteraciones.	Uso adecuado de <code>for</code> , <code>while</code> o <code>do-while</code> según la lógica.
RA3	e) Se han creado programas ejecutables utilizando diferentes estructuras de control.	No integra estructuras de control.	Usa alguna estructura con errores o confusión.	Usa estructuras correctas en orden lógico.	Integra condicionales y bucles de forma fluida y coherente.

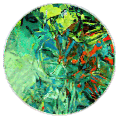


Comentarios

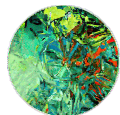
ialcrod1802	
jblagon1804	No me has entregado el ejercicio 1 y 2
jcalgar480	Has escrito Ejercicio2 {}, pero te ha faltado declarar la clase correctamente: <code>public class Ejercicio2 { ... }</code> . Faltan comentarios en el código.
lcascan0310	
acobfra1611	
mdiajim407	<ul style="list-style-type: none">- En Java, el código fuente debe escribirse dentro de una declaración <code>public class</code> cuyo nombre coincide con el nombre del archivo; es decir: <pre>public class NombreDeLaClase { // ... }</pre> y el fichero que contiene esa clase debe llamarse <code>NombreDeLaClase.java</code> .
adorcum696	<ul style="list-style-type: none">- En Java, el código fuente debe escribirse dentro de una declaración <code>public class</code> cuyo nombre coincide con el nombre del archivo; es decir: <pre>public class NombreDeLaClase { // ... }</pre> y el fichero que contiene esa clase debe llamarse <code>NombreDeLaClase.java</code> . <ul style="list-style-type: none">- Hay que realizar comentarios en el código.
rfrabar2206	Has escrito <code>class EjercicioUno;</code> , pero no es correcto: debe llevar llaves <code>{ }</code> . Faltan comentarios en el código.
ggarlob532	Aquí tienes una forma más clara y estructurada de expresar lo mismo: «Has escrito: <div><pre>class Ejercicio1; {</pre></div> Esa sintaxis es incorrecta. La declaración correcta debe ser <pre>public class Ejercicio1 {.</pre>



	<p>Además, no se indica el rango del método <code>main</code> con corchetes (<code>[]</code>).</p> <p>Por último, el programa que has escrito solo informa si cada número introducido por el usuario es positivo, negativo o cero; no realiza la tarea solicitada de contar cuántos valores positivos, negativos y ceros hay en total.*</p>
sherace1405	
mhuecob081	
gllepen645	<p>Has escrito:</p> <pre>int n=1; while(n !=0) { System.out.println("Introduzca un número(0 para salir:"); n = nextInt(); n += n }</pre> <p>«El código que has escrito no funciona como esperas:</p> <p><code>n += n</code> sobrescribe el valor introducido, por lo que pierdes la suma acumulada.</p> <p>Además, la variable <code>n</code> se usa tanto para leer como para acumular.</p> <p>Para contar o sumar números deberías usar una segunda variable (por ejemplo, <code>total</code>) y actualizarla dentro del bucle.</p>
amacpac0204	<p>El código tiene errores de sintaxis y estructura que impiden su ejecución. Debes repasar cómo declarar variables, usar correctamente <code>Scanner</code>, <code>if/else</code> y <code>while</code>, e incluir comentarios. Con práctica y atención a los detalles mejorarás rápidamente</p>
amarloz3112	<p>Has puesto los <code>import</code> dentro de la clase pública, y eso no es correcto.</p>
amarboc	<p>Has colocado la declaración <code>public class</code> dentro del método <code>public static void main</code>; eso no es correcto.</p> <p>El ejercicio 1</p> <p>En el fragmento que has escrito:</p> <pre>while (iteraciones < 10) { iteraciones++; // se incrementa antes de leer num = sc.nextInt(); // se lee un número System.out.println("Introduce otro número");</pre>



	<div><pre>}</pre></div> <p>Incremento prematuro</p> <ol style="list-style-type: none">1. Al aumentar <code>iteraciones</code> al principio del bucle, la condición <code>iteraciones < 10</code> ya no garantiza que el cuerpo se ejecute diez veces exactas; en realidad lo hace $n-1$ veces si empieza en 0.2. Orden de las operaciones<ul style="list-style-type: none">• Primero lees un número con <code>nextInt()</code>.• Luego imprimes el mensaje "Introduce otro número".3. En la última iteración, después de que se imprime el mensaje, el bucle termina y no vuelve a leer ningún número; por eso el último valor leído no tiene asociado una entrada del usuario.
rodsan855	<p>En Java, el código fuente debe escribirse dentro de una declaración <code>public class</code> cuyo nombre coincide con el nombre del archivo; es decir:</p> <pre>public class NombreDeLaClase { // ... }</pre> <p>y el fichero que contiene esa clase debe llamarse <code>NombreDeLaClase.java</code>.</p> <p>Has escrito:</p> <pre>while (n < 10) { ... int positivo = 0, negativo = 0, cero; ... }</pre> <p>las variables <code>positivo</code>, <code>negativo</code> y <code>cero</code> se crean <i>en cada iteración</i> del bucle.</p> <p>Por lo tanto, en la siguiente vuelta sus valores vuelven a ser 0 (o el valor que les asignes), es decir, los datos de la pasada anterior se pierden.</p>
asanesp2907	
jsersan1204	



msieufa855	Has comprendido la idea del ejercicio y el uso de condicionales, pero tu programa no funciona por errores de sintaxis y falta de variables de conteo. Debes repasar la estructura del bucle for, la declaración de variables y el uso correcto de if/else.
ptormar1403	