

UD2Tarea6 - Bucle For

Andrés Pérez Guardiola - 1st DAW

1. Identifica los errores de sintaxis de los siguientes bucles for.

a.

```
public class Main
{
    public static void main(String[] args) {

        for (int k = 5, k < 100, k++)
            System.out.println(k);

    }
}
```

El error de sintaxis es que los elementos dentro del condicional del bucle for deben ir separados con ; ya que son sentencias.

b.

```
public static void main(String[] args) {

    for (int k = 0 ; k < 12; k--)
        System.out.println(k);

}
```

Esto produciría un bucle infinito ya que no estamos incrementando como para romper la condición sino vamos retrocediendo desde cero y la condición $k < 12$ siempre será true.

2. ¿Cuál de los siguientes bucles obtiene una ejecución infinita? Corrígelos.

a.

```
for (int k = 0; k < 100; k = k)
    System.out.println(k);
```

Esto no actualiza el valor de k en cada iteración, produciría un bucle infinito. Corrección:

```
for (int k = 0; k < 100; k++)
    System.out.println(k);
```

b.

```
for (int k = 1; k == 100; k = k + 2)
    System.out.println(k);
```

Este produciría un bucle infinito porque la condición `k == 100` se la salta. Se podía solucionar modificando la condición:

```
for (int k = 1; k <= 100; k = k + 2)
    System.out.println(k);
```

c.

```
for (int k = 10; k > 0; k--)
    System.out.print(k + " ");
```

Esta correcto

d.

```
for (int k = 0; k < 100; k--)
    System.out.print(k + " ");
```

La condición nunca se cumple porque la variable k no se incrementa, decrementa.

Corregido:

```
for (int k = 0; k < 100; k++)
    System.out.print(k + " ");
```

e.

```
for (int k = 100; k != 6; k=k/2)
    System.out.print(k + " ");
```

El bucle termina cuando k es 6, por suerte se cumple en este caso, pero siendo una división hubiera puesto `k > 6` como condición.

f.

```
for (int k = 1; k != 100; k+=2)
    System.out.print(k + " ");
```

Este bucle no termina, se salta la condición. Se debería de cambiar por `k < 100`:

```
for (int k = 1; k < 100; k+=2)
    System.out.print(k + " ");
```

g.

```
for (int k = 1; k != 10; k=k/2)
    System.out.print(k + " ");
```

Este bucle tampoco tiene sentido, la variable k vale cero después de una iteración y nunca se rompe la condición.

```
for (int k = 1; k != 10; k= k + 1)
    System.out.print(k + " ");
```

3. Sin utilizar el IntelliJ, ¿qué valor tendrá j cuando el siguiente bucle termina?

```
for (int i = 0; i < 10; i++) {  
    int j;  
    j = j + 1;  
}
```

J es una variable privada definida dentro del bloque for. Esta deja de existir una vez termina el bucle de ejecutarse.

4. Escribe un código equivalente al siguiente utilizando un bucle for.

```
public class Main
{
    public static void main(String[] args) {
        int suma = 0;

        for (int i = 0 ; i <= 5; i++)
        {
            suma = suma + i;
            System.out.println(i + " ");
        }

        System.out.println("Suma es: " + suma);
    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\j
0
1
2
3
4
5
Suma es: 15

Process finished with exit code 0
```

5. Utiliza un bucle for para escribir las siguientes secuencias. Haz la traza de una de las dos:

a. 1 3 5 7 9

```
public class Main
{
    public static void main(String[] args) {

        for (int i = 1 ; i <= 10; i = i + 2)
        {
            System.out.print(i + " ");

        }

    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\
1 3 5 7 9
Process finished with exit code 0
```

b. 1 2 4 8 16

```
public class Main
{
    public static void main(String[] args) {

        for (int i = 1 ; i <= 16; i = i * 2)
        {
            System.out.print(i + " ");

        }

    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\j
1 2 4 8 16
Process finished with exit code 0
```

6. Escribe un programa que incremente la variable contador tantas veces como un incremento introducido por el usuario. De forma que sea, $\text{contador} = \text{contador} + \text{incremento}$.

```
import java.util.Scanner;

public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Con que numero quieres que incremente?
");
        int incremento = sc.nextInt();

        for (int i = 0; i < 100; i = i + incremento) {
            System.out.print("\t" + i);
        }

    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi
Con que numero quieres que incremente? 4
    0  4  8  12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 92 96
Process finished with exit code 0
```

7. Teniendo en cuenta el siguiente código:

```
public static void main(String[] args) {  
    int contador;  
  
    for(contador = 10; contador >= 0; contador--)  
    {  
        System.out.println("Contador es: " + contador);  
    }  
  
    System.out.println("\nBucle acaba.\nContador ahora es " +  
contador);  
}
```

¿La salida de este programa sería diferente si la expresión `contador--` se cambiara a `--contador`?

Sería lo mismo ya que el contador se incrementa de igual forma. La diferencia entre poner un predecremento o postdecremento es que el valor de contador se usa previamente o posteriormente a realizar el decremento, pero en este caso solo nos interesa reducir el valor en uno.

8. Sintácticamente un bucle consta de

```
for(inicializador; condición; modificador) {  
    //cuerpo del bucle  
}
```

a. ¿Se puede omitir la parte modificador?

Se permite, pero el bucle no tendría final si no se incluye este dentro del cuerpo del bucle.

b. ¿Y la parte inicializador y condición?

Un bucle for necesita un iniciador y condición, puesto que el modificador no estaría declarado en el contexto y daría error sintáctico.

9. ¿Es correcto el siguiente código?

```
import java.util.Scanner;

public class Main
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double i;
        System.out.print("Introduzca un valor o -1 para salir: ");

        i = scanner.nextDouble();
        for ( ; i >= 0.0 ; ) {
            System.out.println("La raíz cuadrada de " + i + " es "
+
                                Math.sqrt(i));
            System.out.print("Introduzca un valor o -1 para salir:
");
            i = scanner.nextDouble();
        }
    }
}
```

El siguiente código ha sacado las sentencias de iniciación y la de actualización de la declaración for y las ha colocado en otras partes del código. El código funciona, por lo que es posible.

El iniciador se ha colocado previo al bucle, por lo que ahora el valor *i* sobrevive una vez termina el bucle. Luego, el valor se actualiza cada vez que se pregunta dentro del bucle nuevamente y solo termina cuando se introduce un valor negativo en este.

Si se eliminara la condición el bucle no terminaría nunca pero sintácticamente sigue siendo correcto.

```
import java.util.Scanner;

public class Main
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double i;
        System.out.print("Introduzca un valor o -1 para salir: ");

        i = scanner.nextDouble();
        for ( ; ; ) {
            System.out.println("La raíz cuadrada de " + i + " es "
+
                                Math.sqrt(i));
            i = scanner.nextDouble();
        }
    }
}
```

```

        Math.sqrt(i));
        System.out.print("Introduzca un valor o -1 para salir:
");
        i = scanner.nextDouble();
    }

}
}

```

Se podría colocar la condición como un condicional (if) y un break para romper el bucle si se cumple.

```

import java.util.Scanner;

public class Main
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double i;
        System.out.print("Introduzca un valor o -1 para salir: ");

        i = scanner.nextDouble();
        for ( ; ; ) {
            if( i < 0.0 )
            {
                break;
            }
            System.out.println("La raíz cuadrada de " + i + " es "
+
                Math.sqrt(i));
            System.out.print("Introduzca un valor o -1 para salir:
");
            i = scanner.nextDouble();
        }

    }
}

```

10. ¿Es correcto el siguiente código?

```
public class Main
{
    public static void main(String[] args) {
        int suma = 0;

        for (int j = 0; j < 8; j++)
        {
            suma = suma + j;
        }

        System.out.println("La suma es: " + suma);
    }
}
```

El código es correcto, este imprime el valor de la suma de todos los números desde el iniciador hasta que se deja de cumplir la condición. j vale 7 al final del bucle pero deja de tener valor una vez sale del mismo porque el scope está dentro del bucle for. Por tanto, no puedo pintar el valor de j una vez terminamos el bucle.

11. ¿Es correcto el siguiente código? Corrígelo.

```
public class Main
{
    public static void main(String[] args) {
        int sumEven = 0;
        int sumOdd = 0;
        for ( int j = 0; j < 8; j= j + 2 )
            sumEven = sumEven + j;

        System.out.println(sumEven);

        for ( int j = 1; j < 8; j= j + 2 )
            sumOdd = sumOdd + j;

        System.out.println(sumOdd );
    }
}
```

Se ha corregido que ahora j se declara e inicializa en el segundo bucle for. j se declara dentro del primer bucle for, pero el scope termina dentro de ese bucle, es necesario declarar la nuevamente en el segundo.

12. Convierte el bucle do-while en un bucle for. El programa debe tener la misma lógica y realizar las mismas acciones.

```
import java.util.Scanner;

public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        String guess;

        for(boolean i = true ; i ; )
        {
            System.out.print("Guess my name: ");
            guess = sc.nextLine();

            i = !"Daffy Duck".equals(guess);
        }
    }
}
```

Entramos al bucle con un true, y actualizamos la condición al final. De esa forma emulamos el funcionamiento de un do-while.

A partir de una cantidad de dinero (sin decimales), mostrar su descomposición en billetes (500, 200, 100, 50, 20, 10, 5) y monedas (2, 1), para que el número de elementos sea mínimo. No se utilizar ninguna instrucción condicional. Por ejemplo, al introducir la cantidad 139 debe mostrar:

1 billete de 100

0 billete de 50

1 billete de 20

1 billete de 10

1 billete de 5

2 moneda de 2

```
import java.util.Scanner;

public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Introduce el valor de dinero a
despomer: ");
        int dinero = sc.nextInt();

        final int BILLETE100 = 100;
        int numBillete100 = dinero / BILLETE100;

        dinero = dinero % BILLETE100;
        System.out.println(numBillete100 + " billete de 100");

        final int BILLETE50 = 50;
        int numBillete50 = dinero / BILLETE50;

        dinero = dinero % BILLETE50;
        System.out.println(numBillete50 + " billete de 50");

        final int BILLETE20 = 20;
```

```

        int numBillete20 = dinero / BILLETE20;

        dinero = dinero % BILLETE20;
        System.out.println(numBillete20 + " billete de 20");

        final int BILLETE10 = 10;
        int numBillete10 = dinero / BILLETE10;

        dinero = dinero % BILLETE10;
        System.out.println(numBillete10 + " billete de 10");

        final int BILLETE5 = 5;
        int numBillete5 = dinero / BILLETE5;

        dinero = dinero % BILLETE5;
        System.out.println(numBillete5 + " billete de 5");

        final int MONEDA2 = 2;
        int numMoneda2 = dinero / MONEDA2;

        dinero = dinero % MONEDA2;
        System.out.println(numMoneda2 + " moneda de 2");

        final int MONEDA1 = 1;
        int numMoneda1 = dinero / MONEDA1;

        dinero = dinero % MONEDA1;
        System.out.println(numMoneda1 + " moneda de 1");

    }
}

```

```

"C:\Program Files\Java\jdk-22\bin\java.exe" "-ja
Introduce el valor de dinero a despomponer: 139
1 billete de 100
0 billete de 50
1 billete de 20
1 billete de 10
1 billete de 5
2 moneda de 2
0 moneda de 1

Process finished with exit code 0

```


No sé como hacerlo con un bucle for sin usar arrays. La forma propuesta también genera mucho código.

```
import java.util.Scanner;

public class Contador {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Introduce el valor de dinero a
despomer: ");
        int dinero = sc.nextInt();

        final int MONEDA1 = 1;
        final int MONEDA2 = 2;
        final int BILLETE5 = 5;
        final int BILLETE10 = 10;
        final int BILLETE20 = 20;
        final int BILLETE50 = 50;
        final int BILLETE100 = 100;

        int numMoneda1 = 0;
        int numMoneda2 = 0;
        int numBillete5 = 0;
        int numBillete10 = 0;
        int numBillete20 = 0;
        int numBillete50 = 0;
        int numBillete100 = 0;

        for (; dinero > BILLETE100; ) {
            dinero = dinero - BILLETE100;
            numBillete100++;
        }
        System.out.println(numBillete100 + " billete de 100");

        for (; dinero > BILLETE50; ) {
            dinero = dinero - BILLETE50;
            numBillete50++;
        }
        System.out.println(numBillete50 + " billete de 50");

        for (; dinero > BILLETE20; ) {
            dinero = dinero - BILLETE20;
            numBillete20++;
        }
        System.out.println(numBillete20 + " billete de 20");

        for (; dinero > BILLETE10; ) {
```

```
        dinero = dinero - BILLETE10;
        numBillete10++;
    }
    System.out.println(numBillete10 + " billete de 10");

    for (; dinero > BILLETE5; ) {
        dinero = dinero - BILLETE5;
        numBillete5++;
    }
    System.out.println(numBillete5 + " billete de 5");

    for (; dinero > MONEDA2; ) {
        dinero = dinero - MONEDA2;
        numMoneda2++;
    }
    System.out.println(numMoneda2 + " moneda de 2");

    for (; dinero > MONEDA1; ) {
        dinero = dinero - MONEDA1;
        numMoneda1++;
    }
    System.out.println(numMoneda1 + " moneda de 1");

    }
}
```

El sistema propuesto genera también mucho código y que se repite.