

UD2Tarea7 - Bucles For anidados

Andrés Pérez Guardiola - 1st DAW

1. ¿Es correcto el siguiente código? ¿Por qué?

```
int a = 0;
int b = 9;
if (a != b) {
    continue;
}
```

No se puede utilizar la sentencia continue; fuera de un bucle. Esto no tiene sentido puesto que esta palabra reservada sirve para que un bucle salte dicho paso del bucle y pase a la siguiente iteración, pero aquí no hay ningún bucle

2. Sin usar el IntelliJ, ¿qué imprime el siguiente for anidado?

```
for (int i=0; i<4; i++) {
    for (int j=i; j>=0; j--)
        System.out.print(j);
    System.out.println();
}
```

El bucle externo va incrementandose desde 0 hasta 4. En cada paso, el bucle for interno inicia desde el valor de i del bucle externo. El primer paso j = 0, pinta 0 en consola y al modificarse j a -1 deja de cumplirse la condición.

En las siguientes iteraciones, ej. i = 3, j se inicializa como 3, y disminuye hasta llegar a -1 donde rompe el bucle. Pintaría en pantalla 3210.

Output:

```
0
10
210
3210
```

3. Sin usar IntelliJ, ¿qué imprime el siguiente for anidado? Realiza la traza del for exterior.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j < i; j++) {  
        if(i == 4)  
            break;  
        System.out.print(j);  
    }  
    System.out.println( );  
}
```

El bucle exterior empieza con i valiendo 1 y termina cuando valga 6 (no se cumple la condición). Cada iteración suma 1 al valor de i. El bucle pinta un salto de línea cuando se ejecuta.

El bucle interno empieza con j valiendo 1 y termina cuando j vale más que el valor de i. Cada iteración incrementa el valor de j en 1. El bucle pinta el valor de j en consola cuando se ejecuta.

Hay una condición extra dentro del bucle interno, ya que cuando i = 4, saltará una sentencia break; que romperá el bucle interno, pero el bucle externo seguirá su curso con i = 5.

Si i vale 1, j valdrá 1 y el bucle interno no se ejecutará porque la condición es que sea menor que j. Solo se pintará la línea vacía al final.

Si i vale 4, el bucle interno no se ejecutará porque saltará el break. Se pintará una línea vacía.

Si i vale 2, 3 o 5, el bucle pintará los números desde 1 hasta el valor de i (no inclusive) en la terminal.

Output:

1
12

1234

4. Escriba un programa que imprima una tabla de símbolos. El usuario deberá introducir los valores de alto y ancho.

Ejemplo:

- Inserta la altura: 2
- Inserta la anchura: 3

```
import java.util.Scanner;

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

        System.out.print("Inserta la altura: ");
        int altura = sc.nextInt();
        System.out.print("Inserta la anchura: ");
        int anchura = sc.nextInt();

        for (int i = 0; i < altura; i++)
        {
            for (int j = 0; j < anchura; j++)
            {
                System.out.print('#');
            }
            System.out.println();
        }
    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\java.exe
Inserta la altura: 2
Inserta la anchura: 3
###
###

Process finished with exit code 0
```

5. Escribe un bucle for anidado por cada uno de los siguientes patrones:

a)

```
public class Main
{
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                System.out.print('*');
            }
            System.out.println();
        }
    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\jav
*
**
***
****
*****

Process finished with exit code 0
```

b)

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

    for (int i = 1; i <= 5 ; i++)
    {
        for (int j = 1; j < i ; j++)
        {
            System.out.print(' ');
        }
        System.out.println('*');
    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\
*
*
*
*
*

Process finished with exit code 0
```

c)

```
public static void main(String[] args) {
    for (int i = 1; i <= 5; i++)
    {
        for (int j = i; j > 0 ; j--)
        {
            System.out.print(j + " ");
        }
        System.out.println();
    }
}
```

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

Process finished with exit code 0
```

d)

```
public static void main(String[] args) {
    for (int i = 1; i <= 5; i++)
    {
        for (int j = 5; j > i ; j--)
        {
            System.out.print(' ');
        }
        for(int j = 1; j <= i; j++)
        {
            System.out.print('*');
        }
        System.out.println();
    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\java
```

```
  *
```

```
 **
```

```
***
```

```
****
```

```
*****
```

```
Process finished with exit code 0
```

6. Imprime la siguiente figura. Ayúdate usando bucles anidados.

```
public class Main
{
    public static void main(String[] args) {
        for (int i = 1; i <= 3 ; i++)
        {
            if(i == 1 || i == 3) System.out.print('+');
            else System.out.print('|');

            for (int j = 1; j <= 10 ; j++)
            {
                if( (i == 1 || i == 3) && j % 2 != 0)
System.out.print('<');
                else if( (i == 1 || i == 3) && j % 2 == 0)
System.out.print('>');
                else System.out.print(' ');

            }

            if(i == 1 || i == 3) System.out.println('+');
            else System.out.println('|');
        }
    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\j
+<><><><><>+
|           |
+<><><><><>+

Process finished with exit code 0
```

7. Imprime la siguiente figura utilizando bucles anidados. Ayúdate de las posiciones pares e impares para imprimir * ó 0.

```
public class Main
{
    public static void main(String[] args)
    {
        for (int i = 1; i <= 5; i++)
        {
            for (int j = 1; j <= i ; j++)
            {
                if (j % 2 == 0) System.out.print("0 ");
                if (j % 2 != 0) System.out.print("* ");

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

```
"C:\Program Files\Java\jdk-22\bin\jav
*
* 0
* 0 *
* 0 * 0
* 0 * 0 *

Process finished with exit code 0
```


8. Realiza un programa que imprima los 50 primeros números primos en diferentes líneas de consola. Recuerda que un número primo es aquel que solo tiene dos divisores: el 1 y el mismo.

Por ejemplo 7 es primo porque si dividimos 7 entre todos los números del {1 al 7} solo tiene resto 0 el 1 y el 7:

$7/1 = 7$ resto 0

$7/2 = 3$ resto 1

$7/3 = 2$ resto 1

$7/4 = 1$ resto 3

$7/5 = 1$ resto 2

$7/6 = 1$ resto 1

$7/7 = 1$ resto 0

Ayuda: utiliza un bucle exterior para recorrer los 50 números. Un bucle interno para recorrer los números anteriores al del bucle externo. Dentro del bucle interno utiliza un if para comprobar si es primo o no el número. Si el número no es primo utiliza una sentencia break etiquetada para salir del primer bucle y seguir con el siguiente número de los 50.

(sentencia break no, continue si. El objetivo es que recorra todos los números no que me haga un break del bucle exterior y no haga nada más)

```

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

        bucleNumerosPrimos:
        for (int i = 2; i <= 50; i++)
        {
            // Ningún número par es primo excepto 2
            if ( (i % 2 == 0) && (i != 2)) continue;

            for (int j = 2; j < i; j++)
            {
                if ( i % j == 0)
                {
                    continue bucleNumerosPrimos;
                }
            }

            System.out.println("Primo: "+ i);
        }
    }
}

```

Por cada número primo se revisa si el resto es igual a cero o no. Los números primos solo deberían tener resto igual a cero para los casos $\text{num} \% 1$ y $\text{num} \% \text{num}$. A parte, todos los números pares que no sean dos no pueden ser primos porque son todos divisibles por dos.

```
"C:\Program Files\Java\jdk-22\bin\java.exe
Primo: 2
Primo: 3
Primo: 5
Primo: 7
Primo: 11
Primo: 13
Primo: 17
Primo: 19
Primo: 23
Primo: 29
Primo: 31
Primo: 37
Primo: 41
Primo: 43
Primo: 47

Process finished with exit code 0
```

Se podría optimizar seguramente también en el bucle interno, porque revisar desde 2 hasta el propio valor de i estamos perdiendo muchas iteraciones donde, por propiedades de los números, no pueden existir divisores.