

Ciclos anidados

Ciclos anidados	1
¿Qué aprenderás?	2
Introducción	2
Escribiendo tablas de multiplicar	3
Tabla de un número	3
Tabla de todos los números	3
Dibujando con ciclos anidados	4
Ejercicio 1 - Cuadrado	4
Ejercicio guiado: Escalera	5



¡Comencemos!

¿Qué aprenderás?

- Implementar ciclos anidados para resolver problemas.

Introducción

Un ciclo anidado no es nada más que un ciclo dentro de otro ciclo. No existe ningún límite en torno a cuántos ciclos pueden haber anidados dentro de un código, aunque cada uno aumentará la complejidad temporal del programa. Los ciclos anidados expanden el universo de los tipos de problemas que podemos resolver.

¡Vamos con todo!



Escribiendo tablas de multiplicar

A modo de ejemplo, para poner en práctica algunos ciclos anidados, vamos a utilizar un patrón conocido como lo son las tablas de multiplicar.

Tabla de un número

Supongamos que queremos mostrar una tabla de multiplicar. Por ejemplo, la tabla del número 5.

```
for(int i=0;i<11;i++) {  
    System.out.printf("5 * %d = %d\n", i,5*i);  
}
```

¿Cómo podríamos hacer para mostrar todas las tablas de multiplicar del 0 al 10? ¡Fácil! Envolviendo el código en otro ciclo que itere de 0 a 11.

Tabla de todos los números

```
for(int j=0;j<11;j++) {  
    for(int i=0;i<11;i++) {  
        System.out.printf("%d * %d = %d\n",j, i,j*i);  
    }  
}
```

Resultado

```
0 * 0 = 0  
0 * 1 = 0  
0 * 2 = 0  
0 * 3 = 0  
...  
...  
...  
9 * 6 = 54  
9 * 7 = 63  
9 * 8 = 72  
9 * 9 = 81
```

Dibujando con ciclos anidados

A continuación, nos centraremos en profundizar el uso de los ciclos y ciclos anidados, analizando distintos patrones para resolverlos de la manera adecuada. Los ejercicios que analizaremos son de ir dibujando patrones, ya que son ideales para practicar ciclos y ciclos anidados.

Ejercicio 1 - Cuadrado

Veamos el siguiente patrón:

```
*****  
*****  
*****  
*****  
*****
```

Si el usuario ingresa por ejemplo el 5, se dibujará un cuadrado de 5*5, donde cada elemento es un asterisco.

Solución

De manera muy similar cuando debíamos dibujar una línea con los asteriscos, ahora debemos extender esta línea a un plano.

Recordemos:

```
Scanner sc = new Scanner(System.in);  
int n = sc.nextInt();  
for(int i=0;i<n;i++) {  
    System.out.printf("*");  
}  
System.out.printf("\n");
```

Ahora, para extender `n` veces, debemos incorporar un nuevo ciclo for:

```
int n = sc.nextInt();
for(int i = 0; i < n; i++) {
    for(int j = 0; j < n; j++) {
        System.out.printf("*");
    }
    System.out.printf("\n");
}
```

Ejercicio guiado: Escalera

Ahora veamos este patrón:

```
*
**
***
****
*****
```

Solución

Para resolver el código tenemos que observar el patrón:

- Si el usuario ingresa 5, se dibujan 5 filas.
- En la fila 1, hay 1 asterisco.
- En la fila 2, hay 2 asteriscos.
- Por lo que podemos decir que en la fila `n` hay `n` asteriscos.
- **Paso 1:** Primer acercamiento a la solución, dibujaremos las `n` filas

```
for(int i = 0; i < n; i++) {
    System.out.printf("*\n");
}
```

- **Paso 2:** Dibujamos las columnas

```
for(int j = 0; j <= i; j++) {  
    System.out.printf("*");  
}  
System.out.printf("\n");  
/*Si estamos en la fila j = 3, correspondiente a la 4ta fila (las j parten  
en cero), debemos dibujar 4 asteriscos. Por ende la condición de borde  
queda dada  
por i <=j*/
```

Ahora, juntando lo anterior quedaría:

```
Scanner sc = new Scanner(System.in);  
System.out.printf("Ingresa un numero");  
int n = sc.nextInt();  
for(int i = 0; i < n; i++) {  
    for(int j = 0; j <= i; j++) {  
        System.out.printf("*");  
    }  
    System.out.printf("\n");  
}
```