

# Tema 6 – S4

## Contenidos:

### 4. Usando un bucle for

- ¿Cuándo? ¿Cómo? Mecánica del for
  - Bucles (iteraciones) anidadas
  - Ejemplos y ejercicios
- **Práctica 6:** Inclusión del error en los puzle de las funciones raíz cuadrada y logaritmo natural en el intervalo real  $[1/2, 1[$

Encontrarás los programas Java correspondientes a esta sesión en el package *sesion4* del proyecto BlueJ *ejercicios – Tema 6*, disponible en mi carpeta *Tema 6* de Recursos de PoliformaT

# Usando un bucle for ¿Cuándo? ¿Cómo?

- Mostrar por pantalla la tabla de multiplicar del 4:

repetir 10 veces la instrucción `System.out.println("4 x " + i + " = " + 4 * i);` con  $1 \leq i \leq 10$

```
Bluej: Bluej: Ventana de Terminal...
Tabla del 4
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40

System.out.println("Tabla del 4");
System.out.println("4 x " + 1 + " = " + 4 * 1);
System.out.println("4 x " + 2 + " = " + 4 * 2);
System.out.println("4 x " + 3 + " = " + 4 * 3);
System.out.println("4 x " + 4 + " = " + 4 * 4);
System.out.println("4 x " + 5 + " = " + 4 * 5);
System.out.println("4 x " + 6 + " = " + 4 * 6);
System.out.println("4 x " + 7 + " = " + 4 * 7);
System.out.println("4 x " + 8 + " = " + 4 * 8);
System.out.println("4 x " + 9 + " = " + 4 * 9);
System.out.println("4 x " + 10 + " = " + 4 * 10);
```

```
public class TablaDel4 {
    public static void main(String[] args) {
        System.out.println("Tabla del 4");
        // REPETIR DESDE i=1 HASTA i=10 (i.e. 10 veces) ...
        for (int i = 1; i <= 10; i++) {
            System.out.println("4 x " + i + " = " + 4 * i);
        }
    }
}
```

# Usando un bucle for

## ¿Cuándo? ¿Cómo? Mecánica del for (I)



BlueJ: ejercicios S4 – Tema 6

- **Sitúa**, como en la imagen, un punto de ruptura en la línea 14 de TablaDel4

```
8 |
9 | public class TablaDel4 {
10 |     public static void main(String[] args) {
11 |
12 |         System.out.println("Tabla del 4");
13 |
14 |         for (int i = 1; i <= 10; i++) {
15 |             System.out.println("4 x " + i + " = " + 4 * i);
16 |         }
17 |     }
18 | }
```

Clase compilada - no hay errores de sintaxis    guardado

Luego, **traza** la ejecución del `main` con ayuda del depurador de BlueJ y **observa** cómo se actualizan las variables `i` (contador) y `4 * i` (acumulador del resultado) en cada repetición (*Step*)

1. ¿Cuánto valen `i` y `4 * i` al inicio?
2. ¿Cuándo se incrementa `i`, antes o después de ejecutar el `System.out.println`?
3. ¿Qué valor tiene `i` al terminar el bucle, tras la última repetición? ¿Y `4 * i`?

A partir de lo visto, **responde**: ¿es correcto el bucle? ¿termina tras 10 repeticiones?

- **Inicializa** el contador del bucle `i` a 0, en lugar de a 1 como está ahora. Luego, **compila y ejecuta el programa** (sin depurador) y **responde**: ¿es correcto el bucle? ¿cuántas repeticiones hace?
- **Vuelve a dejar el código como estaba**. **Cambia** la guarda del bucle por `i < 10` en lugar de `i <= 10` como está ahora. Luego, **compila y ejecuta el programa** (sin depurador) y **responde**: ¿es correcto el bucle? ¿cuántas repeticiones se hacen?

# Usando un bucle for

## ¿Cuándo? ¿Cómo? Mecánica del for (II)

**Solución del puzle del Ejemplo 1 del tema (Sesión 1):** diseña un método que devuelva el producto de a y b, enteros no negativos, **SIN** usar el operador \*

```
// PRECONDICIÓN: a >= 0 AND b >= 0
public static int productoSinUsarX(int a, int b) {
    int i = 0; int res = 0;
    while (i != a) {
        res = res + b;
        i++;
    }
    return res;
}
```

En TestProductoSinUsarX  
de ejemplos S1 – Tema 6

**Completa** el cuerpo del siguiente método usando el bucle for **equivalente** al while que aparece en productoSinUsarX

```
// PRECONDICIÓN: a >= 0 AND b >= 0
public static int productoSinUsarXFor(int a, int b) {
    int res = 0;
    for (int i = 0; i != a; i++) {
        res = res + b;
    }
    return res;
}
```

Para resaltar **diferencias**, solo los cambios se muestran en tinta **negra**

# Usando un bucle for

## ¿Cuándo? ¿Cómo? Mecánica del for (III)

**Solución del puzle del Ejemplo 2 del tema (Sesión 1):** diseña un método que devuelva la suma de las cifras de *a*, entero no negativo, **SIN** usar `Math.log10`

En TestSumarCifras de  
ejemplos S1 – Tema 6

```
// PRECONDICIÓN: a >= 0
public static int sumarCifras(int a) {
    int res = 0; int i = a;
    while (i != 0) { // i > 0
        res = res + i % 10;
        i = i / 10;
    }
    return res;
}
```

**Completa** el cuerpo del siguiente método usando el bucle for **equivalente** al `while` que aparece en `sumarCifras`

```
// PRECONDICIÓN: a >= 0
public static int sumarCifrasFor(int a) {
    int res = 0;
    for (int i = 0; i > 0; i = i / 10) {
        res = res + i % 10;
    }
    return res;
}
```

Para resaltar **diferencias**, solo los cambios se muestran en tinta **negra**

# Usando un bucle for

## ¿Cuándo? ¿Cómo? Mecánica del for: Ejercicios propuestos

- **Nº 8 Transparencias:** ¿qué muestra por pantalla el siguiente bucle?

```
for (int cuenta = 1; cuenta < 5; cuenta++) {  
    System.out.println(2 * cuenta);  
}
```

- **Nº 9 Transparencias:** ¿qué muestra por pantalla el siguiente bucle?

```
for (int i = 0, j = 10; i < j; i++, j--) {  
    System.out.println("i: " + i + " j: " + j);  
}
```

- **Ejercicios 3 y 4 del capítulo 8 del libro de la asignatura**

- **Cuestión:** en el programa `Ejercicio2Capitulo8Libro` del proyecto `BlueJ soluciones ejercicios S1 – Tema 6` tienes 2 versiones de `sumarHasta`, un método que devuelve la suma de los n primeros números naturales ¿Cuál de los bucles `while` que aparecen en ellas traducirías a bucle `for`? ¿Por qué?

# Usando un bucle for

## Bucles Anidados: las tablas de multiplicar

En el programa TablasDeMultiplicar de *sesion4*

El programa TablaDe14 del package *sesion4* muestra por pantalla la tabla de multiplicar del 4

```
for (int j = 1; j <= 10; j++) {  
    System.out.println("Tabla del " + j);  
  
    for (int i = 1; i <= 10; i++) {  
        System.out.println(j + " x " + i + " = " + j * i);  
    }  
}
```

¿Cómo lo modificarías para que mostrara, una tras otra, las tablas de multiplicar del 1 al 10?

# Usando un bucle for

## Bucles Anidados: trazas

- ¿Qué muestra por pantalla el siguiente código?

```
int nfil = 4, ncol = 3;
for (int i = 1; i <= nfil; i++) {
    for (int j = 1; j <= ncol; j++) {
        System.out.print(i + "-" + j + " ");
    }
    System.out.println();
}
```

- ¿Y este?

```
for (int i = 0; i <= n; i++) {
    for (int j = 1; j <= i; j++) { System.out.print('Z'); }
    for (int j = 1; j <= n; j++) { System.out.print('A'); }
    for (int j = 1; j <= n - i; j++) { System.out.print('Z'); }
    System.out.println();
}
```

Ejecuta los métodos `traza1` y `traza2` de la clase `Trazas` (del package [sesion4](#)) y comprueba tus respuestas; para una mejor comprensión de cómo funcionan, puede resultar útil poner puntos de ruptura en las líneas de cada uno de los `for`



# Usando un bucle for

## Bucles Anidados: dibujar un rectángulo

Método del programa DibujaFiguras de *sesion4*

Dibuja en pantalla un rectángulo de base > 1 y altura > 1, usando asteriscos ('\*')

base	altura	dibujo
3	2	*** ***
6	3	***** ***** *****
4	5	***** ***** ***** ***** *****

### ALGORITMO

**DATOS:** altura (nº líneas a dibujar); base (nº de '\*' por línea)

### INSTRUCCIONES:

Repetir desde  $i = 1$  hasta  $i = \text{altura}$  (i.e. altura veces):

- Escribir línea, i.e. repetir desde  $j = 1$  hasta  $j = \text{base}$ : escribir '\*'
- Escribir salto de línea

```
// PRECONDICIÓN: base > 1 AND altura > 1
for (int i = 1; i <= altura; i++) {
    for (int j = 1; j <= base; j++) {
        System.out.print("*");
    }
    System.out.println();
}
```

# Usando un bucle for

## Bucles Anidados: dibujar un triángulo

Método del programa DibujaFiguras de *sesion4*

Dibuja en pantalla un triángulo rectángulo **isósceles** con altura > 1 dada, con '\*'

altura	dibujo
2	* **
3	* ** ***
4	* ** *** ****

### ALGORITMO

**DATOS :** altura (nº de líneas a dibujar)

### INSTRUCCIONES:

Repetir desde  $i = 1$  hasta  $i = \text{altura}$  (i.e. repetir altura veces):

- Escribir línea, o repetir desde  $j = 1$  hasta  $j = \text{base}$ : escribir '\*'
- Escribir salto de línea

```
// PRECONDICIÓN: altura > 1
for (int i = 1; i <= altura; i++) {
    for (int j = 1; j <= i; j++) {
        System.out.print("*");
    }
    System.out.println();
}
```

# Usando un bucle for

## Bucles Anidados: dibujar un cuadrado

Método del programa DibujaFiguras de *sesion4*

Dibuja en pantalla un cuadrado de lado  $l > 3$  dado, con '\*' y espacios en blancos. La siguiente figura muestra un ejemplo con  $l = 5$

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

### ALGORITMO

**DATOS :**  $l$  (nº de líneas a dibujar)

### INSTRUCCIONES:

1. Escribir línea con  $l$  asteriscos
2. Repetir desde  $i = 1$  hasta  $i = l - 2$  (i.e. repetir  $l - 2$  veces):
  - Escribir línea con  $1$  asterisco,  $l - 2$  blancos y  $1$  asterisco
  - Escribir salto de línea
3. Escribir línea con  $l$  asteriscos

```
// PRECONDICIÓN:  $l > 3$ 
for (int i = 1; i <= l; i++) { System.out.print("*"); }
System.out.println();
for (int i = 1; i <= l - 2; i++) {
    System.out.print("*");
    for (int j = 1; j <= l - 2; j++) { System.out.print(" "); }
    System.out.println("*");
}
for (int i = 1; i <= l; i++) { System.out.print("*"); }
System.out.println();
```

# Usando un bucle for

## Bucles Anidados: dibujar una figura de n líneas

Método del programa `DibujaFiguras` de *sesion4*

Dibuja en pantalla una figura con  $n > 1$  líneas tal que: cada línea tendrá  $n - 1$  asteriscos y el carácter 'a' en la posición diagonal principal de la figura. Por ejemplo, para  $n = 5$  se mostraría por pantalla lo siguiente:

a\*\*\*\*

**ALGORITMO**

\*a\*\*\*

**DATOS :** n (nº de líneas a dibujar)

\*\*a\*\*

**INSTRUCCIONES:**

\*\*\*a\*

Repetir desde  $i = 1$  hasta  $i = n$  (o repetir n veces): Escribir línea con...

\*\*\*\*a

- $i - 1$  asteriscos (i.e. desde  $j = 1$  hasta  $j = i - 1$ : escribir '\*')
- Una 'a' en posición  $j == i$  (o posición de la diagonal principal)
- $n - i - 1$  asteriscos (i.e. desde  $j = i + 1$  hasta  $j = n$ : escribir '\*')
- Escribir salto de línea

```
// PRECONDICIÓN:  $n > 1$ 
for (int i = 1; i <= n; i++) {
    for (int j = 1; j < i; j++) { System.out.print("*"); }
    System.out.print('a'); // en posición  $j == i$  dibujar 'a'
    for (int j = i+1; j <= n; j++) { System.out.print("*"); }
    System.out.println();
}
```

# Usando un bucle for

## Bucles Anidados - Ejercicio Nº 8 Transparencias

Dibuja en pantalla un paralelogramo de altura  $> 1$  y base  $> 1$  dadas. Por ejemplo, para altura = 4 y base = 9 se mostraría por pantalla lo siguiente:

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

### ALGORITMO

**DATOS:** altura (nº líneas a dibujar); base (nº de '\*' por línea)

### INSTRUCCIONES:

Repetir desde  $i = 1$  hasta  $i = \text{altura}$  (o repetir altura veces): Escribir línea con...

- Espacios en blanco...  $\text{nBlancos} = \text{altura} - i$
- base asteriscos (i.e. desde  $j = 1$  hasta  $j = \text{base}$ : escribir '\*')
- Escribir salto de línea

**PISTA:** para saber cuántos espacios en blanco poner en cada línea, cuenta los que se escriben en el ejemplo **empezando por su última línea**, la 4ª, **y subiendo hasta la primera**

# Usando un bucle for

## Bucles Anidados: dibujar la copa de un árbol

Dibuja en pantalla un triángulo isósceles de cierta base  $\geq 3$  e impar, con '\*' y blancos (' ')

¿Cuántas líneas se dibujan?

base	dibujo
3	<pre>  *  *↓* *</pre>
7	<pre>    *   ***  ***** *****</pre>
9	<pre>      *     ***   *****  ***** *****</pre>

### ALGORITMO

**DATOS:** base

### INSTRUCCIONES:

1. Calcular el nº de líneas a dibujar, o altura:  $(\text{base} + 1) / 2$
2. Repetir desde  $i = 1$  hasta  $i = \text{altura}$  (o repetir altura veces):  
Escribir línea con...
  - Espacios en blanco...  $n\text{Blancos} = \text{altura} - i$
  - Asteriscos...  $n\text{Asteriscos} = \text{base} - 2$
  - Escribir salto de línea

**PISTA:** para saber cuántos blancos y asteriscos poner en cada línea, cuenta los que se escriben en el **ejemplo con base = 7 empezando por su última línea**, la 4ª, y subiendo hasta la primera

# Ejercicios **CAP**: Bucles Anidados



## **Cubos hasta n (clave CCDII4ai)**

(hasta el 10-01-2018; solución desde el 5-12)



**Entregable #6.2** (15 puntos, no presencial, individual, hasta el 30-11; solución desde el 2-12):

- 1.- Dibujar Rectángulo con \* (clave CCDIJ4ai)**
- 2.- Dibujar Triángulo con \* (clave CCDIK4ai)**
- 3.- Dibujar Árbol de Navidad con \* (clave CCDIL4ai)**