UPN

UNIVERSIDAD
PRIVADA
DEL NORTE

FUNDAMENTOS DE ALGORITMOS

FUNDAMENTOS DE ALGORITMOS

Semana 5: Estructura repetitiva Para (For)

FUNDAMENTOS DE ALGORITMOS

LOGRO DE LA SESIÓN 5:

Al finalizar la sesión, el estudiante crea programas usando estructuras repetitivas for en la solución de problemas.

ESTRUCTURAS REPETITIVAS

Se denominan estructuras repetitivas o estructuras de repetición a aquellas estructuras que permiten repetir instrucciones. A las estructuras repetitivas se conocen también como estructuras iterativas o bucles, a las instrucciones a repetir se conocen como el cuerpo del bucle y al hecho de repetir la secuencia de instrucciones se denomina iteración. En el caso del lenguaje Java, tenemos tres tipos de estructuras repetitivas: las estructuras while, do...while y for.

ESTRUCTURA REPETITIVA FOR

La estructura de repetición while es una estructura de propósito general: puede usarse para resolver cualquier problema que involucre procesos repetitivos. Pero, para los casos en que se conoce el número de iteraciones de un proceso repetitivo, la estructura ideal es la estructura for. Fue construída para ese tipo de procesos y es por ello que puede manejar uno o más contadores propios. En la Figura 1, se muestra el diagrama de flujo de la estructura for

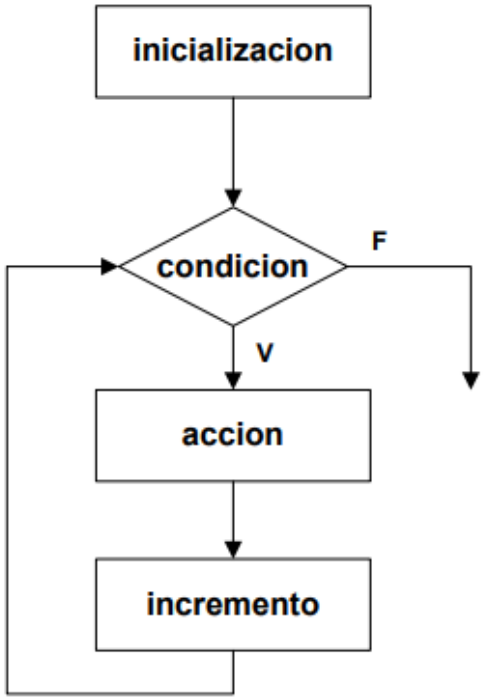

Figura 1 Diagrama de flujo de la estructura **for**

SINTAXIS DE LA ESTRUCTURA REPETITIVA FOR

Para una sóla acción por repetir:

```
for (inicio; condicion; incremento)
    accion;
```

Para más de una acción por repetir:

```
for (inicio; condicion; incremento) {
    accion1;
    accion2;
    .
    .
    .
    accionn;
}
```

FUNCIONAMIENTO DEL FOR

Paso 1:- Se ejecuta la sentencia de inicio. En la sentencia de inicio, se declara y se inicializa el contador del for.

Paso 2:- Se evalúa la condición. En caso de que la condición sea verdadera, va al paso 3; en caso contrario, finaliza el for.

Paso 3:- Se ejecuta la acción o el conjunto de acciones.

Paso 4:- Se ejecuta la sentencia de incremento y vuelve al paso 2. En la sentencia de incremento se incrementa el contador del for.

EJEMPLOS

- **for** que imprime los números: `0, 1, 2, 3, ..., 48, 49, 50`

```
for (int i = 0; i <= 50; i++)
    txtS.append(i + "\n");
```

- **for** que imprime los números: `100, 99, 98, ..., 13, 12, 11, 10`

```
for (int i = 100; i >= 10; i--)
    txtS.append( i + "\n" );
```

- **for** que imprime los números: `10, 12, 14, 16,  ..., 98, 99, 100`

```
for (int i = 10; i <= 100; i += 2)
    txtS.append(i + "\n");
```

- **for** que imprime los números: `100, 97, 94, 91, ..., 18, 15, 12, 9`

```
for (int i = 100; i >= 9; i -= 3)
    txtS.append(i + "\n");
```

CONTADORES Y ACUMULADORES

OPERADORES DE INCREMENTO Y DECREMENTO

Son operadores que permiten incrementar o decrementar en una unidad el valor de una variable numérica.

Operador	Uso	Equivalencia
++	a++;	a = a + 1;
--	a--;	a = a + 1;

Ejemplo 1

```
// Incrementa en uno el valor de x (Forma 1)
x = x + 1;

// Incrementa en uno el valor de x (Forma 2)
x++;

// Decrementa en 1 el valor de la variable z (Forma 1)
z = z - 1;

// Decrementa en 1 el valor de la variable z (Forma 2)
z--;
```

OPERADORES DE ASIGNACIÓN COMPLEJA

Son operadores que permiten asignar a una variable el valor de la variable mas, menos, por o entre el valor de otra variable.

Operador	Ejemplo	Equivalencia
+=	a += b;	a = a + b;
-=	a -= b;	a = a - b;
*=	a *= b;	a = a * b;
/=	a /= b;	a = a / b;

Ejemplo 2

```
// Incrementa en 2 el valor de la variable z (Forma 1)
z = z + 2;

// Incrementa en 2 el valor de la variable z (Forma 2)
z += 2;

// Decrementa en 5 el valor de la variable m (Forma 1)
m = m - 5;

// Decrementa en 5 el valor de la variable m (Forma 2)
m -= 5;
```

CONTADORES

Un ***contador*** es una variable que se utiliza para contar el número de ocurrencias de un suceso o el número de veces que se cumple una determinada condición. El proceso de conteo se efectúa, generalmente, de uno en uno y consiste en incrementar en 1 a la variable conteo cada vez que ocurre el evento o suceso que se pretende contar. Antes de iniciar el proceso de conteo, el contador debe ser inicializado en 0.

Por ejemplo, se necesita un **contador** para determinar:

- La cantidad de veces que se hizo clic en un botón
- La cantidad de notas ingresadas
- La cantidad de notas desaprobatorias
- La cantidad de notas desaprobatorias
- La cantidad de ventas efectuadas
- Etc.

Una instrucción de conteo tiene la siguiente forma:

```
contador = contador + 1;
```

Que puede escribirse también como:

```
contador++;
```

ACUMULADORES

Un ***acumulador*** es una variable que se utiliza para acumular o totalizar cantidades de una misma especie: sueldos, edades, pesos, etc. El proceso de acumulación consiste en incrementar la variable que sirve de acumulador en la cantidad que se pretende acumular. Antes de iniciar el proceso de acumulación, el acumulador debe ser inicializado en 0.

Por ejemplo, se necesita un **acumulador** para determinar:

- El sueldo total de los empleados de una empresa
- La suma total de las notas de un alumno
- La suma total de los pesos de un grupo de personas
- La suma total de las edades de un grupo de personas
- La cantidad total de unidades vendidas de un producto
- Etc.

Una instrucción de acumulación tiene la siguiente forma:

```
acumulador = acumulador + cantidad;
```

Qué puede escribirse también como:

```
acumulador +=  cantidad;
```

¿PREGUNTAS?

UPN

UNIVERSIDAD
PRIVADA
DEL NORTE