

Les estructures iteratives, o de repetició, permeten executar un mateix bloc d'instruccions diverses vegades, atenent a si es compleix o no una condició.

A les iteracions s'ha d'anar en compte a canviar les premisses de la condició per a que en algun moment s'acabi. Sinó hauréem creat un **bucle infinit** que bloquejarà el programa. Segons el comportament de les **variables de control** (les que participen a la condició) tindrém diferents tipus:

- Comptador. És una variable tipus enter que fa augmentant o disminuint. Ens marca clarament les iteracions a fer.
- Acumulador. Va guardant el total dels càlculs que es fan a la iteració per poder-ho comparar amb el valor de referència de la iteració.
- Semàfor/Bandera. Variable, generalment booleana, que funciona com un interruptor de si s'ha de sortir o quedar a la iteració.

### Iteració Mentre

```
mentre <condició> fes  
    accions;  
fmentre;
```

S'avalua la condició abans d'entrar a la iteració i es va repetint mentre es compleixi.

```
while (expressió booleana) {  
    // Instruccions  
}
```

### Estructura Repetir

```
repetir  
    accions;  
fins que <condició>;
```

En aquest cas s'avalua la condició al final del bloc d'instruccions. Per tant sempre s'executarà al manco una vegada.

```
do {  
    // Instruccions  
} while (expressió booleana);
```

### Estructura Per

```
per comptador=valor_ini fins a valor_fi fer (pas n)  
    accions;  
fper;
```

Aquesta il·lustració mostra una instrucció "per" típica que sols permet fer la iteració un nombre determinat de vegades. En Java veurem que la funcionalitat d'aquestes iteracions és molt més gran (i complexa).

```
for (inicialització comptador; <expressió booleana>; increment comptador) {  
    // Instruccions  
}
```

Nota: n'hi ha més però per ara anirem fent amb aquestes.

Ex. 1. Digues que fa el següent codi font:

```
public class F06_Ex01_sorpresa {
    public static final int VALOR = 25;
    public static void main (String[] args) {
        int n=0;
        for (int i=1;i<VALOR;i++) {
            if (i%10==0) {
                n++;
            }
        }
        System.out.println("El resultat és "+n+".");
    }
}
```

Ex. 2. Fes un programa anomenat MostraInterval que:

- llegeixi per teclat dos valors enters, n1 i n2
- comprovi que  $n1 < n2$  i mostri un error si no és així
- escrigui, si no hi ha hagut error, tots els nombres enters dins l'interval  $[n1, n2]$

Ex. 3. Fes un programa anomenat SumaInterval que:

- llegeixi per teclat dos valors enters, n1 i n2
- comprovi que  $n1 < n2$  i mostri un error si no és així (decideix si vols que acabi o no)
- escrigui, si no hi ha hagut error, la suma de tots els enters dins l'interval  $[n1, n2]$

Ex. 4. Fes un programa anomenat LlistaMultiples que:

- llegeixi per teclat dos valors enters, n1 i n2
- comprovi que  $n1 < n2$  i mostri un error si no és així
- escrigui, començant a n1, tots els nombres enters que són múltiples de n1 més petits o iguals que n2.

Ex. 5. Fes un programa anomenat CercarDivisor que llegeixi des del teclat un nombre enter positiu major que 1 i escrigui el divisor positiu més gran diferent d'ell mateix i d'1. En cas de no trobar-ne ho ha d'indicar.

Utilitza una variable tipus semàfor.

Ex. 6. Fes un programa anomenat ComptaXifres que llegeixi un nombre enter des del teclat i escrigui el nombre de xifres que té. Tingues en compte que el valor 0 té una xifra.

Ex. 7. Fes un programa anomenat SumarFinsLimit que llegeixi des del teclat un valor enter i mostri fins a quin punt de la seqüència ordenada de positius (1, 2, 3, ...) s'ha de sumar per arribar al nombre més proper per davall d'aquest valor. També ha de mostrar la suma d'aquests nombres.

Exemple: 18 → 1, 2, 3, 4, 5 (15)

Ex. 8. Fes un programa anomenat ParellsDinsInterval que:

- llegeixi per teclat dos valors enters, n1 i n2
- comprovi que  $n1 < n2$  i mostri un error si no és així
- escrigui els nombres parells que hi ha dins l'interval en ordre decreixent.
- No pots utilitzar la sentència "if"

Ex. 9. Fes un programa que conti el nombre de múltiples de 7 que hi ha entre 1 i 1000.

Ex. 10. Fes un programa que sumi els 100 primers nombres enters positius.

Ex. 11. Fes un programa anomenat DibuixaTriangle que llegeixi un valor enter des del teclat i mostri per pantalla un triangle rectangle fet de lletres "O", amb altura igual al valor llegit.

Exemple: 5

```
O
OO
OOO
OOOO
OOOOO
```

Ex. 12. Fes un programa anomenat PiramideEstrelles que llegeixi des del teclat un valor de tipus enter. S'ha de dibuixar una piràmide amb asteriscs ('\*') d'alçada igual al valor llegit.

Exemple: 4

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

Indicacions:

- El nombre d'espais abans de l'asterisc comença per l'alçada-1 i va decreixent.
- A les files, es comença per 1 asterisc i va creixent de dos en dos.

Ex. 13. Fes un programa que llegeixi per teclat un nombre enter múltiple de 10. No s'acceptarà fins que sigui múltiple de 10, si no ho és el tornarem a demanar. Un cop el tinguem, escriurà una sortida com la de l'exemple:

```
1200 → 12 per 10 elevat a 2
58000000 → 58 per 10 elevat a 6
```

Ex. 14. Fes un programa que calculi el creixement d'un bacteri dins un ésser viu. Els biòlegs han calculat que a partir de 10.000.000, l'organisme hoste no pot sobreviure.

El bacteri es reproduïx per clonació, cada 3 minuts és capaç de generar una còpia d'ella mateixa. Cada còpia té les mateixes capacitats de rèplica que el bacteri pare.

El programa ha de mostrar per pantalla quantes hores triga el bacteri a assolir la xifra mortal.

Ex. 15. Fes un programa que demani a l'usuari quants nombres vol introduir. A continuació demani cadascun d'aquests nombres i digui quin és el menor, el major i la mitjana de tots ells.

Ex. 16. Escriviu un programa que demani l'amplada i l'alçada d'un rectangle i el dibuixi amb asteriscs. Exemple:

```
Amplada: 5
Alçada: 4
*****
* *
* *
*****
```

Ex. 17. Escriviu un programa que demani un nombre i escrigui els seus divisors.

Exemple:

```
Número: 48
Els divisors de 48 són: 1 2 3 4 6 8 12 16 24 48
```

Ex. 18. Escriu un programa que demani quants nombres vols introduir. Demani aquests nombres i retorni quant nombres negatius s'han introduït.

Ex. 19. Escriu un programa que mostri la següent sortida per pantalla:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	13	18	20
20	22	24	26	28	30	32	34	36	38
10	14	18	22	26	30				
40	35	30	25	20	15	10	5	0	

Indicació: Cada línia correspon a un bucle diferent.

Ex. 20. Escriu un programa que mostri la següent sortida per pantalla:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	...				
					...	97	98	99	100