



Unió Europea

Fons Social Europeu

L'FSE inverteix en el teu futur



GENERALITAT
VALENCIANA



UF05. - BUCLES EN JAVA

- Teoria -

PROGRAMACIÓ
CFGs DAM

Autor:

José Manuel Martí Fenollosa

Revisat per:

Àngel Olmos Giner

a.olmosginer@edu.gva.es

2022/2023

BUCLES EN JAVA

ÍNDEX DE CONTINGUTS

1. Introducció
2. Bucle FOR
3. Bucle WHILE
4. Bucle DO-WHILE
5. Exemples

1. INTRODUCCIÓ

INTRODUCCIÓ

Els **bucles** són estructures de repetició



Blocs d'instruccions que es repeteixen un nombre de vegades **mentre o fins que** es compleixi una condició

El **bloc d'instruccions** es trobarà **tancat mitjançant claus** {.....}

Existeixen tres construccions per a aquestes estructures de repetició:

- Bucle ***for***
- Bucle ***while***
- Bucle ***do-while***

Tot problema que requereixi repetició pot fer-se amb qualsevol dels tres, però segons el cas sol ser més senzill o intuïtiu utilitzar l'un o l'altre

1. INTRODUCCIÓ

INTRODUCCIÓ

Com a regla general, **quan utilitzarem un FOR, un WHILE o un DO-WHILE?**

1. INTRODUCCIÓ

INTRODUCCIÓ

Com a regla general **és recomanable**:

- Utilitzar el bucle **FOR** quan **sí** es conega per endavant el nombre exacte de vegades que ha de repetir-se el bloc d'instruccions.
- Utilitzar el bucle **WHILE** quan **no** sabem el nombre de vegades que ha de repetir-se el bloc i és possible que no haja d'executar-se cap vegada.
- Utilitzar el bucle **DO-WHILE** quan **no** sabem el nombre de vegades que ha de repetir-se el bloc i deurà executar-se almenys una vegada.

Amb major o menor esforç, pot utilitzar-se qualsevol de les tres indistintament

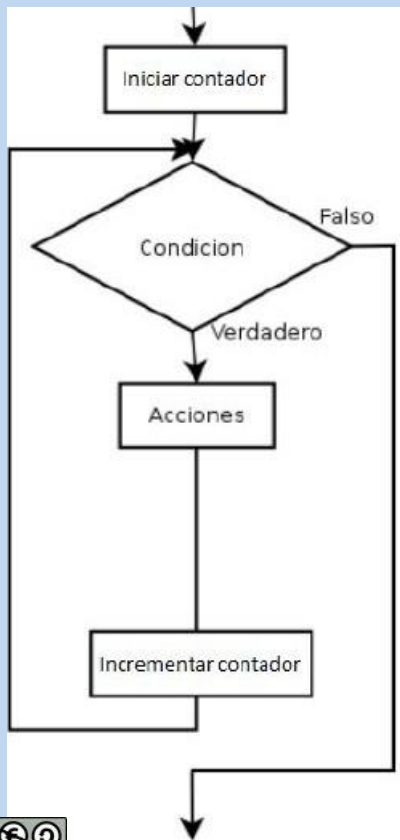
BUCLES EN JAVA

ÍNDEX DE CONTINGUTS

1. Introducció
2. **Bucle FOR**
3. Bucle WHILE
4. Bucle DO-WHILE
5. Exemples

2. BUCLE FOR

DEFINICIÓ



S'executa una sola vegada a l'inici del bucle, normalment per a inicialitzar un comptador.

Per exemple `int i = 1;`

S'executa al FINAL de cada iteració del bucle (després del bloc d'instruccions). Generalment s'utilitza per a incrementar o decrementar el comptador.

Per exemple `i++`; (incrementar i en 1).

```
for (inicialització ; condició ; increment)
{
    bloc d'accions;
}
```

Expressió lògica que s'avalua a l'inici de cada iteració del bucle. En el moment en què aquesta expressió s'avalua a *false* es deixarà d'executar el bucle i el control del programa passarà a la següent instrucció (a continuació del bucle *for*). S'utilitza per a indicar la condició en la qual vols que el bucle continue. Per exemple `i <= 10`;

2. BUCLE FOR

EXEMPLES

Exemple 1: Bucle que mostra per pantalla els nombres naturals de l'1 al 10:

```
for (int i = 1; i <= 10 ; i++)  
{ System.out.println(i) ;  
}
```

- En la inicialització utilitzem **int i=1** per a crear la variable i amb un valor inicial de 1
- La condició **i<=10** indica que el bucle ha de repetir-se mentre i siga menor o igual a 10
- L'actualització **i++** indica que, al final de cada iteració, i ha d'incrementar-se en 1

2. BUCLE FOR

EXEMPLES

Exemple 2: Programa que mostra els nombres naturals (1,2,3,4,5,6,...) fins a un número introduït per teclat:

```
6 public static void main(String[] args) {  
7     Scanner sc = new Scanner(System.in);  
8     int max;  
9     System.out.print("Introduce el número máximo: ");  
10    max = sc.nextInt();  
11    for (int i = 1; i <= max; i++) {  
12        System.out.println("Número: " + i);  
13    }  
14 }  
15 }  
16 }
```

Sent l'eixida??

2. BUCLE FOR

EXEMPLES

Exemple 2: Programa que mostra els nombres naturals (1,2,3,4,5,6,...) fins a un número introduït per teclat:

```
6 public static void main(String[] args) {  
7     Scanner sc = new Scanner(System.in);  
8     int max;  
9     System.out.print("Introduce el número máximo: ");  
10    max = sc.nextInt();  
11    for (int i = 1; i <= max; i++) {  
12        System.out.println("Número: " + i);  
13    }  
14 }  
15 }  
16 }
```

Sent l'eixida:

```
run:  
Introduce el número máximo: 5  
Número: 1  
Número: 2  
Número: 3  
Número: 4  
Número: 5  
BUILD SUCCESSFUL (total time: 6 seconds)
```

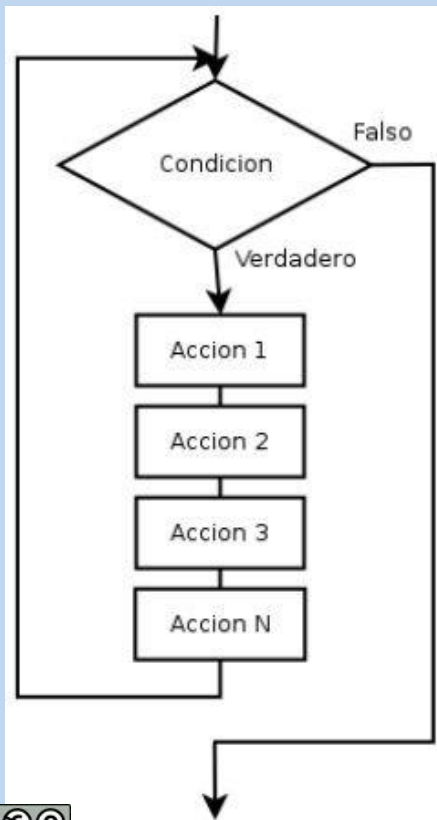
BUCLES EN JAVA

ÍNDEX DE CONTINGUTS

1. Introducció
2. Bucle FOR
3. Bucle WHILE
4. Bucle DO-WHILE
5. Exemples

3. BUCLE WHILE

DEFINICIÓ



El bloc d'instruccions s'executa mentre es compleix una condició (mentre **condició** == true)

La **condició** es comprova **ABANS de començar** a executar per primera vegada **el bucle**

*Per exemple: si a la primera iteració **condicio** == **false** → el bloc d'accions no s'executarà cap vegada*

```
while (condició) {  
    bloc d'accions;  
}
```

3. BUCLE WHILE

EXEMPLES

Exemple 3: El mateix Exemple 2 d'abans però fet amb un bucle WHILE:

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int max, cont;
    System.out.print("Introduce el número máximo: ");
    max = sc.nextInt();
    cont = 1;
    while (cont <= max) {
        System.out.println("Número: " + cont);
        cont++;
    }
}
```

Sent l'eixida:

```
run:
Introduce el número máximo: 5
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
BUILD SUCCESSFUL (total time: 6 seconds)
```

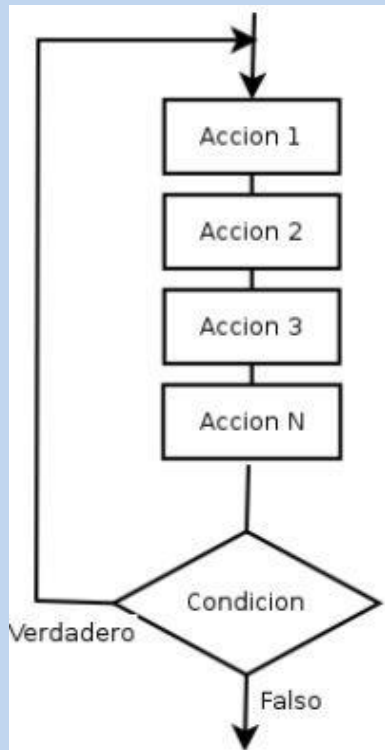
BUCLES EN JAVA

ÍNDEX DE CONTINGUTS

1. Introducció
2. Bucle FOR
3. Bucle WHILE
4. **Bucle DO-WHILE**
5. Exemples

4. BUCLE DO-WHILE

DEFINICIÓ



El bloc d'instruccions s'executa sempre almenys una vegada, i aqueix bloc d'instruccions s'executarà mentre **condició == true**

En el bloc d'instruccions **haurà d'existir alguna iteració en que 'condició' s'avalua a 'false'**. Si no el bucle no acabaria mai

```
do  
{  
    bloc d'accions;  
}  
while (condició);
```

4. BUCLE DO-WHILE

EXEMPLES

Exemple 4: El mateix Exemple 2 d'abans però fet amb un bucle DO-WHILE:

```
public static void main(String[] args) {  
7     Scanner sc = new Scanner(System.in);  
8     int max, cont;  
9     System.out.print("Introduce el número máximo: ");  
10    max = sc.nextInt();  
11    cont = 1;  
12  
13    do {  
14        System.out.println("Número: " + cont);  
15        cont++;  
16    } while (cont <= max);  
17  
18 }  
19 }
```

run:

Introduce el número máximo: 5

Número: 1

Número: 2

Número: 3

Número: 4

Número: 5

BUILD SUCCESSFUL (total time: 6 seconds)

Sent l'eixida:

BUCLES EN JAVA

ÍNDEX DE CONTINGUTS

1. Introducció
2. Bucle FOR
3. Bucle WHILE
4. Bucle DO-WHILE
5. Exemples

5. EXEMPLES

Exemple 1: Programa que mostre per pantalla els 20 primers nombres naturals (1, 2, 3... 20).

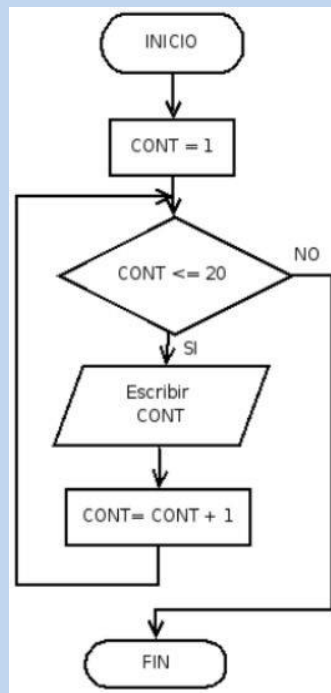
Resoldre utilitzant un *while*

DIY

5. EXEMPLES

Exemple 1: Programa que mostre per pantalla els 20 primers nombres naturals (1, 2, 3... 20).

Resoldre utilitzant un *while*



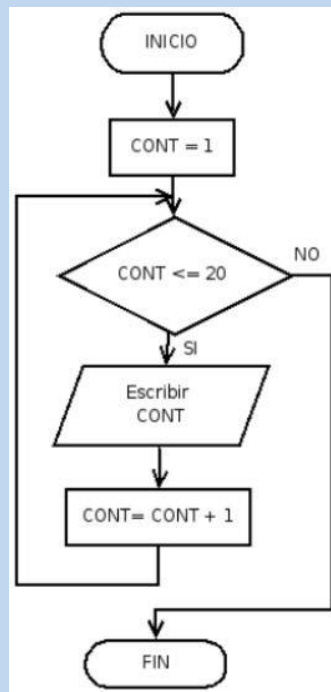
```
6 package exempleuf05;
7
8 /**
9  *
10  * @author aolmosginer
11  */
12 public class ExempleUF051 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int cont = 1;
20         while (cont<=20) {
21             System.out.print(cont + " ");
22             cont ++;
23         }
24         System.out.println("\n");
25     }
26
27 }
```

Sent l'eixida??

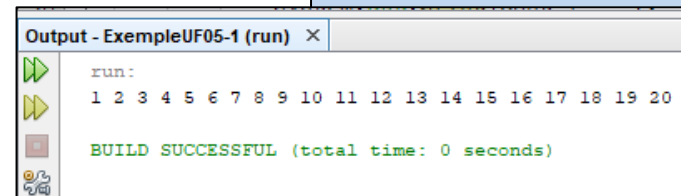
5. EXAMPLES

Exemple 1: Programa que mostre per pantalla els 20 primers nombres naturals (1, 2, 3... 20).

Resoldre utilitzant un *while*



```
6 package exempleuf05;
7
8 /**
9  *
10  * @author aolmosginer
11  */
12 public class ExempleUF051 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int cont = 1;
20         while (cont <= 20) {
21             System.out.print(cont + " ");
22             cont ++;
23         }
24         System.out.println("\n");
25     }
26
27 }
```



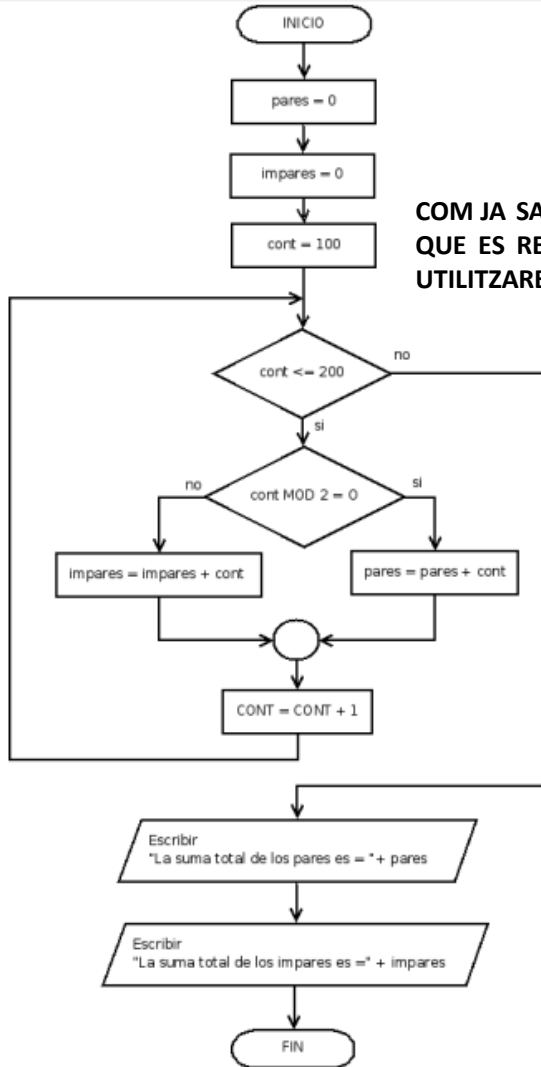
5. EXEMPLES

Exemple 2: Programa que suma independentment els parells i els imparells dels números compresos entre 100 i 200.

DIY

5. EXEMPLES

Exemple 2: Programa que suma independentment els parells i els imparells dels números compresos entre 100 i 200.



COM JA SABEM LES VEGADES QUE ES REPETEIX EL BUCLE, UTILITZAREM UN **FOR**

5. EXAMPLES

Exemple 2: Programa que suma independentment els parells i els imparells dels números compresos entre 100 i 200.

```
12 public class Ejercicio11 {
13
14     public static void main(String[] args) {
15         int pares, impares, cont;
16
17         pares = 0;
18         impares = 0;
19
20         for(cont=100; cont <= 200; cont++)
21         {
22             if(cont % 2 == 0)
23                 pares = pares + cont;
24             else
25                 impares = impares + cont;
26         }
27
28         System.out.println("La suma total de los pares es " + pares);
29         System.out.println("La suma total de los impares es " + impares);
30     }
31
32 }
```



run:

La suma total de los pares es 7650

La suma total de los impares es 7500

BUILD SUCCESSFUL (total time: 0 seconds)

EXERCICIS PROPOSATS

Autor:

José Manuel Martí Fenollosa

Revisat per:

Àngel Olmos Giner

Llicència:



CC BY-NC-SA 3.0 ES Reconeixement – No Comercial – Compartir Igual (by-nc-sa)

No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original. Aquesta és una obra derivada de l'obra original de José Manuel Martí Fenollosa