

ArrayList– For Each - Iterator

1.- DECLARACIÓ I CREACIÓ D'UN ARRAYLIST.

ArrayList és una classe contenidora genèrica, que implenta arrays dinàmics d'objectes de qualsevol tipus.

```
import java.util.ArrayList;
```

De forma general un **ArrayList** en Java es crea de la següent manera:

```
ArrayList nomArray = new ArrayList();
```

Esta instrucció crea l'ArrayList **nomArray** buit.

Un **ArrayList** declarat així, pot contindre objectes de qualsevol tipus.

Per exemple:

```
ArrayList a = new ArrayList();  
a.add("Bon dia xiquets");  
a.add(3);  
a.add('a');  
a.add(23.5);
```

Els elements de l'**ArrayList** a són:

```
"Bon dia xiquets" 3 'a' 23.5
```

Es a dir, **un ArrayList pot contindre objectes de tipus distints.**

En aquest exemple, el primer objecte que s'afegí és l'String **"Bon dia xiquets"**. La resta **no són objectes**. Són dades de tipus bàsics però el compilador els **converteix automàticament en objectes** de la seua classe contenidora abans d'afegir-los l'array.

Un array al qual se li poden assignar elements diferents pot tindre alguna complicació a l'hora de treballar amb ell. Per això, una alternativa a aquesta declaració és indicar el tipus d'objectes que conté.

En aquest cas, l'array només podrà contindre objectes d'este tipus.

De manera general:

```
ArrayList <tipus> nomArray = new ArrayList();
```

on **tipus deu ser una classe** i indica el tipus d'objectes que contindrà l'array.

No es poden usar tipus primitius. Per a un tipus primitiu s'ha d'utilitzar la seua classe contenidora.

Per exemple:

```
ArrayList <Integer> numerosEnters = new ArrayList();
```

```
ArrayList <Double> numerosReals = new ArrayList();
```

Crea un array de numeros d'enters, i un array de numeros reals respectivament.

2.- MÈTODES DE LA CLASSE ARRAYLIST

Alguns dels mètodes que proporciona ArrayList són:

| MÈTODE | DESCRIPCIÓ |
|------------------------|--|
| size() | Retorna el nombre d'elements (int) |
| add(X) | Afig l'objecte X al final. Retorna true. |
| add(posició, X) | Insereix l'objecte X a la posició indicada. |
| get(posició) | Retorna l'element que està en la posició indicada. |
| remove(posició) | Elimina l'element que es troba en la posició indicada. Retorna l'element eliminat. |
| remove(X) | Elimina la primera ocurrència de l'objecte X. Retorna true si l'element està en la llista, false en cas contrari |
| clear() | Elimina tots els elements. |
| set(posició, X) | Substitueix l'element que es troba en la posició indicada per l'objecte X. Retorna l'element substituït. |
| contains(X) | Comprova si la col·lecció conté a l'objecte X. Retorna true o false. |
| indexOf(X) | Retorna la posició de l'objecte X. Si no existeix retorna -1 |

Els pots consultar tots a l'API de Java

<https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/ArrayList.html>

3.- FORMES DE RECÓRRER UN ARRAYLIST

- **Amb un bucle for:**

```
for(int i = 0; i < array.size(); i++){  
    System.out.println(array.get(i));  
}
```

- **Amb un bucle bucle foreach:**

Si suposem l'array d'enters anomenat números:

```
for (Integer i: numeros) {  
    System.out.println(i);  
}
```

Si l'array conté objectes de tipus diferents o desconeixem el tipus:

```
for(Object o: nomArray){  
    System.out.println(o);  
}
```

- **Utilitzant un objecte Iterator.**

L'avantatge d'usar un Iterator és que no necessitem indicar el tipus d'objecte que conté l'array.

S'ha d'importar la classe Iterator.

```
import java.util.Iterator;
```

Iterator té dos mètodes

- **hasNext**: retorna true si hi ha més elements a l'array
- **next**: retorna el següent objecte de l'array.

Per exemple:

```
ArrayList<Integer> numeros = new ArrayList();
.....
// inserim elements
.....

// es crea l'ITERATOR it per a recorre l'array numeros

Iterator it = numeros.iterator();

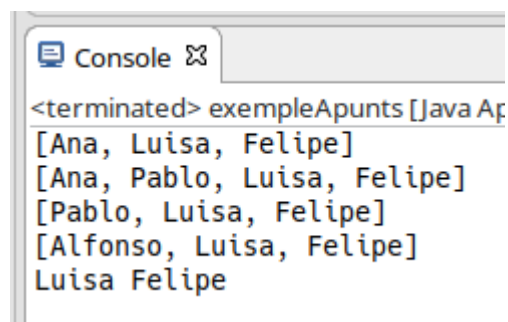
while(it.hasNext())                // mentres queden elements
    System.out.println(it.next()); // s'obtenen i es mostren

}
```

4.- EXAMPLES ARRAYLIST

Exemple 1:

```
public static void main(String[] args) {  
  
    ArrayList<String> noms = new ArrayList();  
  
    noms.add("Ana");  
    noms.add("Luisa");  
    noms.add("Felipe");  
  
    System.out.println(noms); // [Ana, Luisa, Felipe]  
  
    noms.add(1, "Pablo");  
    System.out.println(noms); // [Ana, Pablo, Luisa, Felipe]  
  
    noms.remove(0);  
    System.out.println(noms); // [Pablo, Luisa, Felipe]  
  
    noms.set(0, "Alfonso");  
    System.out.println(noms); // [Alfonso, Luisa, Felipe]  
  
    String s = noms.get(1);  
    String ultim = noms.get(noms.size() - 1);  
    System.out.println(s + " " + ultim); // Luisa Felipe  
  
}
```



Exemple 2: Escriu un programa que demane per teclat números enters i els guardi en un ArrayList fins que introduïm un 0, a continuació mostra els números introduïts, la seua suma i la seua mitjana

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

public class ExempleArrayList {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        ArrayList<Integer> numeros = new ArrayList();
        int n;

        do {
            System.out.println("Introdueix números enters. 0 per finalitzar: ");
            System.out.println("Numero: ");
            n = sc.nextInt();

            if (n != 0){
                numeros.add(n);
            }

        }while (n != 0);

        System.out.println("Has introduït: " + numeros.size() + " números:");

        //mostra l'ArrayList complet.
        System.out.println(numeros);

        // recorrem utilitzant un Iterator per mostrar un element per línia.

        Iterator it = numeros.iterator();
        while(it.hasNext()){
            System.out.println(it.next());
        }

        // recorrem utilitzant foreach per sumar els elements de l'ArrayList

        double suma = 0;
        for(Integer i: numeros){
            suma = suma + i;
        }
        System.out.println("Suma: " + suma);
        System.out.println("Mitjana: " + suma/numeros.size());
    }
}
```

Console

<terminated> ExempleArrayList [Java Application] /usr/lib/jvm/ja

Introdueix números enters. 0 par finalitzar:

Numero:

3

Introdueix números enters. 0 par finalitzar:

Numero:

4

Introdueix números enters. 0 par finalitzar:

Numero:

6

Introdueix números enters. 0 par finalitzar:

Numero:

8

Introdueix números enters. 0 par finalitzar:

Numero:

0

Has introduït: 4 números:

[3, 4, 6, 8]

3

4

6

8

Suma: 21.0

Mitjana: 5.25

5.- PROPIETATS ARRAYLIST

El nom d'un ArrayList conté la referència a l'ArrayList, és a dir, la direcció de memòria on es troba, al igual que succeeix amb els arrays estàtics.

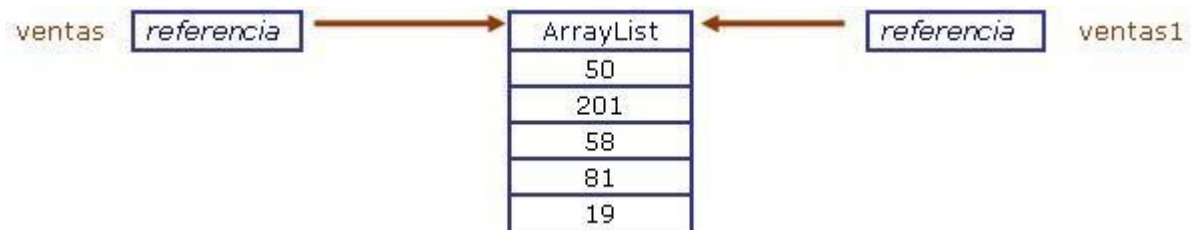
Si disposem d'un ArrayList d'enters anomenat ventas:



La instrucció:

```
ArrayList<Integer> ventas1 = ventas;
```

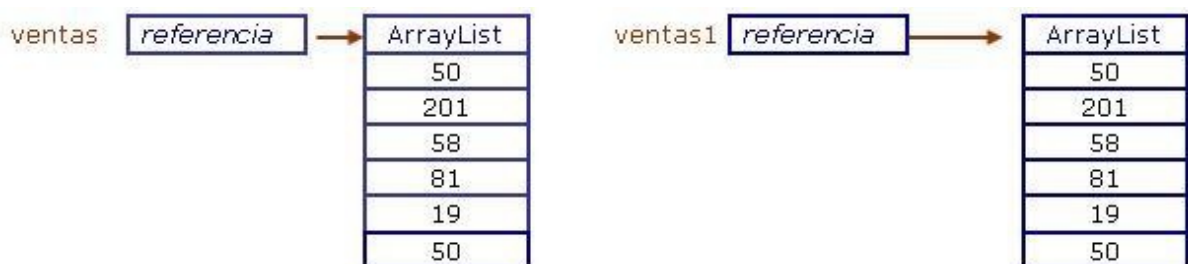
No copia l'array ventas en el nou array ventas1 sino que **crea un alias**:



D'aquesta manera tenim dues maneres d'accedir al mateix ArrayList: mitjançant la referència `ventas` i mitjançant la referència `ventas1`.

Per a fer una còpia podem fer-ho de manera manual element a element o es pot passar la referència del ArrayList original al constructor del nou:

```
ArrayList <Integer> ventas1 = new ArrayList(ventas);
```



- **ARRAYLIST COM A PARÀMETRE D'UN MÈTODE**

Un ArrayList pot ser usat com a paràmetre d'un mètode. A més un mètode pot retornar un ArrayList mitjançant la sentència return.

Exemple: Mètode que rep un **ArrayList** de **String** i ho modifica invertint el seu contingut:

```
import java.util.ArrayList;

public class ExempleArrayList2 {

    public static void main(String[] args) {

        ArrayList<String> noms = new ArrayList();
        noms.add("Ana");
        noms.add("Luisa");
        noms.add("Felipe");
        noms.add("Pablo");

        System.out.println(noms);

        noms = invertir(noms);
        System.out.println(noms);
    }

    public static ArrayList<String> invertir(ArrayList<String> noms) {
        // Crea una llista auxiliar de nom resultat que retornarà el mètode
        // invertir

        ArrayList<String> resultat = new ArrayList();
        // Recorre la lista de noms en ordre invers

        for (int i = noms.size() - 1; i >= 0; i--) {

            // Afegeix cada nom al resultat
            resultat.add(noms.get(i));
        }

        return resultat;
    }
}
```

