

## 9.C. Conjuntos.

### 2. Conjuntos (II).

Y ahora te preguntarás, ¿cómo **accedo a los elementos almacenados** en un conjunto? Para obtener los elementos almacenados en un conjunto hay que **usar iteradores**, que permiten obtener **los elementos del conjunto uno a uno de forma secuencial** (no hay otra forma de acceder a los elementos de un conjunto, es su inconveniente). Los iteradores se ven en mayor profundidad más adelante, de momento, vamos a usar iteradores de forma transparente, a través de una **estructura for especial**, denominada bucle **"for-each"** o bucle "para cada". En el siguiente código se usa un bucle for-each, en él la variable **i** va tomando todos los valores almacenados en el conjunto hasta que llega al último:

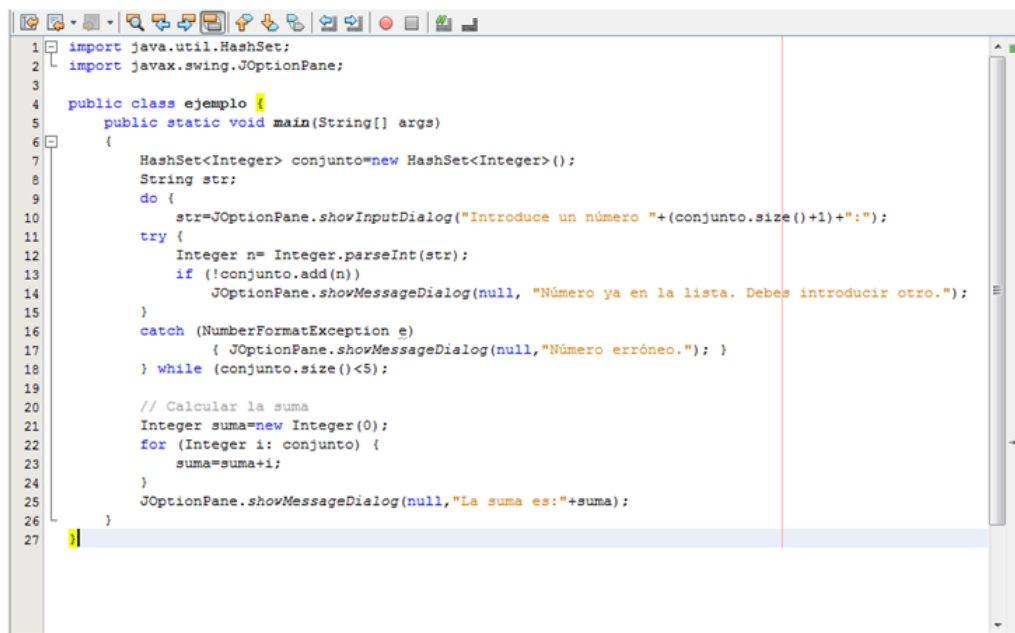
```
for (Integer i: conjunto) {  
  
    System.out.println("Elemento almacenado:"+i);  
  
}
```

Como ves la estructura for-each es muy sencilla: la palabra **for** seguida de "(tipo variable:colección)" y el cuerpo del bucle; tipo es el tipo del objeto sobre el que se ha creado la colección, variable pues es la variable donde se almacenará cada elemento de la colección y colección pues la colección en sí. Los bucles for-each se pueden usar para todas las colecciones.

#### Ejercicio resuelto

Realiza un pequeño programita que pregunte al usuario 5 números diferentes (almacenándolos en un **HashSet**), y que después calcule la suma de los mismos (usando un bucle for-each).

Sol.: Una solución posible podría ser la siguiente. Para preguntar al usuario un número y para mostrarle la información se ha usado la clase **JOptionPane**, pero podrías haber utilizado cualquier otro sistema. Fíjate en la solución y verás que el uso de conjuntos ha simplificado enormemente el ejercicio, permitiendo al programador o la programadora centrarse en otros aspectos:



```
1 import java.util.HashSet;  
2 import javax.swing.JOptionPane;  
3  
4 public class ejemplo {  
5     public static void main(String[] args)  
6     {  
7         HashSet<Integer> conjunto=new HashSet<Integer>();  
8         String str;  
9         do {  
10            str=JOptionPane.showInputDialog("Introduce un número "+(conjunto.size()+1)+":");  
11            try {  
12                Integer n= Integer.parseInt(str);  
13                if (!conjunto.add(n))  
14                    JOptionPane.showMessageDialog(null, "Número ya en la lista. Debes introducir otro.");  
15            }  
16            catch (NumberFormatException e)  
17            { JOptionPane.showMessageDialog(null,"Número erróneo."); }  
18            while (conjunto.size()<5);  
19  
20            // Calcular la suma  
21            Integer suma=new Integer(0);  
22            for (Integer i: conjunto) {  
23                suma=suma+i;  
24            }  
25            JOptionPane.showMessageDialog(null,"La suma es:"+suma);  
26        }  
27    }
```

```
import java.util.HashSet;
```

```
import javax.swing.JOptionPane;
```

```
public class ejemplo {
```

```
    public static void main(String[] args)
```

```
{
```

```

HashSet<Integer> conjunto=new HashSet<Integer>();

String str;

do {

    str=JOptionPane.showInputDialog("Introduce un número "+(conjunto.size()+1)+"-:");

    try {

        Integer n= Integer.parseInt(str);

        if (!conjunto.add(n))

            JOptionPane.showMessageDialog(null, "Número ya en la lista. Debes introducir otro.");

    }

    catch (NumberFormatException e)

        { JOptionPane.showMessageDialog(null,"Número erróneo."); }

} while (conjunto.size()<5);


// Calcular la suma

Integer suma=new Integer(0);

for (Integer i: conjunto) {

    suma=suma+i;

}

JOptionPane.showMessageDialog(null,"La suma es:"+suma);

}

}

```