



3. ArrayList



Date

Empty



Status

Not started



Type

Empty

Units

4. Estructuras de almacenamiento

Colecciones

Las colecciones son contenedores de datos que, a diferencia de los arrays, tienen la capacidad de aumentar y disminuir su tamaño de forma dinámica según las necesidades del programa. Existen multitud de colecciones en el paquete `utils`, pero las principales son que trabajaremos son:

- ArrayList
- HashSet
- Hashtable
- HashMap
- Stack

ArrayList

Lista que utiliza un array de tamaño modificable. Su manejo es

bastante simple, teniendo como inconveniente el tiempo que se emplea en añadir o borrar elementos. Otro de sus elementos es que no es sincronizada, por lo que no puede ser utilizada por varios hilos (en lectura y escritura) al mismo tiempo.

Para crear un arraylist:

```
ArrayList lista = new ArrayList(); ArrayList<String> listaString = new ArrayList<String>();
```

Los dos constructores crean lo mismo, con la diferencia que el segundo "fuerza" a que todos los objetos de la lista sean del tipo indicado, en este caso String

Los principales métodos de un ArrayList son:

```
// Añadir un elemento n a la lista por el final: lista.add(n); // Añadir un elemento n a la lista en una posición a: lista.add(a,n); // Vaciar la lista : lista.clear(); // Clonar la lista : lista.clone(); // Comprobar si la lista contiene un elemento n: lista.contains(n) → boolean // Encontrar la posición de la primera aparición del elemento n en la lista o -1 si no existe: lista.indexOf(n) → int // Encontrar la posición de la última aparición del elemento n en la lista o -1 si no existe: lista.lastIndexOf(n) → int // Obtener el número de elementos de la lista lista.size() // Obtener el dato almacenado en la posición "i" de la lista : n = lista.get(i); // Eliminar el valor entero de la lista que ocupa la posición "n" (se devuelve el entero "a" eliminado): a = lista.remove(n); // Eliminar la primera aparición del valor entero "n" en la lista : lista.remove(n); // Modificar el valor almacenado en la lista en la posición "a" por el valor entero "n": lista.set(a, n);
```

Recorrer elementos de un arraylist

Con los métodos vistos anteriormente se puede por ejemplo recorrer todos los elementos que forman parte de una lista

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

O bien con un foreach

```
for (String elemento: listaString) { System.out.println(eleme  
nto); }
```