

Contenidos

☐ ArrayList

Collections

- Colecciones son **objetos** que referencian a un grupo de objetos.
- **A diferencia de los arreglos**, las colecciones solamente mantienen referencias a objetos de tipo **Object**.
- Cualquier objeto puede ser almacenado en una colección.
- Las colecciones pueden:
 - Cambiar de tamaño dinámicamente.
 - Proveer ordenamiento.
 - insertar y borrar elementos.

ArrayList

- *Similar a un arreglo*
- *Provee metodos para manejar los datos.*
- *Su tamaño crece dinamicamente*
- *Mas lento en ejecucion que un arreglo*

Constructor

ArrayList a = new ArrayList();

ArrayList a = new ArrayList(int capacidadInicial);

- *Si sabemos de que tipo es el objeto que vamos a almacenar podemos especificarlo al crear el ArrayList.*
- *Solo podemos especificar clases del tipo Object*

ArrayList <String> a = new ArrayList<String>();

Import

Para utilizar ArrayList dentro de nuestros programas debemos importar la clase ArrayList.

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

Agregar Objetos

- Add(Tipo elemento)

Agrega el elemento al final de la lista

- Add(int índice, Tipo elemento)

Agrega el elemento en la posición especificada en la lista.

Ejemplo:

```
ArrayList <String> a = new ArrayList<String>();  
a.add("blabla");  
a.add("ah?");
```

Modificar Objetos

- Set(int índice, Tipo elemento)

Reemplaza el elemento en el índice especificado

Ejemplo:

```
ArrayList <String> a = new ArrayList<String>();  
a.add("blabla");  
a.add("ah?");  
a.set(1, "eh?");
```

Eliminar Objetos

- `remove(int indice)`

Reemplaza el elemento en el índice especificado

- `remove (Tipo elemento)`

Elimina la primera coincidencia con el objeto entregado por parametro.

- `removeRange(int indiceInicio, indiceFin)`

Elimina los objetos entre los indices, desde `indiceInicio`(incluido) y `indiceFin`(excluido).

- `clear()`

Elimina todos los elementos.

Eliminar Objetos

Ejemplo:

```
ArrayList <String> a = new ArrayList<String>();  
a.add("blabla");  
a.add("ah?");  
a.remove(1);  
a.remove ("ah?");
```

Recorrer ArrayList(1)

- `get(int índice)`

Retorna el elemento que se encuentra en el índice especificado

- `size()`

Retorna la cantidad de elementos en el ArrayList.

Ejemplo

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

Recorrer ArrayList(2)

- Otra forma de recorrer el array list es:

```
for (Iterator it=a.iterator(); it.hasNext( ); )  
{  
    Object obj = it.next();  
    System.out.println(obj);  
}
```

Nota: Si deseamos utilizar Iterator en ves de importar **java.util.ArrayList** debemos importar **java.util.***;

Recorrer ArrayList(3)

- Otra forma de recorrer el array list es utilizando un ciclo for each.
- Al crear el ArrayList debemos especificar el tipo.

```
ArrayList <String> a = new ArrayList<String>();  
a.add("blabla");  
a.add("ah?");  
for(String elemento : a)  
{  
    System.out.println(elemento);  
}
```

Metodos ArrayList

- La clase ArrayList implementa varios metodos mas que eventualmente podrían ser utiles.
- La API provee la documentación de la clase.

URL: