Programació UT8.1. ArrayList

Introducció

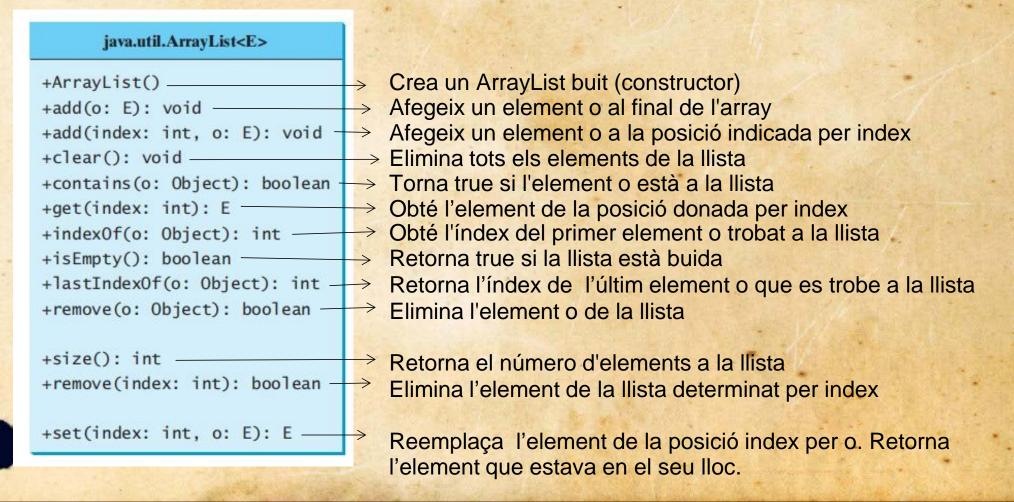
- Fins ara, sempre que hem volgut emmagatzemar una llista de dades/objectes del mateix tipus/classe hem fet ús d'un array.
 - int[] números = new int[3]
 - Interval[] intervals = new Interval[10]
- El problema d'esta estructura, és que una vegada s'ha creat, el seu tamany es fix.

ArrayList

- En l'API de Java, disposem de la classe ArrayList que ens serà de gran utilitat.
- Pot ser utilitzada per emmagatzemar un número il·limitat d'objectes de la mateixa classe.

Interfície d'ArrayList

Estos són alguns de els mètodes que ofereix:

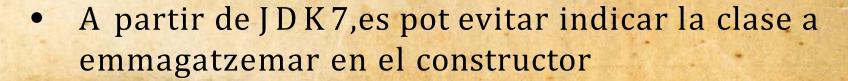


Classe genèrica. Constructor

 ArrayList és coneguda com una classe genérica amb un tipus genèric E. Eixe tipus genèric será reemplaçat per un tipus concret quan es creu un ArrayList.

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

 En este exemple, es crea una llista "cities", la qual emmagatzemarà objectes de la classe String



Exemple

```
public class TestArrayList {
  public static void main(String[] args) {
    // Create a list to store cities
    ArrayList<String> cityList = new ArrayList<>();
   cityList.add("London");
   cityList.add("Denver");
   cityList.add("Paris");
   cityList.add("Miami");
    System.out.println("List size? " + cityList.size());
    System.out.println("Is Miami in the list? " +
      cityList.contains("Miami"));
    System.out.println("The location of Denver in the list? "
      + cityList.indexOf("Denver"));
    System.out.println("Is the list empty? " +
      cityList.isEmpty()); // Print false
```

Comparativa Array vs ArrayList

Operation	Array	ArrayList
Creating an array/ArrayList	String[] a = new String[10]	<pre>ArrayList<string> list = new ArrayList<>();</string></pre>
Accessing an element	a[index]	<pre>list.get(index);</pre>
Updating an element	a[index] = "London";	<pre>list.set(index, "London");</pre>
Returning size	a.length	list.size();
Adding a new element		list.add("London");
Inserting a new element		<pre>list.add(index, "London");</pre>
Removing an element		<pre>list.remove(index);</pre>
Removing an element		list.remove(Object);
Removing all elements		list.clear();

I per a crear arrays de tipus de dades bàsics amb ArrayList?

- Hem vist que podem crear llistes amb objectes, però què passa amb els tipus de dades primitius?
- Farem ús de les classes envoltori: Integer, Double, Boolean, Character, etc....

ArrayList<Integer> list = new ArrayList<>();

Com es fan las següents operacions?

- Crear un array que emmagatzeme números (double)
- Afegir un objecte String a una lista
- Inserir un objete interval al principi de la lista
- Saber el número d'objectes d'una lista d'enters
- Eliminar el número 4 de l'array del primer punt
- Eliminar l'últim objecte del primer punt
- Comprovar que un interval donat està en la lista
- Obtindre el segon element de la llista del punt 4
 - Comprovar que un element es troba de forma única a la primera llista.

Quins errors hi ha en el següent codi?

```
ArrayList<String> list = new ArrayList<>();
list.add("Denver");
list.add("Austin");
list.add(new java.util.Date());
String city = list.get(0);
list.set(3, "Dallas");
System.out.println(list.get(3));
```

Pregunta

- Suposa que un ArrayList anomenat list conté els valors {"Dallas", "Dallas", "Houston", "Dallas"}.
 - ¿Cómo quedaria la llista després d'invocar a list.remove("Dallas") una volta?
 - Eliminaria el següent codi totes les ocurrències de "Dallas" de l'array? Si la resposta és negativa, corregeix el codi:

```
for (int i = 0; i < list.size(); i++)
list.remove("Dallas");</pre>
```



Pregunta

```
Explica per què el següent codi mostra [1, 3] i no [2, 3]
ArrayList<Integer> list = new ArrayList<>();
list.add(1);
list.add(2);
list.add(3);
list.remove(1);
System.out.println(list);
```

El següent codi és erroni?

```
ArrayList<Double> list = new ArrayList<>();
list.add(1);
```

Pràctica 1

 Fent ús d'ArrayList, crea un programa que demane 10 números enters, els guarde, i mostre per pantalla només aquells que no estiguen repetits.

Mètodes útils

 Per a transformar d'un array a un ArrayList no és necessari recórrer l'array per anar emplenant l'ArrayList:

```
String[] array = {"red", "green", "blue"};
ArrayList<String> list = new ArrayList<>(Arrays.asList(array));
```

```
String[] array1 = new String[list.size()];
list.toArray(array1);
```



Classe Collections

 Java disposa d'una classe amb mètodes static que serveixen de suport als ArrayList i altres estructures.

https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/Collections.html

```
Integer[] array = {3, 5, 95, 4, 15, 34, 3, 6, 5};
ArrayList<Integer> list = new ArrayList<>(Arrays.asList(array));
java.util.Collections.sort(list);
System.out.println(list);
System.out.println(java.util.Collections.max(list));
System.out.println(java.util.Collections.min(list));
java.util.Collections.shuffle(list);
System.out.println(list);
```

