USO DE LA COLECCIÓN ARRAYLIST

DIFICULTADES CON LOS ARRAYS

- Conocer a priori el tamaño
- El tamaño no se puede modificar una vez creado.
- Problemas para insertar objetos en posiciones intermedias
- No son realmente objetos
- …

COLECCIONES

Java provee todo un API de colecciones con decenas de interfaces y clases.

Beneficios

- Menos esfuerzo de programación.
- Aumento de la calidad y velocidad.
- Interoperabilidad
- Curva de aprendizaje pequeña
- Reusabilidad.

ARRAYLIST

- De todas las colecciones de Java es quizá la más usada.
- Estructura de datos secuencia

Operaciones

- Acceso posicional
- Búsqueda
- Iteración
- Tomar un fragmento

CONSTRUCCIÓN DE UN ARRAYLIST

ArrayList()

- Construye un *arraylist* con capacidad para 10 elementos.

ArrayList(Collection c)

- Construye un *arraylist* a partir de otra colección.

ArrayList(int initialCapacity)

- Construye un *arraylist* indicando la capacidad inicial.

CONSTRUCCIÓN DE UN ARRAYLIST

Hasta Java 1.4

- Todas las listas eran de Object

```
List cars = new ArrayList();
cars.add(new Object());
cars.add("car");
cars.add(new Integer(1));
```

CONSTRUCCIÓN DE UN ARRAYLIST

A partir de Java 1.5

- Inclusión de los genéricos
- Permiten parametrizar el tipo

```
List<String> cars = new ArrayList<String>();
```

A partir de Java 1.7

- Operador *diamond*
- Nos ahorra indicar dos veces el tipo

```
List<String> cars = new ArrayList<>();
```

ALGUNOS MÉTODOS DE ARRAYLIST

Nombre	Uso
add	Añade un elemento al final la lista
addAll	Añade todos los elementos de la colección pasada como argumento
clear	Elimina todos los elementos de la lista.
contains	Comprueba si un elemento está o no en la lista
get	Devuelve el elemento de la posición especificada de la lista
isEmpty	Verifica si la lista está vacía
remove	Elimina un elemento de la lista
size	Devuelve el número de elementos de la lista
toArray	Devuelve la lista como un array