**ITERATOR**

// Import the ArrayList class and the Iterator class

import java.util.ArrayList;

import java.util.Iterator;

public class Main {

public static void main(String[] args) {

// Make a collection

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

cars.add("Volvo");

cars.add("BMW");

cars.add("Ford");

cars.add("Mazda");

// Get the iterator

Iterator<String> it = cars.iterator();

// Print the first item

System.out.println(it.next());

}

}

**ARRAYLIST**

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

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

cars.add("Volvo");

cars.add("BMW");

cars.add("Ford");

cars.add("Mazda");

System.out.println(cars);

}

}

import java.util.HashMap;

public class Main {

public static void main(String[] args) {

// Create a HashMap object called capitalCities

HashMap<String, String> capitalCities = new HashMap<String, String>();

// Add keys and values (Country, City)

capitalCities.put("England", "London");

capitalCities.put("Germany", "Berlin");

capitalCities.put("Norway", "Oslo");

capitalCities.put("USA", "Washington DC");

System.out.println(capitalCities);

}

}

|  |  |
| --- | --- |
| clear() | Elimina todas las entradas del mapa. |
| clone() | Devuelve una copia de la instancia. Claves y valores no se clonan |
| compute(K key, BiFunction<? super K, ? super V,? extends V> remappingFunction) | Calcula un valor para la clave especificada y su valor asignado actual. |
| computeIfAbsent(K key, Function<? super K,? extends V> mappingFunction) | Si la clave no está asociada con un valor (o el valor es null), intenta calcular su valor utilizando la función pasada por parámetro y añade el valor. |
| computeIfPresent(K key, BiFunction<? super K, ? super V,? extends V> remappingFunction) | Si el valor de la clave existe y no es null, calcula un nuevo valor con el valor actual. |
| containsKey(Object key) | Devuelve true si existe la clave. |
| containsValue(Object value) | Devuelve true si existe el valor |
| entrySet() | Devuelve una vista de los valores del mapa. |
| get(Object key) | Devuelve el valor para esa key. |
| isEmpty() | Devuelve true si el mapa esta vacío |
| keySet() | Devuelve un conjunto de las claves del mapa. |
| merge(K key, V value, BiFunction<? super V, ? super V,? extends V> remappingFunction) | Si la clave no se encuentra se asocia con el valor que no sea null que se proporciona. |
| put(K key, V value) | Se asocia la clave con el valor. |
| putAll(Map<? extends K,? extends V> m) | Copia todo lo de un mapa en el otro. |
| remove(Object key) | Elimina el elemento proporcionado por la clave. |
| size() | Devuelve el número de elementos Clave/Valor del mapa. |
| values() | Devuelve la colección de valores del mapa. |