

Clase ArrayList

La **clase ArrayList** en Java, es una clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el numero de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño como pasa con los Arrays.

Un **ArrayList** se trata de un arreglo en el cual se en listan referencias a objetos mediante índices, el índice es la posición que ocupa un determinado objeto, como en todo arreglo la posición inicial o índice es 0. Cuando se agregan elementos a un objeto ArrayList su capacidad aumenta de forma automática.

Para la creación de un ArrayList declaramos su constructor que sería el siguiente:

```
ArrayList nombre_del_objeto=new ArrayList()
```

Metodos de la Clase ArrayList

boolean add(Objeto)

Agrega el elemento especificado al final de esta lista.

void add(int índice, Objeto)

Inserta el elemento especificado en la posición especificada en esta lista.

void clear()

Elimina todos los elementos de la lista.

boolean contains(Objeto)

Devuelve true si la lista contiene el elemento especificado.

get (int indice)

Devuelve el elemento en la posición especificada en la lista.

int indexOf (Objeto)

Nos devuelve la posición o índice del Objeto declarado en el parámetro del método.

boolean isEmpty()

Nos regresa true si la lista no contiene ningún elemento.

E remove(int index)

Quita el elemento en la posición especificada en esta lista.

boolean remove(Object o)

Quita la primera aparición del elemento especificado de la lista, si está presente.

E set(int index, E element)

Reemplaza el elemento en la posición especificada en esta lista con el elemento especificado.

int size()

Devuelve el numero de elementos almacenados en la colleccion

Ejemplo

//Ejemplo de la utilizacion de ArrayList con Objetos

```
package ArrayList_Objetos;

public class PartidoBaloncesto {

    private String equipoLocal;
    private String equipoVisitante;
    private int canastasLocal;
    private int canastasVisitante;

    public String getEquipoLocal() {
        return equipoLocal;
    }

    public void setEquipoLocal(String equipoLocal) {
        this.equipoLocal = equipoLocal;
    }

    public String getEquipoVisitante() {
        return equipoVisitante;
    }

    public void setEquipoVisitante(String equipoVisitante) {
        this.equipoVisitante = equipoVisitante;
    }

    public int get CanastasLocal() {
        return canastasLocal;
    }

    public void setCanastasLocal(int canastasLocal) {
        this.canastasLocal = canastasLocal;
    }

    public int getCanastasVisitante() {
        return canastasVisitante;
    }

    public void setCanastasVisitante(int canastasVisitante) {
        this.canastasVisitante = canastasVisitante;
    }
}
```

Vectores

Un Vector representa un array de objetos que puede crecer y reducirse, según el número de elementos que contenga.

Cuando creamos un vector, podemos especificar su dimensión inicial.

```
Vector vector=new Vector(20);
```

Añadir elemento a un vector

```
vector.add("hola");
```

Para acceder a los elementos del vector podemos utilizar un índice y un bucle, si queremos acceder a todos los elementos, o poner directamente la posición donde se encuentra el elemento que queremos mostrar o utilizar.

```
for (int i=0; i<10; i++)  
    { return(this.vector[i]);  
    }
```