


ArrayList

Declaración y Creación de Arreglos

```
8      *
9      * @author JAIRO F
10     */
11     public class Arreglos {
12
13         /**
14          * @param args the command line arguments
15          */
16         public static void main(String[] args) {
17             // TODO code application logic here
18             int [] numeros = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
19
20             System.out.printf("Posicion    -    Valor \n");
21             for(int i=0; i<numeros.length;i++){
22                 System.out.printf("%d    -    %d \n",i,numeros[i]);
23             }
24
25
26     }
27 }
```

Paso de Arreglos a Métodos

```
int [] arreglo2 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
```

```
30  
31  public void metodo (int [] arreglo2){  
32  
33     //instruccion 1  
34     //instruccion 2  
35     //instruccion n  
36  
37 }
```

```
pa.metodo(arreglo2); // Llamada del Método
```

Paso de Arreglos a Métodos

```
2  package arreglos;
3  public class PasoArreglo {
4
5      private int [] arreglo;
6
7      public PasoArreglo(int[] arreglo) {
8          this.arreglo = arreglo;
9      }
10
11     public PasoArreglo() { }
12
13     public int[] getArreglo() {
14         return arreglo;
15     }
16
17     public void setArreglo(int[] arreglo) {
18         this.arreglo = arreglo;
19     }
20
21     public int sumarArreglo(int [] arreglo2){
22         this.arreglo = arreglo2;
23         int suma=0;
24         for(int valor:arreglo)
25             suma+=valor;
26         return suma;
27     }
28 }
```

```
8 public class Arreglos {
9
10 /**
11  * @param args the command line arguments
12  */
13 public static void main(String[] args) {
14     // TODO code application logic here
15     int [] numeros = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
16
17     System.out.printf("Posicion    -    Valor \n");
18     for(int i=0; i<numeros.length;i++){
19         System.out.printf("%d    -    %d \n",i,numeros[i]);
20     }
21
22     // agregar esta parte
23     PasoArreglo pa = new PasoArreglo();
24     System.out.printf("Suma de los elementos: %d \n", pa.sumarArreglo(numeros));
25
26 }
27
28 }
```

La clase Arrays - java.util

En el paquete **java.util** se encuentra una clase estática llamada **Arrays**. Una clase estática permite ser utilizada como si fuera un objeto (como ocurre con **Math**). Esta clase posee métodos muy interesantes para utilizar sobre arreglos.

Su uso es : **Arrays.método(argumentos);**

fill - Permite rellenar todo un array unidimensional con un determinado valor. **int**

```
valores[]=new int[23];  
Arrays.fill(valores,-1);//Todo el array vale -1
```

Equals - Compara dos arrays y devuelve true si son iguales. Se consideran iguales si son del mismo tipo, tamaño y contienen los mismos valores.

Sort - Permite ordenar un array en orden ascendente.

```
int x[]={4,5,2,3,7,8,2,3,9,5};  
Arrays.sort(x);//Estará ordenado
```

binarySearch - Permite buscar un elemento de forma ultrarrápida en un array ordenado. Devuelve el índice en el que está colocado el elemento. Ejemplo:

```
int x[]={1,2,3,4,5,6,7,8,9,10,11,12};  
Arrays.sort(x);  
System.out.println(Arrays.binarySearch(x,8));//Da 7
```


Clase ArrayList

Implementa la interfaz List

Estas clases proveen **métodos** que ayudan a reducir el tiempo de desarrollo de aplicaciones

Proviene del paquete **Java.util**

Cambia su tamaño de forma **dinámica** para tener mas elementos

Solo se puede **implementar** con tipos de datos no primitivos.

M
E
T
O
D
O
S



MÉTODO	DESCRIPCIÓN
size()	Devuelve el número de elementos (int)
add(X)	Añade el objeto X al final. Devuelve true.
add(posición, X)	Inserta el objeto X en la posición indicada.
get(posicion)	Devuelve el elemento que está en la posición indicada.
remove(posicion)	Elimina el elemento que se encuentra en la posición indicada. Devuelve el elemento eliminado.
remove(X)	Elimina la primera ocurrencia del objeto X. Devuelve true si el elemento está en la lista.
clear()	Elimina todos los elementos.
set(posición, X)	Sustituye el elemento que se encuentra en la posición indicada por el objeto X. Devuelve el elemento sustituido.
contains(X)	Comprueba si la colección contiene al objeto X. Devuelve true o false.
indexOf(X)	Devuelve la posición del objeto X. Si no existe devuelve -1

Objeto ArrayList

Creación de objetos ArrayList

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

```
ArrayList < clase > nombre_de_la_coleccion = new ArrayList < clase > ();
```

```
ArrayList <String> coleccion = new ArrayList();
```


Objeto ArrayList

```
1  package colecciones;
2
3  public class Estudiante {
4      private String nombre;
5      private String Apellido;
6      private String cedula;
7
8      public Estudiante(String nombre, String Apellido, String cedula) {
9          this.nombre = nombre;
10         this.Apellido = Apellido;
11         this.cedula = cedula;
12     }
13     public Estudiante() { }
14     public String getNombre() {
15         return nombre;
16     }
17     public void setNombre(String nombre) {
18         this.nombre = nombre;
19     }
20     public String getApellido() {
21         return Apellido;
22     }
23     public void setApellido(String Apellido) {
24         this.Apellido = Apellido;
25     }
26     public String getCedula() {
27         return cedula;
28     }
29     public void setCedula(String cedula) {
30         this.cedula = cedula;
31     }
32 }
```



Objeto ArrayList

```
5 package colecciones;
6
7 import java.util.ArrayList;
8
9 public class Colecciones {
10
11     public static void main(String[] args) {
12         // TODO code application logic here
13         ArrayList <Estudiante> listado = new ArrayList();
14         Estudiante e1,e2,e3;
15
16         e1= new Estudiante("Jairo","Seoanes","77097016");
17         listado.add(e1);
18         e2=new Estudiante("Luis C.","Galan S","109876534");
19         listado.add(e2);
20         e3=new Estudiante("Pablo E.","Escobar G.","66666666");
21         listado.add(e3);
22
23         System.out.println("Listado de Estudiante: ");
24         for(Estudiante e: listado){
25             System.out.println(e.getCedula()+" "+e.getApellido()+" "+e.getNombre());
26         }
27     }
28 }
```



EJERCICIO

Se necesita implementar un sistema en el que se puedan cargar alumnos, a los cuales los caracterizan el nombre y apellido, el código, el sexo, condición (regular o condicional) y la nota final. Estos alumnos se deben cargar en una asignatura, que adicionalmente debe poseer una descripción (nombre) y el programa al que pertenece.

Implemente las clases y métodos necesarios para esta situación, y métodos que permitan realizar lo que se pide a continuación:

- a. Crear asignaturas
- b. Agregar estudiante a la asignatura
- c. Mostrar en pantalla todos los alumnos que se encuentren en la asignatura.
- d. Mostrar en pantalla los alumnos que se encuentren como condicional y su cantidad.
- e. Calcular el promedio de notas de la asignatura
- f. Consultar la información de un estudiante a través de su código