

UT 1. POO, EDA y API JAVA.

Tipos compuestos de datos.

Arrays simples.

Arrays de una dimensión.

- Un array es un conjunto de posiciones de memoria, todas del mismo tipo, que se referencian con el mismo nombre y un índice para cada posición.
- Los corchetes [] son los que denotan que la estructura de datos es array. Los arrays se declaran de acuerdo a una de los dos siguientes formatos:

```
<tipo> [ ] <identificador_array>;  
<tipo> <identificador_array> [];
```

```
<tipo> [ ] <identificador_array> = new <tipo>[<numero_  
elementos>];  
<tipo> <identificador_array> [ ] = new <tipo>[<numero_  
elementos>];  
Si el identificador de array ya esta declarado con  
anterioridad:  
<identificador_array> = new <tipo>[<numero_  
elementos>];
```

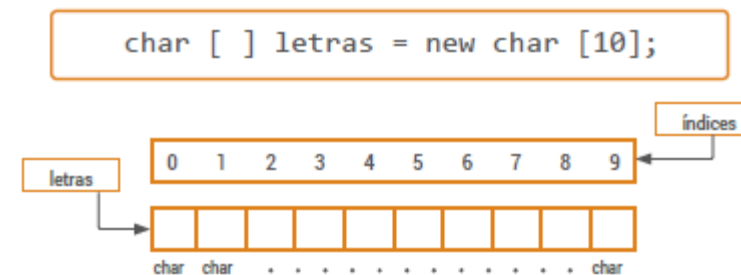
Arrays de una dimensión.

- Ejemplos

```
int listanumeros [ ]; //listanumeros identificador para
                      //acceder al array de int
listanumeros = new int [10]; //creación de array
para 10 int,
                      //que podrá ser utilizado con identificador
listanumeros

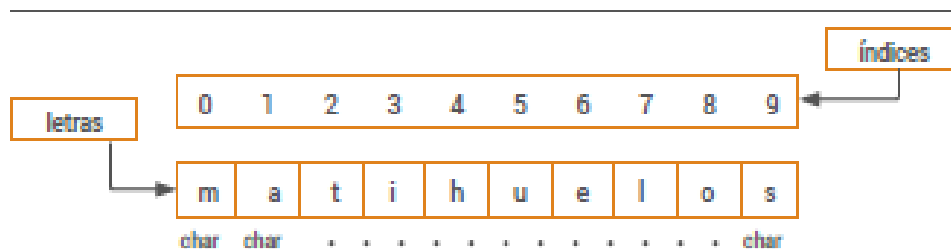
char letras = new letras [10];
```

- Es el operador new el que crea las posiciones de memoria consecutivas y del mismo tipo que se podrán utilizar con el identificador declarado.
- Todo identificador de array tiene asociado el atributo llamado **length**



Arrays de una dimensión

- Para acceder a los elementos de un array usamos: `<identificador_array> [<indice>]`
- Utilizamos `length` para controlar que el valor utilizado nunca es igual o superior al tamaño del array.



`letras[2] ==> tiene la letra 't'`

`letras[8] = 'a' ==> se almacena 'a' en posición 9 e índice 8`

`letras[10] = 'a' ==> excepcion IndexOutOfBoundsException.`

```
Random aleatorio = new Random();
int size = aleatorio.nextInt(25)+1;
int lista [] = new int[size];
//Recorrer el array y almacenar un valor en cada
posición
for (int i=0; i< lista.length; i++){
    lista[i] = aleatorio.nextInt(1000);
}
//Recorrer el array y obtener el valor de cada posición
for (int i=0; i< lista.length; i++){
    System.out.println("lista["+i+"] :"+lista[i]);
}
```

Arrays de una dimensión

- Cuando se crea un array también puede inicializarse con valores concretos y determinados, el formato es el siguiente:

```
<tipo> [ ] <identificador_array> = { <lista_valores_tipo> }
```

```
//array de cinco char
```

```
char vocales [ ] = {'A', 'E', 'I', 'O', 'U'};
```

```
//array de 10 números, es opcional poner el tamaño
```

```
int lista [10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

```
//array de tres cadenas
```

```
String nombres [ ] = {"Pepe Pérez", "Gil Lopez", "Ana Ruiz"};
```

Bucle for-each.

-
- Bucle muy indicado para el recorrido de arrays.
 - Permite recorrerlos sin usar índices, ni el valor lenght.
 - Asigna a la variable **<identificador>** el valor de cada uno de los elementos de la colección y después ejecuta las **<sentencias>**.
 - En <sentencias> no puede haber ninguna que cambie el tamaño de **<colección>**, ni el valor de ningún elemento.

```
for ( <tipo> <identificador> : <coleccion> ){  
    <sentencias>  
}
```

Bucle for-each.

- El bucle ***for each*** es para recorrer colecciones y acceder a los elementos para obtener su valor.

```
Random aleatorio = new Random();
    int size = aleatorio.nextInt(25)+1;
    int lista [] = new int[size];
    //Recorrer el array y almacenar un valor en cada posición
    for (int i=0; i< lista.length; i++){
        lista[i] = aleatorio.nextInt(1000);
    }
    //Recorrer el array y obtener el valor de cada posición
    int pos=0;
    for(int n : lista)
        System.out.println("Posicion "+(++pos)+" : "+n);
```

Ejercicios Arrays

-
- Ejercicio_1: Leer 5 números, guardarlos en un array y mostrarlos en el mismo orden introducido.
 - Ejercicio_2: Leer 5 números, guardarlos en un array y mostrarlos en el orden inverso al introducido.
 - Ejercicio_3: Leer 5 números por teclado, almacenarlos en un array y a continuación realizar la media de los números positivo, la media de los negativos y contar el número de ceros.
 - Ejercicio_4: Leer 10 números enteros, guardarlos en un array. Debemos mostrarlos en el siguiente orden: el primero, el último, el segundo, el penúltimo, el tercero, etc..
 - Ejercicio_5: Leer por teclado dos tablas de 10 números enteros y mezclarlas en una tercera de la forma: el 1º de A, el 1º de B, el 2º de A, el 2º de B, etc....
-