

ARREGLOS PRIMITIVOS

UNIDAD 2: CLASES Y OBJETOS

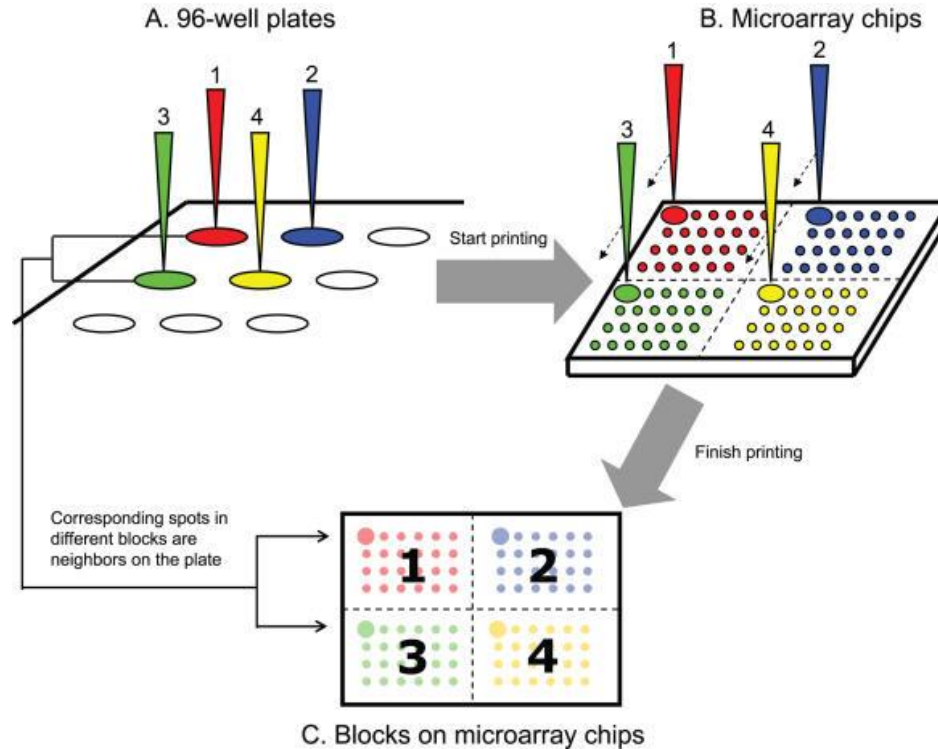
SEMANA 6



Temario

- Arreglos de primitivos: unidimensionales, bidimensionales.

MOTIVACIÓN



1. Arreglos de tipo primitivo de una sola dimensión: Definición

- Son Estructuras de Datos que contienen elementos relacionados del mismo tipo.
- Mantienen el mismo tamaño una vez que han sido creados

Arreglo de 10 posiciones

Arreglo de enteros
llamado **arNum**

Índice del elemento
en el arreglo.

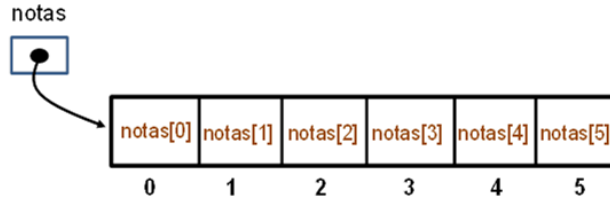
Valor del arreglo en
la posición del índice.

arNum [0]	- 45
arNum [1]	6
arNum [2]	10
arNum [3]	72
arNum [4]	11
arNum [5]	-9
arNum [6]	221
arNum [7]	87
arNum [8]	14
arNum [9]	22

1. Arreglos de tipo primitivo de una sola dimensión: Definición

Ejemplo1:

```
int [] notas;  
notas = new int [6];
```



O también:

```
int [] notas = new int [6];
```

Ejemplo2:

La asignación de valores a los elementos de un arreglo se hace de las siguientes 2 formas:

Forma 1

```
int [] datos;  
datos = new int [3];  
datos [0] = 250;  
datos [1] = 120;  
datos [2] = 174;
```

1. Arreglos de tipo primitivo de una sola dimensión: Recorrido

- Para recorrer un arreglo se utiliza alguna estructura de repetición, preferiblemente for.
- La variable contador del ciclo maneja el índice del arreglo y el límite es la variable length

Ejemplo3:

```
// declaramos e inicializamos un arreglo de enteros de 5 posiciones  
// (del 0 al 4) usando una lista de valores.
```

```
int [ ] numeros= {10, 20, 30, 40, 50};  
for (int i = 0; i < numeros.length ; i++){  
    System.out.println ("índice " + i + "valor " + numeros [i])  
}
```

1. Arreglos de tipo primitivo de una sola dimensión: Sintaxis

- La sintaxis para declarar e inicializar un array será:
 - `Tipo_de_variable[] Nombre_del_array = new Tipo_de_variable[dimensión];`
- También podemos alternativamente usar esta declaración:
 - `Tipo_de_variable[] Nombre_del_array;`
 - `Nombre_del_array = new Tipo_de_variable[dimensión];`

1. Arreglos de tipo primitivo de una sola dimensión: Sintaxis

- El tipo de variable puede ser cualquiera de los admitidos por Java. Ejemplos de declaración e inicialización con valores por defecto de arrays usando todos los tipos de variables Java, serían:
 - `byte[] edad = new byte[4];`
 - `short[] edad = new short[4];`
 - `int[] edad = new int[4];`
 - `long[] edad = new long[4];`
 - `float[] estatura = new float[3];`
 - `double[] estatura = new double[3];`
 - `boolean[] estado = new boolean[5];`
 - `char[] sexo = new char[2];`
 - `String[] nombre = new String[2];`

1. Arreglos de tipo primitivo de una sola dimensión: Sintaxis

- Valores numéricos entero:
 - `int[] edad = {45, 23, 11, 9};` //Array de 4 elementos
 - De la misma forma procederíamos para los otros tipos de enteros : byte, short, long.
- Valores del tipo cadena:
 - `String[] nombre = {"María", "Gerson"};` //Array de 2 elementos
- Valores del tipo booleano:
 - `boolean[] nombre = {true,false};` //Array de 2 elementos

1. Arreglos de tipo primitivo de una sola dimensión: Sintaxis

- Valores numéricos reales o de punto flotante:
 - `double[] estatura = {1.73, 1.67, 1.56};` //Array de 3 elementos
 - De la misma forma procederemos para el tipo float, pero teniendo en cuenta que los números deberán llevar al final la letra “f” o “F”. Por ejemplo 1.73f o 1.73F.
- Valores del tipo caracteres:
 - `char[] sexo = {'m', 'f'};` //Array de 2 elementos

1. Arreglos de tipo primitivo de una sola dimensión: Sintaxis

- Dimensionamiento de arreglos:
 - Cuando creamos un array de nombre “a” y de dimensión “n” (`int[] a = new int[n]`) estamos creando n variables que son `a[0]`, `a[1]`, `a[2]`, ..., `a[n-1]`. Los arrays se numeran desde el elemento cero, que sería el primer elemento, hasta el `n-1` que sería el último elemento. Es decir, si tenemos un array de 5 elementos, el primer elemento sería el cero y el último elemento sería el 4

1. Arreglos de tipo primitivo de una sola dimensión: Sintaxis

- En caso se acceda a una posición no válida del arreglo, se muestra una Excepción:

```
public static void main(String[] args)
{
    String[] nombre = new String[4];
    nombre[0] = "Luis";
    nombre[1] = "María";
    nombre[2] = "Carlos";
    nombre[3] = "Jose";
    nombre[4] = "Ismael"; //Error:No existe esta variable array de índice 4
}
```

```
run:
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 4
    at mi.paquete.principal.UnDiaEnLaClinica.main(UnDiaEnLaClinica.java:25)
Java Result: 1
BUILD SUCCESSFUL (total time: 1 second)
```

2. Arreglos Bidimensionales - Definición

- Un **arreglo** bidimensional, tabla o matriz es un arreglo con dos índices. Para localizar un elemento en el arreglo se deben especificar dos posiciones (subíndices), uno para la fila y otro para la columna.
- En Java una matriz es un vector de vectores filas, es por eso que cada fila podría tener un número de elementos diferentes
- Un **arreglo** bidimensional, es una matriz con filas y columnas. El primer índice indica la fila y el segundo la columna.
- Ejemplo:
 - $P=[4][3]$
 - Se refiere al elemento ubicado en la cuarta fila y tercera columna

2. Arreglos Bidimensionales - Declaración

- Declaración de arreglo bidimensional:
 - `tipoDato [][] nombreDelArreglo;`
- La sintaxis para crear arreglos es:
 - `nombreDelArreglo = new tipoDato [numFilas][numColumnas];`

2. Arreglos Bidimensionales - Declaración

- Ejemplo de declaración:
 - `int a[][] = { (4,2,7),(4,3,8),(6,2,7),(9,1,0) } ;`

4	2	7
4	3	8
6	2	7
9	1	0

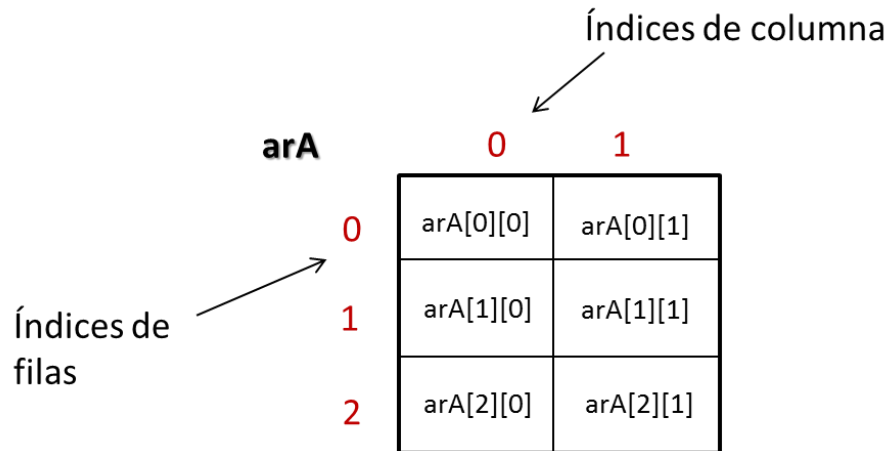
2. Arreglos Bidimensionales - Recorrido

- El recorrido de un arreglo de dos dimensiones se hará mediante el uso de dos estructuras repetitivas

```
for(int i=0; i<20; i++)  
{  
    for(int j=0; j<10; j++)  
    {  
        puntaje[i][j]=rand.nextInt(10)+1;  
    }  
}
```

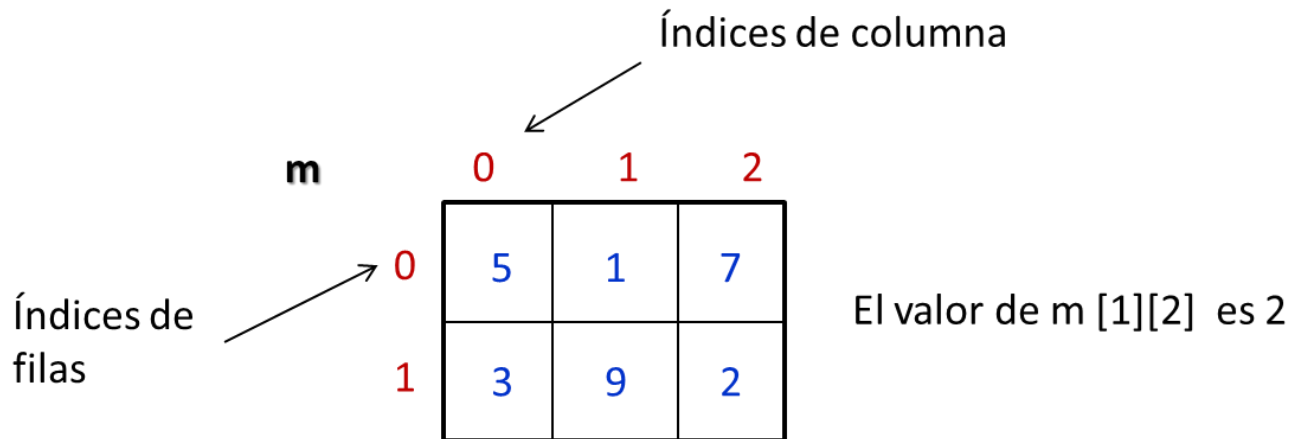

2. Arreglos Bidimensionales - Ejemplos

- Ejemplo1:
 - `int [][] arA;`
 - `arA = new int [3][2];`
- Matriz llamada arA con 3 filas y 2 columnas, distribuidas de la siguiente manera:



2. Arreglos Bidimensionales - Ejemplos

- Ejemplo2:
 - Creación de una matriz de dos filas por tres columnas con valores iniciales.
 - `int m [] = {{5, 1, 7}, {3, 9, 2}};`



REFERENCIAS

1. Paul S. Wang, Java con programación orientada a objetos y aplicaciones en la WWW, México, 2000.
2. Sun microsystem, Fundamentals of the Java™ Programming Language SL-110-SE6
3. A.M. Vozmediano, Java para novatos, 2017.
4. BEGOÑA MOROS VALLE, <http://dis.um.es/~bmoros/>