Arreglos (Arrays) en Java

Definición: Un arreglo es un objeto contenedor que consiste de una cantidad fija de posiciones o celdas para almacenar valores del mismo tipo en ellas.

Cada posición o celda del arreglo tiene un identificador o nombre común, llamado el nombre del arreglo. Para diferenciar entre las diferentes posiciones o celdas que forman el arreglo, se hace uso de uno o más índices en sus identificadores respectivos, dependiendo del tipo de arreglo particular que se defina.

Físicamente, se puede suponer un arreglo como un conjunto secuencial de posiciones en memoria principal a las que se accede a través de uno o más índices de posición. La cantidad de memoria que se le asigna a un arreglo dependerá del tipo de arreglo y de la cantidad de posiciones o celdas en él.

Los arreglos en Java cuentan con propiedades y métodos para manipularlos. Se pueden declarar arreglos de tipos de datos primitivos y de objetos. Al igual que para todos los objetos, una declaración de arreglo no crea un arreglo. Para crear un arreglo se usa el operador new; como consecuencia, se le asigna memoria para almacenar los objetos en el arreglo.

Arreglos Unidimensionales

En este tipo de arreglo se hace uso de un índice solamente para hacer referencia a una posición particular del arreglo.
Ejemplos de declaraciones de arreglos unidimensionales:

1. doublé [ ] vector;
2. doublé vector [ ];

Ambas formas anteriores son equivalentes.

3. String [ ] nombres;
4. boolean [ ] valores_booleanos;
5. int [ ] enteros;

Para asignarle posiciones de memoria principal a un arreglo se usa el operador new.
Se puede usar una sola oración para declarar y asignarle memoria a un arreglo.
Ejemplos:

1. vector = new double [12]; (Se crea un arreglo de tamaño 12 y cada posición se inicializa a 0.)
2. doublé [ ] vector = new double [12]; (Equivalente al ejemplo 1.)
3. String [ ] nombre = new String [100]; (El identificador de la primera posición de un arreglo unidimensional tiene índice 0.)
4. boolean [ ] valor_booleano = new boolean [4]; (Cada variable booleana se inicializa con el valor false.)
5. int [ ] entero = new int [10]; (Los identificadores de las posiciones son entero[0], entero[1], ..., entero[9].)

En los ejemplos 1 y 2, ambas formas son equivalentes y se crea un arreglo con 12 posiciones cuyos identificadores respectivos tienen subíndices del 0 al 11.
Notas:

1. Cuando se crea un arreglo, cada una de sus posiciones será inicializada con el valor por defecto (default) del tipo de dato u objeto. Por ejemplo, si el arreglo es de tipo boolean , todas las posiciones del arreglo serán inicializadas con el valor false (ya que este es valor por defecto del tipo de datos boolean). Por otro lado, si el arreglo fuese de un tipo no primitivo o elemental, el valor que contendrá cada posición será null .
2. Se le puede asignar valores a un arreglo al momento de declararlo o luego, cuando sea necesario.

Ejemplos:

1. int [ ] arreglo = new int[4];
2. int [ ] arreglo = {100,200,302,400};

Nota: La constante length está asociada con un arreglo y su valor es el tamaño del arreglo.
Ejemplo: for (int i = 0; i < vector.length; i = i + 1;) {
...//cuerpo del ciclo.
}

Ejemplos de Aplicaciones con Arreglos Unidimensionales:

1. Sumar todos los valores en un arreglo de tamaño 10 y mostrar el resultado de la suma. /* Suma de todos los valores de los elementos de un arreglo.

```
/*Archivo SumarArreglo.java */
public class SumarArreglo {
    public static void main( String [ ] args ){
            int arreglo[ ] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
            int suma = 0;
            System.out.println ("Esta aplicación suma todos los elementos de un arreglo y muestra el resultado.");
            // Ciclo for para sumar todos los valores del arreglo.
            for (inti = 0; i < arreglo.length; i++ )
                    suma+= arreglo[i];//forma equivalente a suma = suma+arreglo[i]
            System.out.println("Resultado de la suma: " + suma + ".");
    } // fin de la clase principal main
} // fin de la clase SumarArreglo
```

2. Simular la tirada de un dado de seis lados 600 veces, contar y mostrar la frecuencia de cada resultado.

Versión no modular:

```
/* Tirar un dado de seis lados la cantidad de veces indicada, sumar y mostrar la frecuencia de cada resultado.
Archivo: TirarDados.java */
public class TirarDado {
    public static void main( String[ ] args )
    {
            int frecuencia[ ] = new int [7];
            // Ciclo for para tirar un dado 600 veces; usar el valor del dado como índice de frecuencia
            for ( inttirada = 1; tirada <= 600; tirada++ ){
```

```
            int cara = (int)Math.floor(6* Math.random()) +1;//(int) convierte el resultado
            en un entero.
            frecuencia[cara] = frecuencia[cara] + 1;
        }//termina ciclo for
        System.out.println("Cara Frecuencia");
        for ( intcara = 1; cara < frecuencia.length; cara++ ){
            System.out.println(" " +cara + " " + frecuencia[cara]);
        }
    } // fin de main
} // fin de la clase TirarDado
```

Versión Modular 1:

```
/* Tirar un dado de seis lados la cantidad de veces indicada, sumar y mostrar la frecuencia de
cada resultado.
Archivo: TirarDados.java */
import java.util.Scanner;
public class TirarDado {

    public static void main( String[ ] args ) {
        mensajeInicial();
        int cantidadTiradas = entrada();
        tiraDados(cantidadTiradas);
    } // termina main

    public static void mensajeInicial(){
        //Explica el propósito de la aplicación.
        System.out.println("Esta aplicación simula la tirada de un dado la cantidad de veces
        que se indique y muestra la frecuencia de cada resultado.");
    }//termina mensajeInicial

    public static int entrada(){
        //Solicita la cantidad de veces que se desea tirar el dado.
        Scanner lectorTeclado = new Scanner(System.in);
        System.out.println();
        System.out.println("Indique la cantidad de veces que desea tirar el dado.");
        int cantidadTiradas = lectorTeclado.nextInt();
        return cantidadTiradas;
    }//termina entrada

    public static void tiraDados(int cantidadTiradas){
        //Simula la tirada de un dado la cantidad de veces indicada.
        int frecuencia[ ] = new int[7];
        for (int tirada = 1; tirada <= cantidadTiradas; tirada++ ){
            int cara = (int)Math.floor(6* Math.random()) +1;
            frecuencia[cara] = frecuencia[cara] + 1;
        }//termina ciclo for
    salida(frecuencia);
```

```
    }//termina tiraDados

    public static void salida(int frecuencia[]){
        System.out.println("Cara Frecuencia");
        for (int cara = 1; cara < frecuencia.length; cara++ ){
            System.out.println(" " +cara + " " + frecuencia[cara]);
        } //termina ciclo for
    }//termina salida

} // fin de la clase TirarDado
```

Versión Modular 2:

/* Tirar un dado de seis lados la cantidad de veces indicada, sumar y mostrar la frecuencia de cada resultado.

```
/*Archivo: TirarDados.java */
import java.util.Scanner;
public class TirarDado {

    public static void main( String[ ] args ) {
        mensajeInicial();
        int cantidadTiradas = entrada();
        int frecuencia[] = tiraDados(cantidadTiradas);
        salida(frecuencia);
    } // termina main

    public static void mensajeInicial(){
        //Explica el propósito de la aplicación.
        System.out.println("Esta aplicación simula la tirada de un dado la cantidad de veces
        que se indique y muestra la frecuencia de cada resultado.");
    }//termina mensajeInicial

    public static int entrada(){
        //Solicita la cantidad de veces que se desea tirar el dado.
        Scanner lectorTeclado = new Scanner(System.in);
        System.out.println();
        System.out.println("Indique la cantidad de veces que desea tirar el dado.");
        int cantidadTiradas = lectorTeclado.nextInt();
        return cantidadTiradas;
    } //termina entrada

    public static int [] tiraDados(int cantidadTiradas){
        //Simula la tirada de un dado la cantidad de veces indicada.
        int frecuencia[] = new int[7];
        for (int tirada = 1; tirada <= cantidadTiradas; tirada++ ){
            int cara = (int)Math.floor(6* Math.random()) +1;
            frecuencia[cara] = frecuencia[cara] + 1;
```

```
            }//termina ciclo for
            return frecuencia;
        }//termina tiraDados

        public static void salida(int frecuencia[]){
            System.out.println("Cara Frecuencia");
            for (int cara = 1; cara < frecuencia.length; cara++ ){
                System.out.println(" " +cara + " " + frecuencia[cara]);
            } //termina ciclo for
        }
        //termina salida

} // fin de la clase TirarDado
```

Trabajo Practico - Ejercicios: (Arreglos unidimensionales)

1. ¿Qué sucede si tratamos de acceder un elemento fuera del tamaño del array?

2. Crea un array o arreglo unidimensional con un tamaño de 5, asignale los valores numéricos manualmente (los que tú quieras) y muestralos por pantalla. Solicite los números mediante un bucle.

3. Crea un array o arreglo unidimensional donde tú le indiques el tamaño por teclado y crear una función que rellene el array o arreglo con los múltiplos de un numero pedido por teclado. Por ejemplo, si defino un array de tamaño 5 y elijo un 3 en la función, el array contendrá 3, 6, 9, 12, 15. Muéstralos por pantalla usando otra función distinta.

4. Escriba una aplicación que:
a) lea 20 números decimales ingresados por teclado por el usuario y los coloque en un arreglo unidimensional.
b) determine y muestre el mayor de los números en el arreglo
c) determine y muestre el menor de los números del arreglo calcule y muestre el rango (diferencia entre el mayor y el menor) de los elementos en el arreglo

5. Escriba una aplicación para almacenar aleatoriamente 20 números enteros positivos pares del 1 al 100, de los cuales se desea saber:
a) su promedio aritmético
b) cuántos de los números son iguales al promedio aritmético
c) cuántos de los números son mayores que el promedio aritmético
d) cuántos de los números son menores que el promedio aritmético

6. Realice una búsqueda secuencial en un arreglo unidimensional de tamaño 50 generado aleatoriamente mediante la función java **random**.
Realice una búsqueda secuencial de la siguiente forma:
- se lee el valor que se desea buscar,
- se compara la primera posición;
- si son iguales,
- fin de la búsqueda.
- De lo contrario, compararlo con la segunda posición, y así sucesivamente.
- Si se llega al final del arreglo y no se encontró el valor, debe indicarlo con un mensaje apropiado.
- Si se encuentra, se debe especificar la posición que ese valor ocupa en el arreglo por primera vez.

7. Escriba una aplicación que solicite y cargue en un arreglo 10 dígitos enteros, luego cree dos nuevos arreglos y asigne al primero los números ingresados por el usuario de forma ascendente y en el segundo de forma descendente. Muestre los 2 arreglos por pantalla.

8. Solicite al usuario el ingreso de una cadena de números separadas por guiones medio, por ejemplo:
45-9-8-6-45-23-21-74-96-32-45-25

Posteriormente aplique el método **SPLIT** de los String de Java para convertir la cadena en un arreglo, y calcular la suma total de los elementos y el valor promedio calculado.

9. En Argentina cada persona está identificada con un Documento Nacional de Identidad (DNI) en el que figura un número y una letra, por ejemplo 56999545W

La letra que sigue al número se calcula siguiendo la metodología que vamos a indicar. Crea un programa que calcule la letra de un DNI a partir del número de DNI que introduzca el usuario. Es decir, se debe pedir el DNI sin la letra por teclado y el programa nos devolverá el DNI completo (con la letra).

Para calcular la letra, se debe tomar el resto de dividir nuestro número de DNI entre 23. El resultado debe estar por tanto entre 0 y 22.

Crea un método obtenerLetra(int numeroDNI) donde según el resultado de la anterior fórmula busque en un array de caracteres la posición que corresponda a la letra. Esta es la tabla de caracteres:

Posición	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Letra	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

Por ejemplo, si introducimos el DNI 20267079, el resto de dividirlo por 23 sería 8, luego la letra sería la P, que es la que ocupa esa posicion en la matriz de caracteres.

10. Dado 2 array de enteros, el primero de tamaño 5 y el segundo de tamaño 10, pedir los valores numéricos enteros para cargar cada uno de los arreglos, cree un tercer arreglo de tamaño 10 que contenga en casa posición la suma de la multiplicación de cada elemento del array uno por cada elemento del array 2
ARRAY3 = (posición 0 del arreglo 1 * posición 0 del arreglo 2) + (posición 0 del arreglo 1 * posición 1 del arreglo 2) + (posición 0 del arreglo 1 * posición 2 del arreglo 2) + ……..+ (posición 4 del arreglo 1 * posición 9 del arreglo 2)
Use 2 estructuras iterativas anidadas para realizar el cálculo y asignación de los valores en el 3 array.