


| | |
|---|---|
|  | <p style="text-align: center;">UNIVERSIDAD DON BOSCO FACULTAD DE ESTUDIOS TECNOLÓGICOS COORDINACIÓN DE COMPUTACIÓN Y MÓVILES</p> |
| <p style="text-align: center;">Ciclo I</p> | <p style="text-align: center;">Programación Orientada a Objetos Guía de Laboratorio No. 6 Arreglos</p> |

I. OBJETIVOS.

- Introducir al alumno en el desarrollo de aplicaciones en JAVA.
- Que el alumno conozca los principios básicos de la programación estructurada a partir del concepto de Arreglos

II. INTRODUCCIÓN.

En java, a diferencia del lenguaje C, existe un tipo de variable “especial”, el Array o matriz. Este tipo de variables no es más que un conjunto secuencial de memoria a las que se accede a través de un índice de posición.

Las matrices en Java son objetos, por lo que cuentan con propiedades y métodos para manipularlos. Se pueden declarar matrices de tipos de datos primitivos y de objetos.

¿Cómo se declaran?

```
tipodedato nombre [] = new tipodedato[capacidad];
```

```
tipodedato[] nombre = new tipodedato[capacidad] ;
```

Para inicializar un array existen 2 maneras:

```
int[] matriz=new int[4];
```

```
int[] matriz=
```

```
{ 100,200,302,400} ;
```

Como se puede concluir, en la primera forma se declara el array nada más, diciéndole la cantidad de memoria secuencial que se debe reservar, en el segundo ejemplo se inicializa el array dándole los valores que va a contener (obviamente la cantidad de memoria secuencial reservada será igual a la cantidad de elementos insertados). Al momento de inicializar una matriz de la manera:

int[] matriz=new int[4] cada posición de la matriz será inicializada con el valor predeterminado del tipo de variable. Por ejemplo si la matriz es de tipo boolean, todas las posiciones de la matriz serán inicializadas con el valor false (ya que este es valor predeterminado de este tipo de dato), por el contrario si la matriz fuese de un tipo no primitivo, el valor que contendrá cada casilla será null.^[1]_{SEP}

for mejorado

Supongamos que tenemos un arreglo de enteros que deseamos presentar en pantalla, usando la orden for tradicional, el código, podría quedar así:

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

```
for (int i = 0; i < 10; i ++)
```

```
System.out.println (laiEnteros[i]);
```

Java ofrece una funcionalidad extra para la orden for, mediante la que se puede simplificar notablemente el código anterior, quedando así:

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

```
for (int liElemento : laiEnteros)
```

```
System.out.println (liElemento);
```

III. Material y equipo necesario.

Para la realización de la guía de práctica se requerirá lo siguiente:

| No. | Requerimiento | Cantidad |
|-----|-------------------------------|----------|
| 1 | Guía de Laboratorio #5 de PRO | 1 |

| | | |
|---|-----------------------------|---|
| 2 | Computadora con Eclipse IDE | 1 |
|---|-----------------------------|---|

IV. Procedimiento.

Ejemplo 01: Arreglos

```
public class Ejemplo01 {
    public static void main(String[] args) {
        String nombres[] = new String[5];
        nombres[0] = "Leonid Brezhnev";
        nombres[1] = "Jhon F. Kenedy";
        nombres[2] = "Harry Truman";
        nombres[3] = "Mijail Gorbachov";
        nombres[4] = "Nelson Mandela";
        for (int i = 0; i < nombres.length; i++) {
            System.out.println(nombres[i]);
        }
    }
}
```

Ejemplo 02:

Ingresando manualmente los elementos de un arreglo.

```
import java.util.Scanner;

public class Ejemplo02 {
    public static void main(String[] args) {

        Scanner reader = new Scanner(System.in);
        System.out.println("Cuántos elementos desea ingresar?");
        int num_elementos = reader.nextInt();
        int[] numeros = new int[num_elementos];
        for (int i = 0; i < num_elementos; i++) {
            System.out.print("Ingrese el valor indice[" + i + "]: ");
            numeros[i] = reader.nextInt();
        }
        for (int valor : numeros) { // for mejorado System.out.println(valor);
        }
    }
}
```

```
}
```

Ejemplo 3:

Uso de arreglos.

```
public class Ejemplo03 {  
    public static void main(String[] args) {  
  
        Double[] calificaciones = { 3.0, 7.8, 9.87, 5.68, 6.9, 9.0, 4.7 };  
        System.out.println("La mayor nota fue:" + mayorNota(calificaciones));  
        System.out.println("La menor nota fue:" + menorNota(calificaciones));  
        System.out.println("El promedio fue:" + promedio(calificaciones));  
  
    }  
  
    static Double mayorNota(Double[] calificaciones) {  
        Double mayor = -1.0;  
        for (int i = 0; i < calificaciones.length; i++) {  
            if (calificaciones[i] > mayor) {  
  
                mayor = calificaciones[i];  
  
            }  
        }  
        return mayor;  
    }  
  
    static Double menorNota(Double[] calificaciones) {  
  
        // Le definimos el menor valor como el primero del arreglo Double menor  
        // = calificaciones[0];  
        Double menor = calificaciones[0];  
        for (int i = 0; i < calificaciones.length; i++) {  
  
            if (calificaciones[i] < menor) {  
                menor = calificaciones[i];  
  
            }  
        }  
        return menor;  
    }  
}
```

```

    static Double promedio(Double[] calificaciones) {
        Double acumulado = 0.0;
        for (Double notaIndividual : calificaciones) {
            acumulado += notaIndividual;
        }
        Double promedio = acumulado / calificaciones.length;
        return promedio;
    }
}

```

Ejemplo 4:

Arreglo con valores iniciales.

```

import java.util.Scanner;

public class Ejemplo04 {
    public static void main(String[] args) {
        String meses[] = { "Enero", "Febrero", "Marzo", "Abril", "Mayo",
                           "Junio", "Julio", "Agosto", "Septiembre", "Octubre",
                           "Noviembre", "Diciembre" };
        for (int i = 0; i < meses.length; i++) {
            System.out.println(meses[i]);
        }
    }
}

```

Ejemplo05:

Llene un vector con una lista de números del 1 al 20 y que se despliegue este vector indicando a la derecha de cada uno si es divisible por 3 o no.

```

import java.util.Scanner;

public class Ejemplo05 {
    public static void main(String[] args) {
        int[] vector = new int[20];
        int n = 20;
        for (int i = 0; i < n; i++)
            // llenado del vector {
            vector[i] = i + 1;
    }
}

```

```
for (int i = 0; i < n; i++) {
    if (vector[i] % 3 == 0)
        System.out.println(vector[i] + " es Divisible por 3");
    else
        System.out.println(vector[i] + " No es divisible por 3");
}
```

Ejemplo 06:

```
import java.util.Scanner;

public class Ejemplo06 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int i;
        int pos = 0, neg = 0;
        // contadores de los números positivos y negativos
        int[] numeros = new int[10];
        // array que contendrá los números leídos por teclado
        double sumaPos = 0, sumaNeg = 0;
        // acumuladores para las sumas de positivos y negativos
        // lectura de datos y llenar el array
        System.out.println("Lectura de los elementos del array: ");
        for (i = 0; i < 10; i++) {
            System.out.print("numeros[" + i + "]= ");
            numeros[i] = sc.nextInt();
        }
        // recorrer el array para sumar por separado los números positivos // y
        // los negativos
        for (i = 0; i < 10; i++) {
            if (numeros[i] > 0) { // sumar positivos
                sumaPos += numeros[i];
                pos++;
            } else if (numeros[i] < 0) { // sumar negativos
                sumaNeg += numeros[i];
                neg++;
            }
        }
        // Calcular y mostrar las medias
        if (pos != 0) {
            System.out.println("Media de los valores positivos: " + sumaPos / pos);
        } else {
            System.out.println("Media de los valores negativos: " + sumaNeg / neg);
        }
    }
}
```

```

        System.out.println("No ha introducido numeros positivos");
    }
    if (neg != 0) {
        System.out.println("Media de los valores negativos: " + sumaNeg
                           / neg);
    } else {
        System.out.println("No ha introducido numeros negativos");
    }
}
}

```

V. Ejercicios complementarios.

1) Se necesita realizar un programa para nómina de empleados en donde debe almacenar los sueldos **en un arreglo**. Usted debe solicitar la cantidad de empleados.

Luego de registrar los empleados, imprimir las deducciones sobre un sueldo neto. Las deducciones que se aplican son iss (3%), afp (6.25%), renta (7.1%) y debe recorrerse el contenido del arreglo mostrando las deducciones para cada uno de ellos.

Ejemplo: Empleado 1: Sueldo: 450 Renta: 31.95 AFP: 28.125 Isss: 13.5 Total a pagar: 376.425 Empleado 2:

Sueldo: 600 Renta: 42.6 AFP: 37.5 Isss: 18 Total a pagar: 501.9 Etc....

2) Crear una calculadora que permita registrar 10 operaciones cualquiera que sean éstas. Crear una función para cada operación básica (suma, resta, multiplicación, división). El usuario por medio de un menú puede decidir qué operación debe realizar (crear un método para mostrar el menú). La operación y el resultado de la operación debe registrarse en un arreglo de tipo String y finalmente debe mostrar el histórico de tales operaciones recorriendo dicho arreglo.

