

PROGRAMACIÓN II

Trabajo Práctico N.º 2: Programación Estructurada

Estructuras Condicionales:

1. Verificación de Año Bisiesto.

Escribe un programa en Java que solicite al usuario un año y determine si es bisiesto. Un año es bisiesto si es divisible por 4, pero no por 100, salvo que sea divisible por 400.

Ejemplo de entrada/salida:

Ingrese un año: 2024

El año 2024 es bisiesto.

Ingrese un año: 1900

El año 1900 no es bisiesto.

```
package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int anio;

        //Se solicita un año
        System.out.print("Ingrese el año: ");
        anio = Integer.parseInt(sc.nextLine());

        //Estructura condicional para evaluar si el año es bisiesto
        if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
            System.out.println("El año " + anio + " es bisiesto");
        } else {
            System.out.println("El año " + anio + " no es bisiesto");
        }
    }
}
```

2. Determinar el Mayor de Tres Números.

Escribe un programa en Java que pida al usuario tres números enteros y determine cuál es el mayor.

Ejemplo de entrada/salida:

Ingrese el primer número: 8

Ingrese el segundo número: 12

Ingrese el tercer número: 5

El mayor es: 12

```
package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer2 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        //Declaracion de variables
        int num1, num2, num3;

        //Se solicita 3 numeros
        System.out.print("Ingrese el primer número: ");
        num1 = Integer.parseInt(sc.nextLine());

        System.out.print("Ingrese el segundo número: ");
        num2 = Integer.parseInt(sc.nextLine());

        System.out.print("Ingrese el tercer número: ");
        num3 = Integer.parseInt(sc.nextLine());

        //Estructura condicional para evaluar cual es el numero mayor y mostrarlo
        if ((num1 >= num2) && (num1 >= num3)) {
            System.out.println("El mayor es: " + num1);
        } else if ((num2 >= num1) && (num2 >= num3)) {
            System.out.println("El mayor es: " + num2);
        } else {
            System.out.println("El mayor es: " + num3);
        }
        sc.close();
    }
}
```

3. Clasificación de Edad.

Escribe un programa en Java que solicite al usuario su edad y clasifique su etapa de vida según la siguiente tabla:

Menor de 12 años: "Niño"

Entre 12 y 17 años: "Adolescente"

Entre 18 y 59 años: "Adulto"

60 años o más: "Adulto mayor"

Ejemplo de entrada/salida:

Ingrese su edad: 25

Eres un Adulto.

Ingrese su edad: 10

Eres un Niño.

```
package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer3 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int edad;

        //Solicitar la edad al usuario
        System.out.print("Ingrese su edad: ");
        edad = Integer.parseInt(sc.nextLine());

        //Estructura condicional para evaluar la edad y clasificar etapa de vida
        if (edad < 12) {
            System.out.println("Eres un Niño");
        } else if (edad < 18) {
            System.out.println("Eres un Adolescente");
        } else if (edad < 60) {
            System.out.println("Eres un Adulto");
        } else {
            System.out.println("Eres un Adulto mayor");
        }
        sc.close();
    }
}
```

4. Calculadora de Descuento según categoría.

Escribe un programa que solicite al usuario el precio de un producto y su categoría (A, B o C).

Luego, aplique los siguientes descuentos:

Categoría A: 10% de descuento

Categoría B: 15% de descuento

Categoría C: 20% de descuento

El programa debe mostrar el precio original, el descuento aplicado y el precio final

Ejemplo de entrada/salida:

Ingrese el precio del producto: 1000

Ingrese la categoría del producto (A, B o C): B

Descuento aplicado: 15%

Precio final: 850.0

```
package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer4 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Declaracion e inicializacion de variables
        double precioFinal = 0.0;
        int descuento = 0;

        //Solicitar el precio del producto
        System.out.print("Ingrese el precio del producto: ");
        double precioInicial = sc.nextDouble();

        //Solicitar la categoria del producto
        System.out.print("Ingrese la categoria del producto (A, B o C): ");
        char categoria = sc.next().charAt(0);

        //Estructura condicional para verificar categoria y descuento
        switch (categoria) {
            case 'A':
            case 'a':
                precioFinal = precioInicial * 0.9;
                descuento = 10;
                break;

            case 'B':
            case 'b':
                precioFinal = precioInicial * 0.85;
                descuento = 15;
                break;

            case 'C':
            case 'c':
                precioFinal = precioInicial * 0.8;
                descuento = 20;
                break;

            default:
                System.out.println("Categoria inválida");
        }

        //Mostrar el descuento aplicado y el precio final
        System.out.println("Descuento aplicado: " + descuento + "%");
        System.out.println("Precio final: " + precioFinal);
    }
}
```

Estructuras de Repetición:

5. Suma de Números Pares (while).

Escribe un programa que solicite números al usuario y sume solo los números pares. El ciclo debe continuar hasta que el usuario ingrese el número 0, momento en el que se debe mostrar la suma total de los pares ingresados.

Ejemplo de entrada/salida:

Ingrese un número (0 para terminar): 4

Ingrese un número (0 para terminar): 7

Ingrese un número (0 para terminar): 2

Ingrese un número (0 para terminar): 0

La suma de los números pares es: 6

```
package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer5 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        //Declaracion e inicializacion de variables
        int num, suma = 0;

        //Solicitar un numero al usuario hasta que ingrese 0
        System.out.print("Ingrese un número (0 para terminar): ");
        num = Integer.parseInt(sc.nextLine());

        //Estructura repetitiva para que el usuario ingrese varios numeros hasta que el numero sea 0
        while (num != 0) {
            if (num % 2 == 0) {
                suma += num;
            }
            System.out.print("Ingrese un número (0 para terminar): ");
            num = Integer.parseInt(sc.nextLine());
        }
        sc.close();
        System.out.println("La suma de los números pares es: " + suma);
    }
}
```

6. Contador de Positivos, Negativos y Ceros (for).

Escribe un programa que pida al usuario ingresar 10 números enteros y cuente cuántos son positivos, negativos y cuántos son ceros.

Ejemplo de entrada/salida:

Ingrese el número 1: -5

Ingrese el número 2: 3

Ingrese el número 3: 0

Ingrese el número 4: -1

Ingrese el número 5: 6

Ingrese el número 6: 0

Ingrese el número 7: 9

Ingrese el número 8: -3

Ingrese el número 9: 4

Ingrese el número 10: -8

Resultados:

Positivos: 4

Negativos: 4

Ceros: 2

```

package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer6 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        //Declaracion e inicializacion de variables
        int positivos = 0, negativos = 0, ceros = 0;

        //Estructura iterativa para solicitar el ingreso de 10 numeros enteros
        for (int i = 1; i < 11; i++) {
            System.out.print("Ingrese el numero " + i + ": ");
            int num = sc.nextInt();

            //Estructura condicional para evaluar que numeros son positivos, negativos o cero y acumularlos
            if (num > 0) {
                positivos++;
            } else if (num < 0) {
                negativos++;
            } else {
                ceros++;
            }
        }
        sc.close();

        //Mostrar los resultados
        System.out.println("\nResultados: ");
        System.out.println("Positivos: " + positivos);
        System.out.println("Negativos: " + negativos);
        System.out.println("Ceros: " + ceros);
    }
}

```

7. Validación de Nota entre 0 y 10 (do-while).

Escribe un programa que solicite al usuario una nota entre 0 y 10. Si el usuario ingresa un número fuera de este rango, debe seguir pidiéndole la nota hasta que ingrese un valor válido.

Ejemplo de entrada/salida:

Ingrese una nota (0-10): 15

Error: Nota inválida. Ingrese una nota entre 0 y 10.

Ingrese una nota (0-10): -2

Error: Nota inválida. Ingrese una nota entre 0 y 10.

Ingrese una nota (0-10): 8

Nota guardada correctamente.

```

package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer7 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int nota;

        do {
            // Solicitar una nota
            System.out.print("Ingrese una nota (0-10): ");
            nota = Integer.parseInt(sc.nextLine());

            // Verificar si la nota es invalida
            if (nota < 0 || nota > 10) {
                System.out.println("Error: Nota inválida. Ingrese una nota entre 0 y 10.");
            }

        } while (nota < 0 || nota > 10);
        //Confirmar nota guardada correctamente
        System.out.println("Nota guardada correctamente ");
        sc.close();
    }
}

```

8. Cálculo del Precio Final con impuesto y descuento.

Crea un método `calcularPrecioFinal(double impuesto, double descuento)` que calcule el precio final de un producto en un e-commerce. La fórmula es:

PrecioFinal = PrecioBase + (PrecioBase×Impuesto) – (PrecioBase×Descuento)

PrecioFinal = PrecioBase + (PrecioBase \times Impuesto) - (PrecioBase \times Descuento)

Desde `main()`, solicita el precio base del producto, el porcentaje de impuesto y el porcentaje de descuento, llama al método y muestra el precio final.

Ejemplo de entrada/salida:

Ingrese el precio base del producto: 100

Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%): 10

Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%): 5

El precio final del producto es: 105.0

```
package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer8 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Solicitar el precio base
        System.out.print("Ingrese el precio base del producto: ");
        double precioBase = sc.nextDouble();

        // Solicitar el porcentaje de impuesto
        System.out.print("Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%): ");
        double impuestoPorcentaje = sc.nextDouble();

        // Solicitar el porcentaje de descuento
        System.out.print("Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%): ");
        double descuentoPorcentaje = sc.nextDouble();

        // Convertir porcentajes a decimales
        double impuesto = impuestoPorcentaje / 100.0;
        double descuento = descuentoPorcentaje / 100.0;

        // Llamar al método y obtener el precio final
        double precioFinal = calcularPrecioFinal(precioBase, impuesto, descuento);

        // Mostrar el resultado
        System.out.println("El precio final del producto es: " + precioFinal);

        sc.close();
    }

    // Metodo para calcular el precio final
    public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento) {
        return precioBase + (precioBase * impuesto) - (precioBase * descuento);
    }
}
```

9. Composición de funciones para calcular costo de envío y total de compra.

a. **calcularCostoEnvio(double peso, String zona)**: Calcula el costo de envío basado en la zona de envío (Nacional o Internacional) y el peso del paquete.

Nacional: \$5 por kg

Internacional: \$10 por kg

b. **calcularTotalCompra(double precioProducto, double costoEnvio)**: Usa **calcularCostoEnvio** para sumar el costo del producto con el costo de envío.

Desde **main()**, solicita el peso del paquete, la zona de envío y el precio del producto.

Luego, muestra el total a pagar.

Ejemplo de entrada/salida:

Ingrese el precio del producto: 50

Ingrese el peso del paquete en kg: 2

Ingrese la zona de envío (Nacional/Internacional): Nacional

El costo de envío es: 10.0

El total a pagar es: 60.0

```
package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer9 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Solicitar el precio
        System.out.print("Ingrese el precio del producto: ");
        double precioProducto = sc.nextDouble();

        // Solicitar el peso del paquete
        System.out.print("Ingrese el peso del paquete en Kg: ");
        double pesoPaquete = sc.nextDouble();

        // Solicitar zona de envio
        System.out.print("Ingrese la zona de envío (Nacional/Internacional): ");
        String zonaEnvio = sc.next();

        //Verificacion de que la zona ingresada es válida
        if (zonaEnvio.equalsIgnoreCase("Nacional") || zonaEnvio.equalsIgnoreCase("Internacional")) {

            //se llama al metodo para calcular el costo del envio y el total de la compra
            double costoEnvio = calcularCostoEnvio(pesoPaquete, zonaEnvio);
            double totalPagar = calcularTotalCompra(precioProducto, costoEnvio);

            //Se muestra el costo del envio y el total de la compra
            System.out.println("El costo del envio es: " + costoEnvio);
            System.out.println("El total a pagar es: " + totalPagar);
        } else {
            System.out.println("Zona no válida."); //Zona ingresada inválida
        }
        sc.close();
    }

    //Metodo para calcular el costo del envio
    public static double calcularCostoEnvio(double peso, String zona) {

        if (zona.equalsIgnoreCase("Nacional")) {
            return peso * 5;
        } else {
            return peso * 10;
        }
    }

    //Metodo para calcular el precio total
    public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
        return precioProducto + costoEnvio;
    }
}
```


10. Actualización de stock a partir de venta y recepción de productos.

Crea un método **actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida)**, que calcule el nuevo stock después de una venta y recepción de productos:

NuevoStock = StockActual – CantidadVendida + CantidadRecibida

NuevoStock = CantidadVendida + CantidadRecibida

Desde **main()**, solicita al usuario el stock actual, la cantidad vendida y la cantidad recibida, y muestra el stock actualizado.

Ejemplo de entrada/salida:

Ingrese el stock actual del producto: 50

Ingrese la cantidad vendida: 20

Ingrese la cantidad recibida: 30

El nuevo stock del producto es: 60

```
package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer10 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Solicitar el stock actual
        System.out.print("Ingrese el stock actual del producto: ");
        int stockActual = Integer.parseInt(sc.nextLine());

        // Solicitar cantidad vendida
        System.out.print("Ingrese la cantidad vendida: ");
        int cantidadVendida = Integer.parseInt(sc.nextLine());

        // Solicitar cantidad recibida
        System.out.print("Ingrese la cantidad recibida: ");
        int cantidadRecibida = Integer.parseInt(sc.nextLine());

        int stockProducto = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);
        System.out.println("El nuevo stock del producto es: " + stockProducto);

        sc.close();
    }

    //Metodo para actualizar el stock
    public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
        return stockActual - cantidadVendida + cantidadRecibida;
    }
}
```

11. Cálculo de descuento especial usando variable global.

Declara una variable global **Ejemplo de entrada/salida:** = 0.10. Luego, crea un método **calcularDescuentoEspecial(double precio)** que use la variable global para calcular el descuento especial del 10%.

Dentro del método, declara una variable local **descuentoAplicado**, almacena el valor del descuento y muestra el precio final con descuento.

Ejemplo de entrada/salida:

Ingrese el precio del producto: 200

El descuento especial aplicado es: 20.0

El precio final con descuento es: 180.0

```
package tp.pkg2.prog.estructurada;
import java.util.Scanner;

public class ejer11 {

    public static final double DESCUENTO_GLOBAL = 0.10; //Declaracion de la variable global

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Solicitar el precio del producto
        System.out.print("Ingrese el precio del producto: ");
        double precioProducto = sc.nextDouble();

        //Calcular y mostrar el descuento y el precio final con el descuento aplicado
        calcularDescuentoEspecial(precioProducto);

        sc.close();
    }

    //Metodo para calcular el descuento aplicado y el precio final
    public static void calcularDescuentoEspecial(double precio) {
        double descuentoAplicado = precio * DESCUENTO_GLOBAL;
        double precioFinal = precio - descuentoAplicado;

        System.out.println("El descuento especial aplicado es: " + descuentoAplicado);
        System.out.println("El precio final con descuento es: " + precioFinal);
    }
}
```

Arrays y Recursividad:

12. Modificación de un array de precios y visualización de resultados.

Crea un programa que:

- Declare e inicialice un array con los precios de algunos productos.
- Muestre los valores originales de los precios.
- Modifique el precio de un producto específico.
- Muestre los valores modificados.

Salida esperada:

Precios originales:

Precio: \$199.99

Precio: \$299.5

Precio: \$149.75

Precio: \$399.0

Precio: \$89.99

Precios modificados:

Precio: \$199.99

Precio: \$299.5

Precio: \$129.99

Precio: \$399.0

Precio: \$89.99

Conceptos Clave Aplicados:

- ✓ Uso de arrays (double[]) para almacenar valores.
- ✓ Recorrido del array con for-each para mostrar valores.
- ✓ Modificación de un valor en un array mediante un índice.
- ✓ Reimpresión del array después de la modificación.

```
package tp.pkg2.prog.estructurada;

public class ejer12 {

    public static void main(String[] args) {

        //Se declara e inicializa la array con los precios
        double[] preciosOriginales = {199.99, 299.5, 149.75, 399.0, 89.99};

        //Se muestra todos los precios sin modificar de la array
        System.out.println("Precios originales: ");
        mostrarPrecios(preciosOriginales);

        //Se modifica el 3er precio de la array
        preciosOriginales[2] = 129.99;

        //Se muestra todos los precios modificados de la array
        System.out.println("Precios modificados: ");
        mostrarPrecios(preciosOriginales);

    }

    //Metodo iterativo para mostrar todos los precios de la array
    public static void mostrarPrecios(double[] precios) {
        for (double precio : precios) {
            System.out.println("Precio: $" + precio);
        }
    }

}
```

13. Impresión recursiva de arrays antes y después de modificar un elemento.

Crea un programa que:

- Declare e inicialice un array con los precios de algunos productos.
- Use una función recursiva para mostrar los precios originales.
- Modifique el precio de un producto específico.
- Use otra función recursiva para mostrar los valores modificados.

Salida esperada:

Precios originales:

Precio: \$199.99

Precio: \$299.5

Precio: \$149.75

Precio: \$399.0

Precio: \$89.99

Precios modificados:

Precio: \$199.99

Precio: \$299.5

Precio: \$129.99

Precio: \$399.0

Precio: \$89.99

Conceptos Clave Aplicados:

- ✓ Uso de arrays (double[]) para almacenar valores.
- ✓ Recorrido del array con una función recursiva en lugar de un bucle.
- ✓ Modificación de un valor en un array mediante un índice.
- ✓ Uso de un índice como parámetro en la recursión para recorrer el array.

```
package tp.pkg2.prog.estructurada;

public class ejer13 {

    public static void main(String[] args) {

        //Se declara e inicializa la array con los precios
        double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};

        //Se muestra todos los precios sin modificar de la array
        System.out.println("Precios originales: ");
        mostrarPrecios(precios, 0);

        //Se modifica el 3er precio de la array
        precios[2] = 129.99;

        //Se muestra todos los precios modificados de la array
        System.out.println("Precios modificados: ");
        mostrarPrecios(precios, 0);

    }

    //Metodo recursivo para mostrar todos los precios de la array
    public static void mostrarPrecios(double[] precios, int indice) {
        if (indice >= precios.length) {
            return;
        }
        System.out.println("precio: $" + precios[indice]);
        mostrarPrecios(precios, indice + 1);
    }

}
```