

PROGRAMACIÓN II.

Modulo 2: Programación Estructurada.

Trabajo Práctico 2: Programación Estructurada.

Alumno: LEPKA AGUSTIN

Comisión: 13

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.

Respuesta:

//Se declara la variable scan de tipo Scanner para poder leer los datos de entrada

```
Scanner scan = new Scanner(System.in);
```

```
/*  
  Estructuras condicionales:  
  Ejercicio 1: Verificación de año bisiesto  
*/
```

```
//Se pide al usuario que ingrese un año a verificar si es bisiesto  
System.out.print("Ingresa un año: ");
```

```
//Se lee el dato ingresado por el usuario  
int año = scan.nextInt();
```

```
//Mediante estructura condicional IF-ELSE se determina si el año  
ingresado es o no bisiesto
```

```
if ((año % 4 == 0 && año % 100 != 0) || (año % 400 == 0))  
{  
    System.out.println("El año " + año + " es bisiesto.");  
}  
else  
{  
    System.out.println("El año " + año + " 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

Respuesta:

//Se declara la variable scan de tipo Scanner para poder leer los datos de entrada

```
Scanner scan = new Scanner(System.in);
```

```
/*  
  Estructuras condicionales:  
  Ejercicio 2: Determinar el Mayor de Tres Números.  
*/
```

//Se pide el primer numero al usuario

```
System.out.print("Ingresa el primer número: ");
```

```
int num1 = scan.nextInt();
```

//Se pide el primer numero al usuario

```
System.out.print("Ingresa el segundo número: ");
```

```
int num2 = scan.nextInt();
```

//Se pide el primer numero al usuario

```
System.out.print("Ingresa el tercer número: ");
```

```
int num3 = scan.nextInt();
```

//Se calcula cual es el mayor numero ingresado llamando al metodo

Math.max()

```
int mayor = Math.max(num1, Math.max(num2, num3));
```

```
System.out.println("El mayor es: " + mayor);
```

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.

Respuesta:

//Se declara la variable scan de tipo Scanner para poder leer los datos de entrada

```
Scanner scan = new Scanner(System.in);
```

```
/*  
  Estructuras condicionales:  
  Ejercicio 3: Clasificación de Edad.  
*/
```

//Se le indica al usuario que ingrese su edad para clasificarla

```
System.out.print("Ingrese su edad: ");
```

```
int edad = scan.nextInt();
```

//Mediante estructura IF-ELSE se clasifica la edad introducida

```
if (edad < 12)
```

```
{  
    System.out.println("Eres un Niño.");  
}
```

```
else if (edad <= 17)
```

```
{  
    System.out.println("Eres un Adolescente.");  
}
```

```
else if (edad <= 59)
```

```
{  
    System.out.println("Eres un Adulto.");  
}
```

```
else
```

```
{  
    System.out.println("Eres un Adulto mayor.");  
}
```

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

Respuesta:

//Se declara la variable scan de tipo Scanner para poder leer los datos de entrada

```
Scanner scan = new Scanner(System.in);
```

```
/*
```

```
 Estructuras condicionales:
```

```
 Ejercicio 4: Calculadora de Descuento según categoría.
```

```
*/
```

//Se le solicita al usuario que ingrese un precio de un producto

```
System.out.print("Ingrese el precio del producto: ");
```

```
double precio = scan.nextDouble();
```

//Se le solicita al usuario que ingrese la categoría del producto

```
System.out.print("Ingrese la categoría del producto (A, B o C): ");
```

```
char categoria = scan.next().toUpperCase().charAt(0);
```

//Se declara e inicia la variable que contendrá el descuento según categoría

```
double descuento = 0;
```

//Con la estructura condicional SWITCH se establece el descuento por categoría

```
switch (categoria)
```

```
{
```

```
    case 'A': descuento = 10;
```

```
    break;
```

```
    case 'B': descuento = 15;
```

```
    break;
```

```

        case 'C': descuento = 20;
        break;
        default: System.out.println("Categoría inválida. No se aplica
descuento.");
    }
    //Se declara e inicia la variable en la que se calcula el precio final del
producto
    double precioFinal = precio - (precio * descuento / 100);
    //Se imprimien los resultados por consola
    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

Respuesta:

```

    //Se declara la variable scan de tipo Scanner para poder leer los datos de
entrada
    Scanner scan = new Scanner(System.in);

    /*
    Estructuras de Repetición:
    Ejercicio 5: Suma de Número Pares (while).
    */

    //se declara e inicializa la variable sumaPares que servirá para acumular y
sumar los numeros ingresados
    int sumaPares = 0;
    //se declara la variable num que es la que va a almacenar el dato
    ingresado por el usuario
    int num;

    //inicio del ciclo do-while
    do
    {

```

```

//se le pide al usuario que ingrese un numero
System.out.print("Ingrese un número (0 para terminar): ");
num = scan.nextInt();

//mediante estructura if-else se establece que si el numero es par se
suma y si es 0 se termina
if (num % 2 == 0 && num != 0)
{
    sumaPares += num;
}
}
//el do-while tiene como condicion seguir ejecutandose siempre que el
numero ingresado sea distinto de 0
while (num != 0);
System.out.println("La suma de los números pares es: " + sumaPares);

```

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

```

Respuesta:

```

//Se declara la variable scan de tipo Scanner para poder leer los datos de
entrada
Scanner scan = new Scanner(System.in);

/*
Estructuras de Repetición:
Ejercicio 6: Contador de Positivos, Negativos y Ceros (for)
*/

```

```

//se declaran e inicializan las variables que van a contar la cantidad de
tipos de numeros
int positivos = 0, negativos = 0, ceros = 0;

//mediante estructura ciclo for se va
for (int orden = 1; orden <= 10; orden++)
{
    //se le solicita al usuario que ingrese un numero y se le indica que orden
    System.out.print("Ingrese el número " + orden + ": ");
    //en la variable n se almacena lo introducido por el usuario
    int n = scan.nextInt();
    //mediante estructura if-else se determina que tipo de numero es y lo
    cuenta en la variable n
    if (n > 0)
    {
        positivos++;
    }
    else if (n < 0)
    {
        negativos++;
    }

    else
    {
        ceros++;
    }
}

//se imprimen los resultados
System.out.println("Resultados:");
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.

```

Respuesta:

```
//Se declara la variable scan de tipo Scanner para poder leer los datos de
entrada
Scanner scan = new Scanner(System.in);

/*
Estructuras de Repetición:
Ejercicio 7: Validación de Nota entre 0 y 10 (do-while).
*/

//se declara la variable nota que almacena el input del user
int nota;
//mediante estructura ciclo do-while se solicita el input hasta que sea
correcto
do
{
    //se le pide al usuario que ingrese una nota valida
    System.out.print("Ingrese una nota (0-10): ");
    nota = scan.nextInt();
    //mediante estructura condicional if-else se establece el rango valido
para el input
    if (nota < 0 || nota > 10)
    {
        System.out.println("Error: Nota inválida. Ingrese una nota entre 0 y
10.");
    }

}
//para el do-while se establece que si la nota esta entre los valores validos
la nota se guarda correctamente, sino vuelve a empezar el ciclo
while (nota < 0 || nota > 10);
System.out.println("Nota guardada correctamente.");
```

Funciones:

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:

$$\text{PrecioFinal} = \text{PrecioBase} + (\text{PrecioBase} \times \text{Impuesto}) - (\text{PrecioBase} \times \text{Descuento})$$
$$\text{PrecioFinal} = \text{PrecioBase} + (\text{PrecioBase} \times \text{Impuesto}) - (\text{PrecioBase} \times \text{Descuento})$$

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

Respuesta:

```
public class Tp2_programacion2Codetest {

    //se declara la función(metodo) para luego llamarla y calcular el precio final

    static double calcularPrecioFinal(double precioBase, double impuesto,
double descuento) {
        //se declara e inicializa la variable precioFinal que calcula el impuesto y
descuento
        double precioFinal = precioBase + (precioBase * impuesto / 100) -
(precioBase * descuento / 100);
        //la funcion devolvera un valor
        return precioFinal;
    }
    public static void main(String[] args) {

        //Se declara la variable scan de tipo Scanner para poder leer los datos de
entrada
        Scanner scan = new Scanner(System.in);

        /*
        Funciones:
        Ejercicio 8: Cálculo del Precio Final con impuesto y descuento.
        */
        //se le solicita al usuario que ingrese el precio del producto
        System.out.print("Ingrese el precio base del producto: ");
        double precioBase = scan.nextDouble();
        //se le solicita al usuario que ingrese el porcentaje de impuesto
        System.out.print("Ingrese el impuesto en porcentaje (Ejemplo: 10 para
10%): ");
        double impuesto = scan.nextDouble();
        //se le solicita al usuario que ingrese el precio del producto
        System.out.print("Ingrese el descuento en porcentaje (Ejemplo: 5 para
5%): ");
        double desc = scan.nextDouble();
        //se llama al metodo(funcion) y se le indica los argumentos (precioBase,
impuesto, desc) para el calculo
        double precioFinal2 = calcularPrecioFinal(precioBase, impuesto, desc);
        //se imprime el resultado
        System.out.println("El precio final del producto es: " + precioFinal2);
    }
}
```

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

Respuesta:

```
public class Tp2_programacion2Codetest {  
  
    //se declara la funcion(metodo) que va a calcular el costo de envio al llamarla  
    static double calcularCostoEnvio(double peso, String zona) {  
        //mediante estructura if-else se establece el tipo de zona y valor  
        if (zona.equalsIgnoreCase("Nacional"))  
        {  
            return 5 * peso;  
        }  
        else if (zona.equalsIgnoreCase("Internacional"))  
        {  
            return 10 * peso;  
        }  
        else  
        {  
            System.out.println("Zona no reconocida. Se aplicará costo 0.");  
            return 0;  
        }  
    }  
  
    //se declara la funcion(metodo) que va a calcular el total de la compra al  
    llamarla  
    static double calcularTotalCompra(double precioProducto, double peso,  
    String zona) {  
        //se declara la variable que va a calcular el costo de envio segun zona  
        double costoEnvio = calcularCostoEnvio(peso, zona);
```

```

        //imprime el resultado
        System.out.println("El costo de envío es: " + costoEnvio);
        //devuelve el costo total (costo de envío y el precio del producto)
        return precioProducto + costoEnvio;
    }

    public static void main(String[] args) {

        //Se declara la variable scan de tipo Scanner para poder leer los datos de
        entrada
        Scanner scan = new Scanner(System.in);

        /*
        Funciones:
        Ejercicio 9: Composición de funciones para calcular costo de envío y total
        */

        //se le pide al usuario el precio del producto
        System.out.print("Ingrese el precio del producto: ");
        double precioProducto = scan.nextDouble();
        //se le pide al usuario el peso del producto
        System.out.print("Ingrese el peso del paquete en kg: ");
        double peso = scan.nextDouble();
        //se le pide al usuario la zona de envío
        System.out.print("Ingrese la zona de envío (Nacional/Internacional): ");
        //se llama al metodo .nextLine() porque se necesita limpiar el salto de linea
        que queda "\n" en el buffer para que pueda leerse un input tipo string a
        continuacion
        scan.nextLine();

        String zona = scan.nextLine();
        //se calcula el total de la compra llamando a la funcion(metodo) declarada
        al principio de este codigo
        double totalCompra = calcularTotalCompra(precioProducto, peso, zona);
        //se imprime el total a pagar
        System.out.println("El total a pagar es: " + totalCompra);
    }
}

```

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

Respuesta:

```
public class Tp2_programacion2Codetest {
    //se declara la funcion(metodo) para calcular stock despues de venta y
    recepcion de productos
    public static int actualizarStock(int stockActual, int cantidadVendida, int
    cantidadRecibida) {
        //devolvera un valor total
        return stockActual - cantidadVendida + cantidadRecibida;
    }

    public static void main(String[] args) {

        //Se declara la variable scan de tipo Scanner para poder leer los datos de
        entrada
        Scanner scan = new Scanner(System.in);

        /*
        Funciones:
        Ejercicio 10: Actualización de stock a partir de venta y recepción de
        productos
        */

        //se le pide al usuario que ingrese el stock actual del producto
        System.out.print("Ingrese el stock actual del producto: ");
        int stockActual = scan.nextInt();
        //se le pide al usuario que ingrese la cantidad vendida del producto
        System.out.print("Ingrese la cantidad vendida: ");
        int cantVendida = scan.nextInt();
        //se le pide al usuario que ingrese la cantidad recibida del producto
        System.out.print("Ingrese la cantidad recibida: ");
        int cantRecibida = scan.nextInt();
        //se llama al metodo que calculara el nuevo stock usando como argumento
        los datos ingresados anteriormente
        int nuevoStock = actualizarStock(stockActual, cantVendida, cantRecibida);
        //se imprime el resultado por consola
        System.out.println("El nuevo stock del producto es: " + nuevoStock);
    }
}
```

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

Respuesta:

```
class ClaseParaVariableGlobal {
    public static double descuentoEspecial = 0.10;
}

public class Tp2_programacion2Codetest {

    public static double calcularDescuentoEspecial(double precio){

        double descuentoAplicado = precio *
ClaseParaVariableGlobal.descuentoEspecial;
        double precioFinal = precio - descuentoAplicado;

        return precioFinal;

    }

    public static void main(String[] args) {

        //Se declara la variable scan de tipo Scanner para poder leer los datos de
        entrada
        Scanner scan = new Scanner(System.in);

        /*
        Funciones:
        Ejercicio 11: Cálculo de descuento especial usando variable global.
        */

        //se le pide al usuario que ingrese el valor del producto
        System.out.print("Ingrese el precio del producto: ");
        double precio = scan.nextDouble();

        double precioFinal = calcularDescuentoEspecial(precio);
```

```

        //se imprimen los valores de descuento y el precio final
        System.out.println("El descuento especial aplicado es: " +
ClaseParaVariableGlobal.descuentoEspecial);
        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

Respuesta:

```

        //Se declara la variable scan de tipo Scanner para poder leer los datos
        de entrada
        Scanner scan = new Scanner(System.in);

        /*
        Arrays y Recursividad:
        Ejercicio 12: Modificación de un array de precios y visualización de
        resultados.
        */

        //se crea un array de precios que son de tipo double
        double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};

        //se imprimen los precios originales
        System.out.println("Precios originales:");
    }
}

```

```

//mediante un bucle for se recorre todo el arreglo y se imprimen los valores
for (double indice : precios) {
    System.out.println("Precio: $" + indice);
}

//Se selecciona en el array lugar que ocupa el precio a modificar
precios[2] = 129.99;
//se imprimen los precios modificados
System.out.println("Precios modificados:");
//mediante un bucle for se recorre todo el arreglo y se imprimen los valores
modificados
for (double indice : precios)
{
    System.out.println("Precio: $" + indice);
}

```

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

Respuesta:

```

public class Tp2_programacion2Codetest {
    //se crea la funcion(metodo) para mostrar un array de forma recursiva
    public static void mostrarPreciosRecursivo(double[] precios, int indice) {

        //mediante estructura condicional if se establece que si el indice es menor
        a la longitud del array va a imprimir los precios.
        if (indice < precios.length)
        {
            System.out.println("Precio: $" + precios[indice]);
            //llama al mismo metodo de manera recursiva

```

```

        mostrarPreciosRecursivo(precios, indice + 1);
    }
}

public static void main(String[] args) {

    //Se declara la variable scan de tipo Scanner para poder leer los datos de
    entrada
    Scanner scan = new Scanner(System.in);

    /*
    Funciones:
    Ejercicio 13: Impresión recursiva de arrays antes y después de modificar
    un elemento.
    */

    // se establece el array de precios
    double[] preciosRec = {199.99, 299.5, 149.75, 399.0, 89.99};
    System.out.println("Precios originales (recursivo):");
    mostrarPreciosRecursivo(preciosRec, 0);

    // se modifica el array en el precio necesario
    preciosRec[2] = 129.99;
    System.out.println("Precios modificados (recursivo):");
    mostrarPreciosRecursivo(preciosRec, 0);

}

}

```