

PROGRAMACIÓN II

Trabajo Práctico 2: Programación Estructurada

Repositorio Github: <https://github.com/AguP10/UTN-TUPaD-P2.git>

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.

```
13 public class Tp2 {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         Scanner input = new Scanner(System.in);
20         int anio;
21         System.out.println("Ingrese un año: ");
22         anio = Integer.parseInt(input.nextLine());
23
24         if (anio % 400 == 0) {
25             System.out.println("El año " + anio + " es bisiesto");
26         } else if (anio % 100 == 0) {
27             System.out.println("El año " + anio + " no es bisiesto");
28         } else if (anio % 4 == 0) {
29             System.out.println("El año " + anio + " es bisiesto");
30         } else {
31             System.out.println("El año " + anio + " no es bisiesto");
32         }
33     }
34 }
35
```

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.

```
13 public class Tp2 {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         Scanner input = new Scanner(System.in);
20         int num1, num2, num3, max;
21
22         System.out.print("Ingrese el numero 1: ");
23         num1 = Integer.parseInt(input.nextLine());
24
25         max = num1;
26
27         System.out.print("Ingrese el numero 2: ");
28         num2 = Integer.parseInt(input.nextLine());
29         if (num2 > max) {
30             max = num2;
31         }
32
33         System.out.print("Ingrese el numero 3: ");
34         num3 = Integer.parseInt(input.nextLine());
35         if (num3 > max) {
36             max = num3;
37         }
38         System.out.println("El mayor es: " + max);
39     }
40 }
```

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"

```
13 public class Tp2 {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         Scanner input = new Scanner(System.in);
20         int edad;
21
22         System.out.print("Ingrese su edad : ");
23         edad = Integer.parseInt(input.nextLine());
24
25         if (edad < 12){
26             System.out.println("Eres un niño");
27         }else if (edad >= 12 && edad < 18){
28             System.out.println("Eres un Adolescente");
29         }else if (edad >= 18 && edad < 60){
30             System.out.println("Eres un Adulto");
31         }else if (edad >= 60){
32             System.out.println("Eres un Adulto mayor");
33         }
34     }
35 }
```

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.

```
18 public static void main(String[] args) {
19     Scanner input = new Scanner(System.in);
20     int precio;
21     double descuento, precioFinal;
22     char cat;
23     System.out.println("Ingrese el precio del producto: ");
24     precio = Integer.parseInt(input.nextLine());
25     System.out.println("Ingrese el la categoria (A, B o C): ");
26     cat = input.nextLine().charAt(0);
27     switch (cat) {
28         case 'A':
29             descuento = precio * 0.10;
30             precioFinal = precio - descuento;
31             System.out.println("El precio es: " + precio);
32             System.out.println("Su descuento es de: " + descuento);
33             System.out.println("El precio con descuento es: " + precioFinal);
34             break;
35         case 'B':
36             descuento = precio * 0.15;
37             precioFinal = precio - descuento;
38             System.out.println("El precio es: " + precio);
39             System.out.println("Su descuento es de: " + descuento);
40             System.out.println("El precio con descuento es: " + precioFinal);
41             break;
42         case 'C':
43             descuento = precio * 0.20;
44             precioFinal = precio - descuento;
45             System.out.println("El precio es: " + precio);
46             System.out.println("Su descuento es de: " + descuento);
47             System.out.println("El precio con descuento es: " + precioFinal);
48             break;
49     }
50 }
51
52
53 }
```

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.

```
13 public class Tp2 {
14
15
16     public static void main(String[] args) {
17         int numero, suma;
18         Scanner input = new Scanner(System.in);
19         numero = Integer.parseInt(input.nextLine());
20         suma = 0;
21
22         while (numero != 0) {
23             if (numero % 2 == 0) {
24                 suma = suma + numero;
25                 numero = Integer.parseInt(input.nextLine());
26             } else if (numero % 2 != 0) {
27                 numero = Integer.parseInt(input.nextLine());
28             }
29         }
30         System.out.println("La suma de los numeros pares es: " + suma );
31     }
32 }
```

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 zeros.

```
13 public class Tp2 {
14
15
16     public static void main(String[] args) {
17         int numero, positivos, negativos, ceros;
18         Scanner input = new Scanner(System.in);
19         positivos = 0;
20         negativos = 0;
21         ceros = 0;
22         for (int i = 1; i <= 10; i++){
23             System.out.println("Ingrese un numero entero: ");
24             numero = Integer.parseInt(input.nextLine());
25             if(numero > 0){
26                 positivos++;
27             }else if (numero < 0){
28                 negativos++;
29             }else{
30                 ceros++;
31             }
32         }
33         System.out.println("Positivos: " + positivos);
34         System.out.println("Negativos: " + negativos);
35         System.out.println("Ceros: " + ceros);
36     }
37
38 }
39
```

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.

```
5  package tp2;
6
7  import java.util.Scanner;
8
9  /**
10   *
11   * @author agu-p
12   */
13  public class Tp2 {
14
15      public static void main(String[] args) {
16          int nota;
17          Scanner input = new Scanner(System.in);
18
19          do {
20              System.out.println("Ingrese una nota: ");
21              nota = Integer.parseInt(input.nextLine());
22          } while (nota < 0 || nota > 10);
23
24      }
25
26  }
```

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:

$\text{PrecioFinal} = \text{PrecioBase} * (1 + \text{Impuesto} - \text{Descuento});$ 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.

```
9 public class Tp2 {
10     static double precioBase;
11
12     public static void main(String[] args) {
13         Scanner input = new Scanner(System.in);
14         double impuesto, descuento, precioFinal;
15
16         System.out.println("Ingrese el precio base del producto: ");
17         precioBase = Double.parseDouble(input.nextLine());
18         System.out.println("Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%):");
19         impuesto = Double.parseDouble(input.nextLine());
20         System.out.println("Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%):");
21         descuento = Double.parseDouble(input.nextLine());
22
23         precioFinal = calcularPrecioFinal(impuesto, descuento);
24         System.out.println("precio final: " + precioFinal);
25     }
26
27     static double calcularPrecioFinal(double impuesto, double descuento){
28         impuesto = impuesto / 100;
29         descuento = descuento / 100;
30         return precioBase * (1 + impuesto - descuento);
31     }
32 }
```


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.

```
9 public class Tp2 {
10
11     static double nacional = 5;
12     static double internacional = 10;
13     public static void main(String[] args) {
14         Scanner input = new Scanner(System.in);
15         double peso, total, precioProducto;
16         String zona;
17
18         System.out.println("Ingrese el precio del producto: ");
19         precioProducto = Double.parseDouble(input.nextLine());
20         System.out.println("Ingrese el peso del paquete en kg:");
21         peso = Double.parseDouble(input.nextLine());
22         System.out.println("Ingrese la zona de envío (Nacional/Internacional): ");
23         zona = input.nextLine();
24
25         total = calcularTotalCompra(precioProducto, calcularCostoEnvio(peso, zona));
26         System.out.println("El total es: " + total);
27     }
28
29     static double calcularCostoEnvio (double peso, String zona){
30         if (zona.equalsIgnoreCase("nacional")) {
31             return peso * nacional;
32         }else if (zona.equalsIgnoreCase("internacional")) {
33             return peso * internacional;
34         }else {
35             System.out.println("Zona inválida. Se cobrará 0 de envío.");
36             return 0;
37         }
38     }
39     static double calcularTotalCompra(double precioProducto, double costoEnvio ){
40         return precioProducto + costoEnvio;
41     }
42 }
43
```

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.

```
1  package tp2;
2
3
4  import java.util.Scanner;
5
6  public class Tp2 {
7
8      public static void main(String[] args) {
9          Scanner input = new Scanner(System.in);
10
11          int stockActual, cantidadVendida, cantidadRecibida, nuevoStock;
12
13          System.out.print("Ingrese el stock actual del producto: ");
14          stockActual = Integer.parseInt(input.nextLine());
15
16          System.out.print("Ingrese la cantidad vendida: ");
17          cantidadVendida = Integer.parseInt(input.nextLine());
18
19          System.out.print("Ingrese la cantidad recibida: ");
20          cantidadRecibida = Integer.parseInt(input.nextLine());
21
22          nuevoStock = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);
23
24          System.out.println("El nuevo stock del producto es: " + nuevoStock);
25
26      }
27
28
29      static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
30          return stockActual - cantidadVendida + cantidadRecibida;
31      }
32  }
```

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

```
1  package tp2;
2
3  import java.util.Scanner;
4
5  public class Tp2 {
6
7      static double descuentoEspecial = 0.10;
8
9      public static void main(String[] args) {
10         Scanner input = new Scanner(System.in);
11
12         double precio;
13
14         System.out.print("Ingrese el precio del producto: ");
15         precio = Double.parseDouble(input.nextLine());
16
17         calcularDescuentoEspecial(precio);
18     }
19
20
21     static void calcularDescuentoEspecial(double precio) {
22
23         double descuentoAplicado = precio * descuentoEspecial;
24         double precioFinal = precio - descuentoAplicado;
25
26         System.out.println("El descuento especial aplicado es: " + descuentoAplicado);
27         System.out.println("El precio final con descuento es: " + precioFinal);
28     }
29 }
```

Arrays y Recursividad:

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

Crea un programa que:

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

```
1  package tp2;
2
3  public class Tp2 {
4
5      public static void main(String[] args) {
6
7          double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
8
9          System.out.println("Precios originales:");
10         mostrarPrecios(precios);
11
12         precios[2] = 129.99;
13
14         System.out.println("Precios modificados:");
15         mostrarPrecios(precios);
16     }
17
18     static void mostrarPrecios(double[] array) {
19         for (double precio : array) {
20             System.out.println("Precio: $" + precio);
21         }
22     }
23 }
```

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.

```
1  package tp2;
2
3  public class Tp2 {
4
5      public static void main(String[] args) {
6          double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
7          System.out.println("Precios originales:");
8          mostrarPreciosRecursivo(precios, 0);
9
10         precios[2] = 129.99;
11
12         System.out.println("Precios modificados:");
13         mostrarPreciosRecursivo(precios, 0);
14     }
15
16     static void mostrarPreciosRecursivo(double[] array, int indice) {
17
18         if (indice >= array.length) {
19             return;
20         }
21         System.out.println("Precio: $" + array[indice]);
22         mostrarPreciosRecursivo(array, indice + 1);
23     }
24 }
```