**TRABAJO PRÁCTICO Nº2º: PROGRAMACIÓN ESTRUCTURADA**

**CHRISTIAN EMMANNUEL OLIVERO**

**COMISIÓN 10**

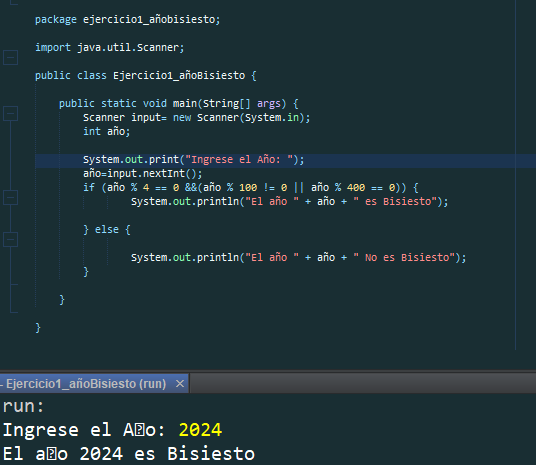
***ESTRUCTURAS CONDICIONALES:***

**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

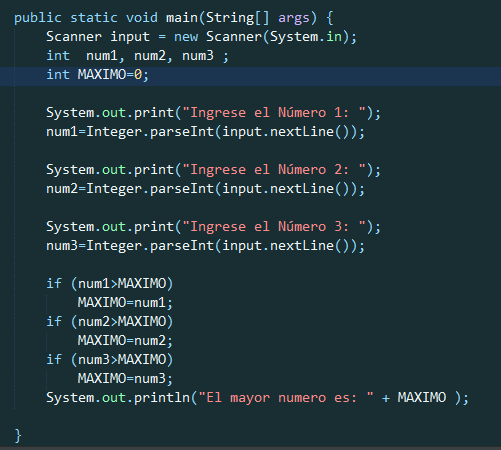
divisible por 400.

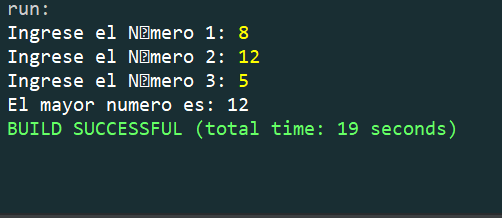


**2. Determinar el Mayor de Tres Números.**

Escribe un programa en Java que pida al usuario tres números enteros y

determine cuál es el mayor.





**3. Clasificación de Edad.**

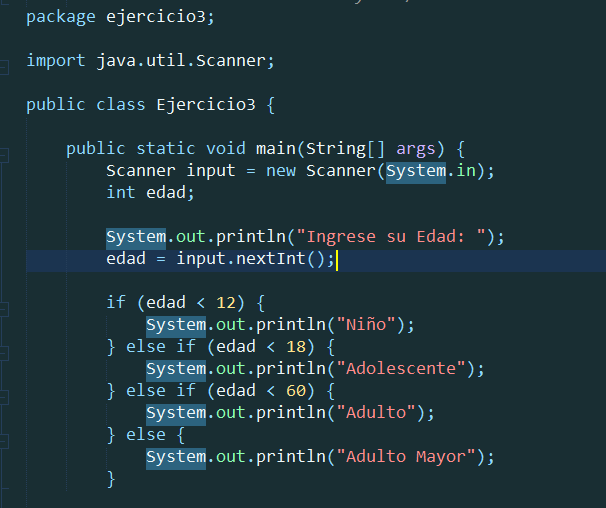
Escribe un programa en Java que solicite al usuario su edad y clasifique su etapa de vida según la siguiente tabla:

Menor de 12 años: "Niño"

Entre 12 y 17 años: "Adolescente"

Entre 18 y 59 años: "Adulto"

60 años o más: "Adulto mayor"



**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).

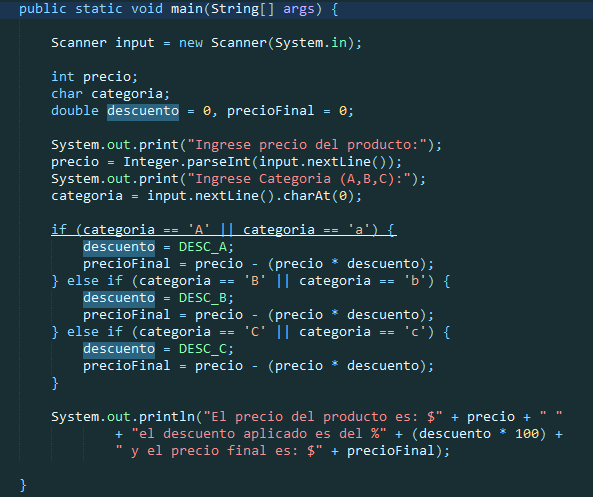
Luego, aplique los siguientes descuentos:

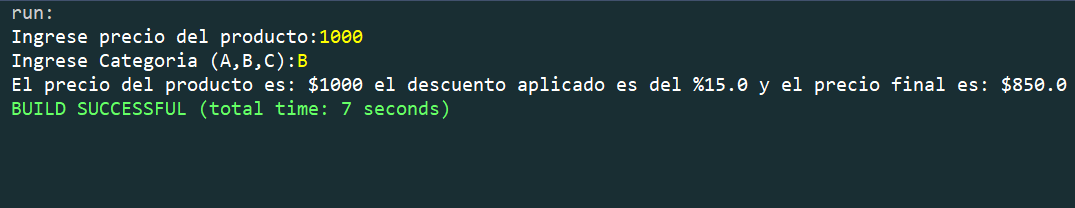
Categoría A: 10% de descuento

Categoría B: 15% de descuento

Categoría C: 20% de descuento

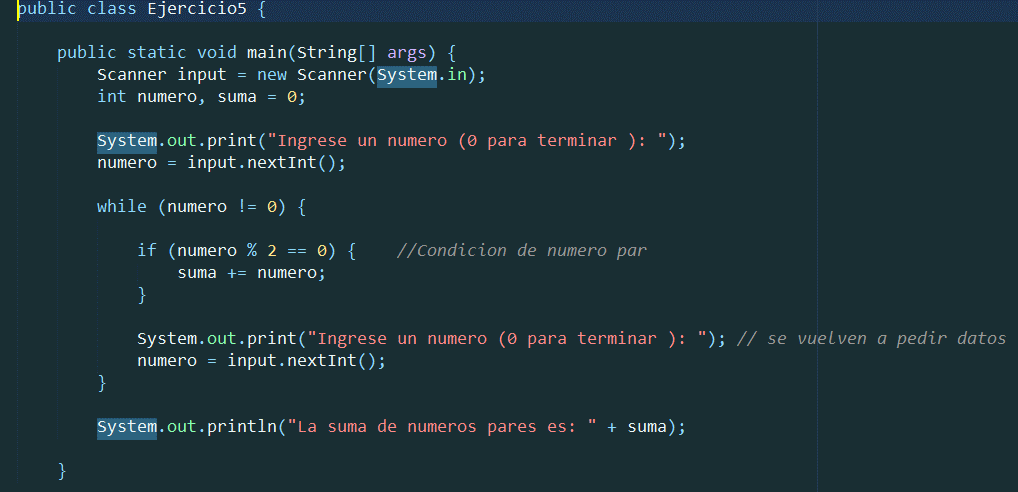
El programa debe mostrar el precio original, el descuento aplicado y el precio final

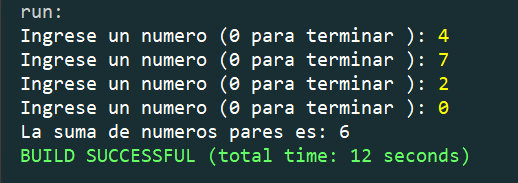




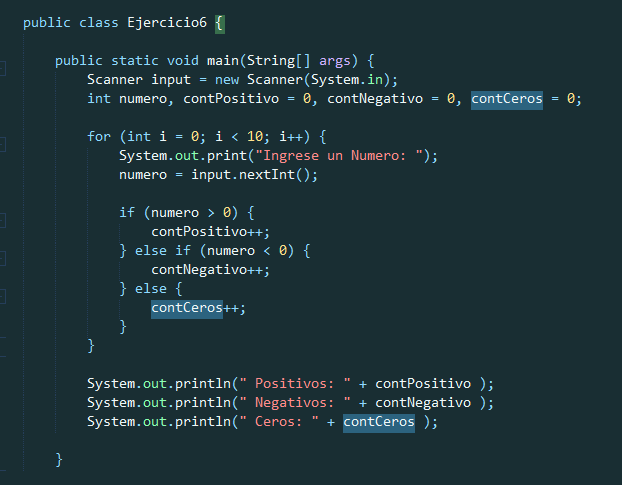
***ESTRUCTURAS DE REPETICIÓN:***

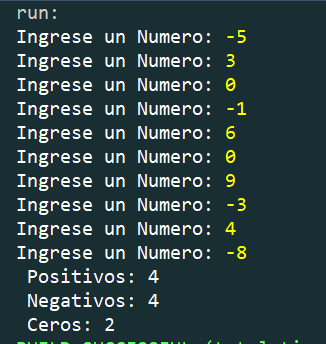
**5. Suma de Números Pares (while).**

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.

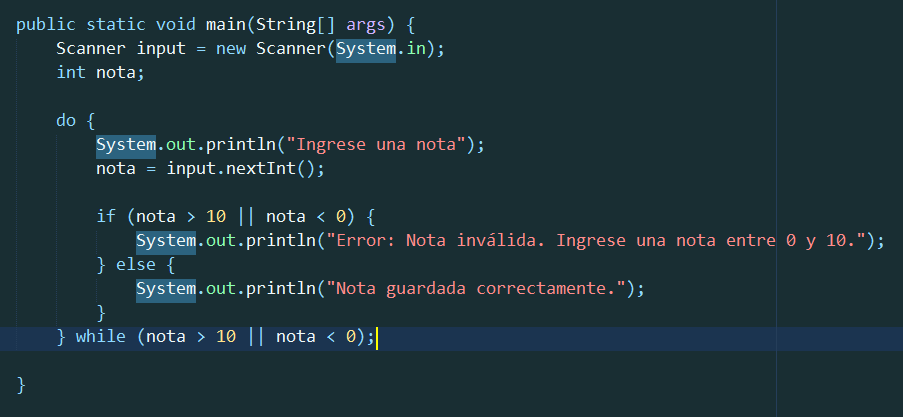


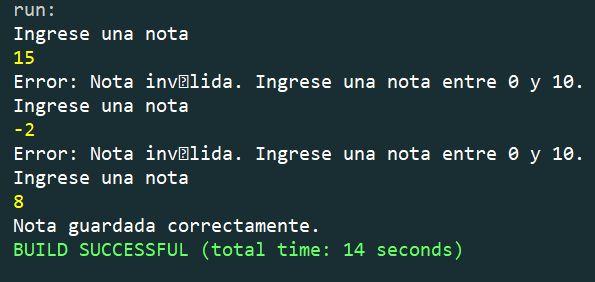
**6. Contador de Positivos, Negativos y Ceros (for).**

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



**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.

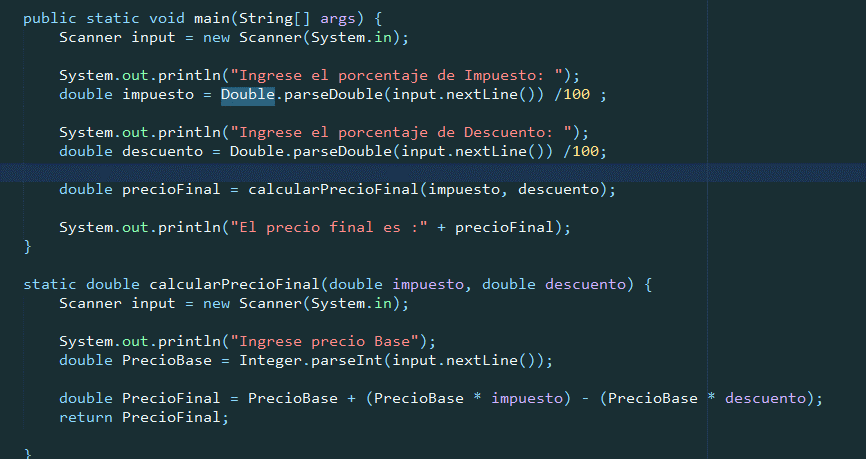


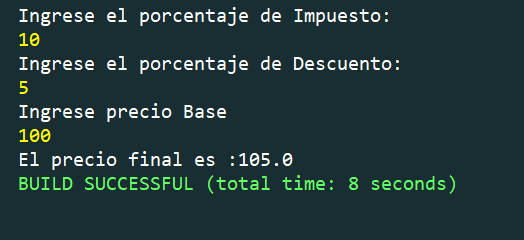
***FUNCIONES:***

**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:

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.



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

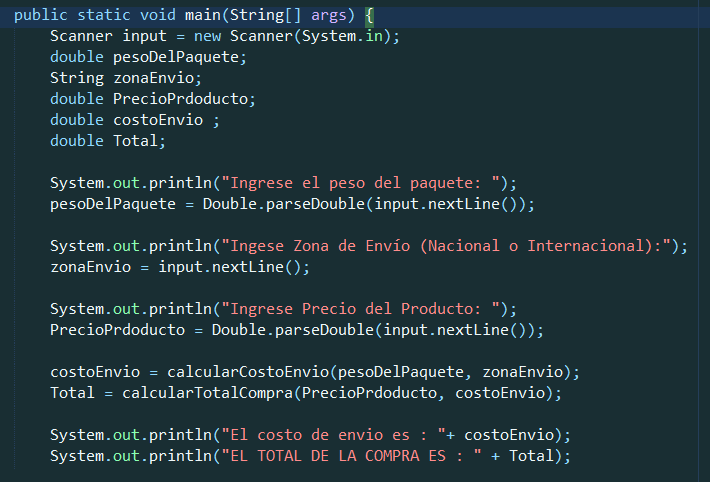
a. calcularCostoEnvio(double peso, String zona): Calcula el costo de envío basado en la zona de envío (Nacional o Internacional) y el peso del paquete.

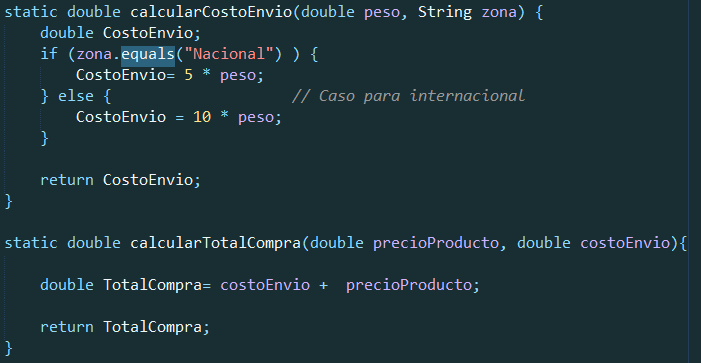
Nacional: $5 por kg

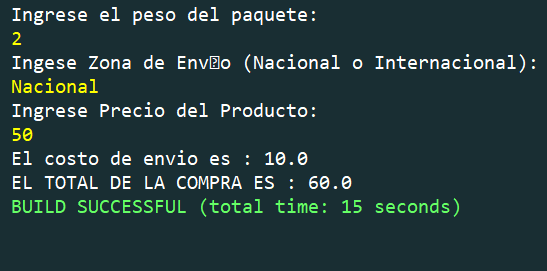
Internacional: $10 por kg

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.







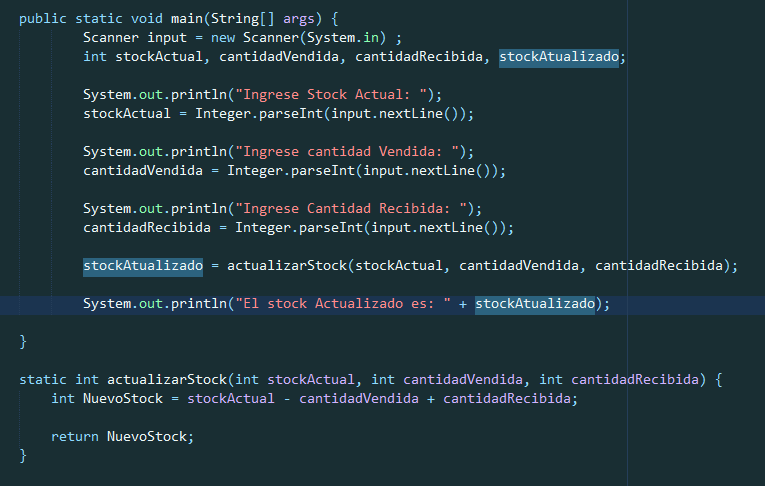
**10. Actualización de stock a partir de venta y recepción de productos.**

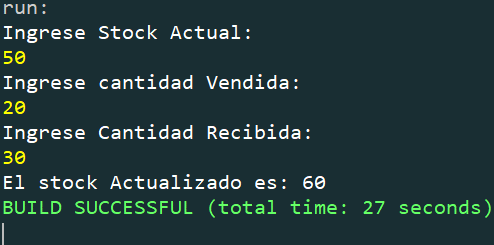
Crea un método actualizarStock(int stockActual, int cantidadVendida,

int cantidadRecibida), que calcule el nuevo stock después de una venta y recepción de productos:

NuevoStock = StockActual − CantidadVendida + CantidadRecibida

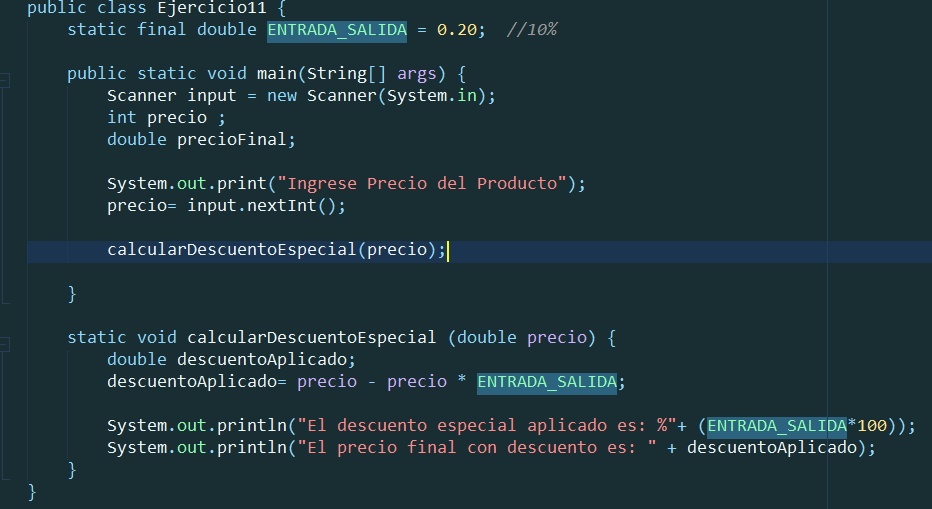
NuevoStock = CantidadVendida + CantidadRecibida

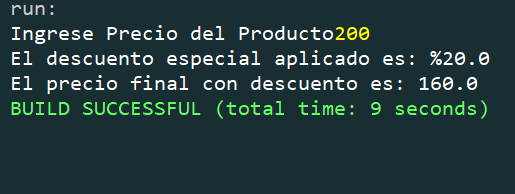
Desde main(), solicita al usuario el stock actual, la cantidad vendida y la cantidad recibida, y muestra el stock actualizado.



**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.



***ARRAYS Y RECURSIVIDAD:***

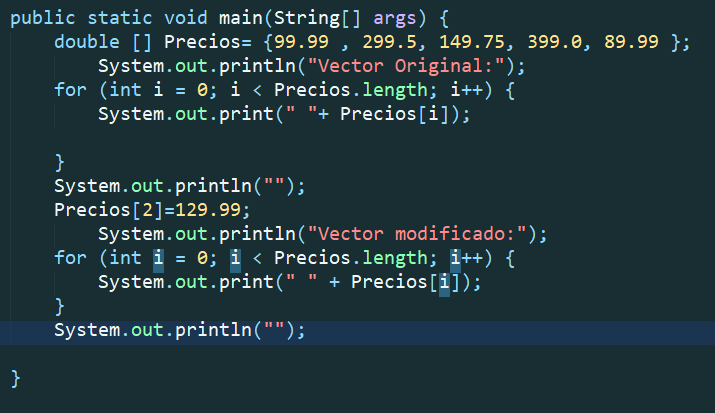
**12. Modificación de un array de precios y visualización de resultados.**

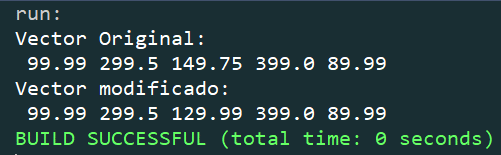
Crea un programa que:

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.



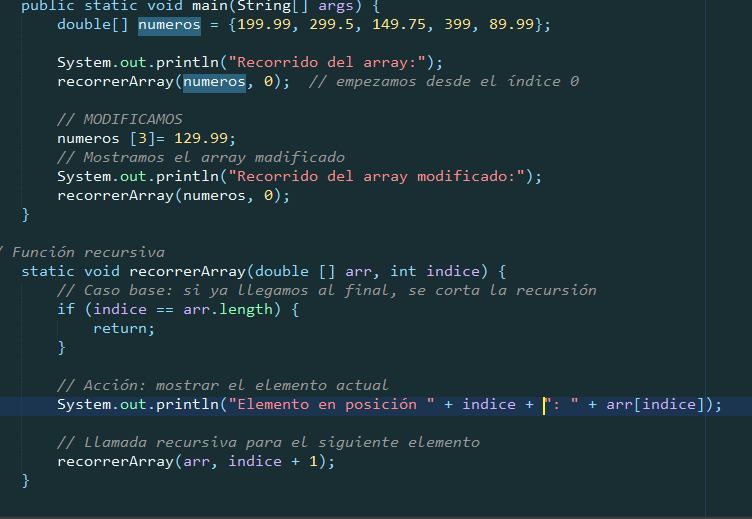
**13. Impresión recursiva de arrays antes y después de modificar un elemento.**

Crea un programa que:

a. Declare e inicialice un array con los precios de algunos productos.

b. Use una función recursiva para mostrar los precios originales.

c. Modifique el precio de un producto específico.

d. Use otra función recursiva para mostrar los valores modificados.

