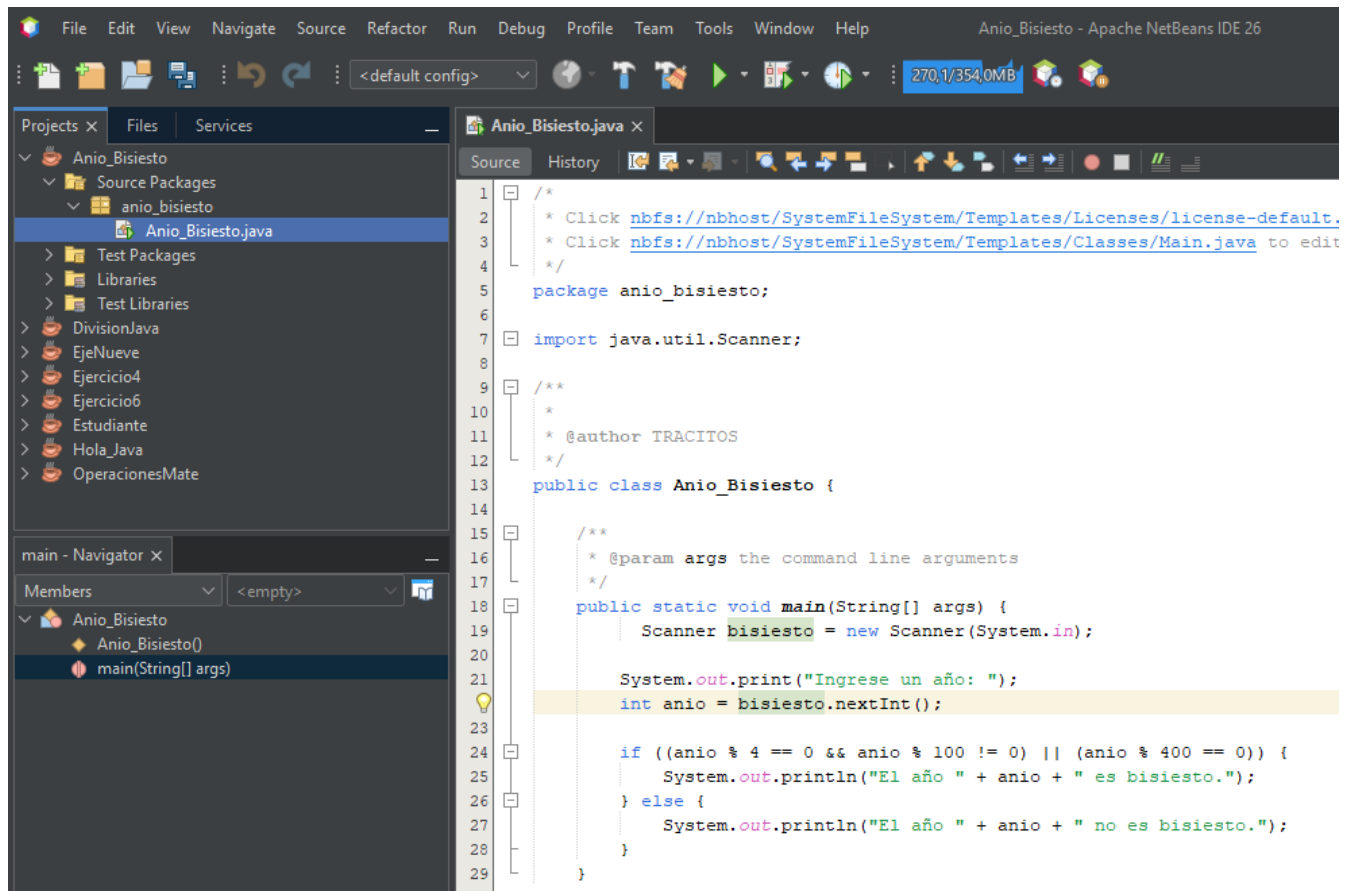


Estructuras Condicionales

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.



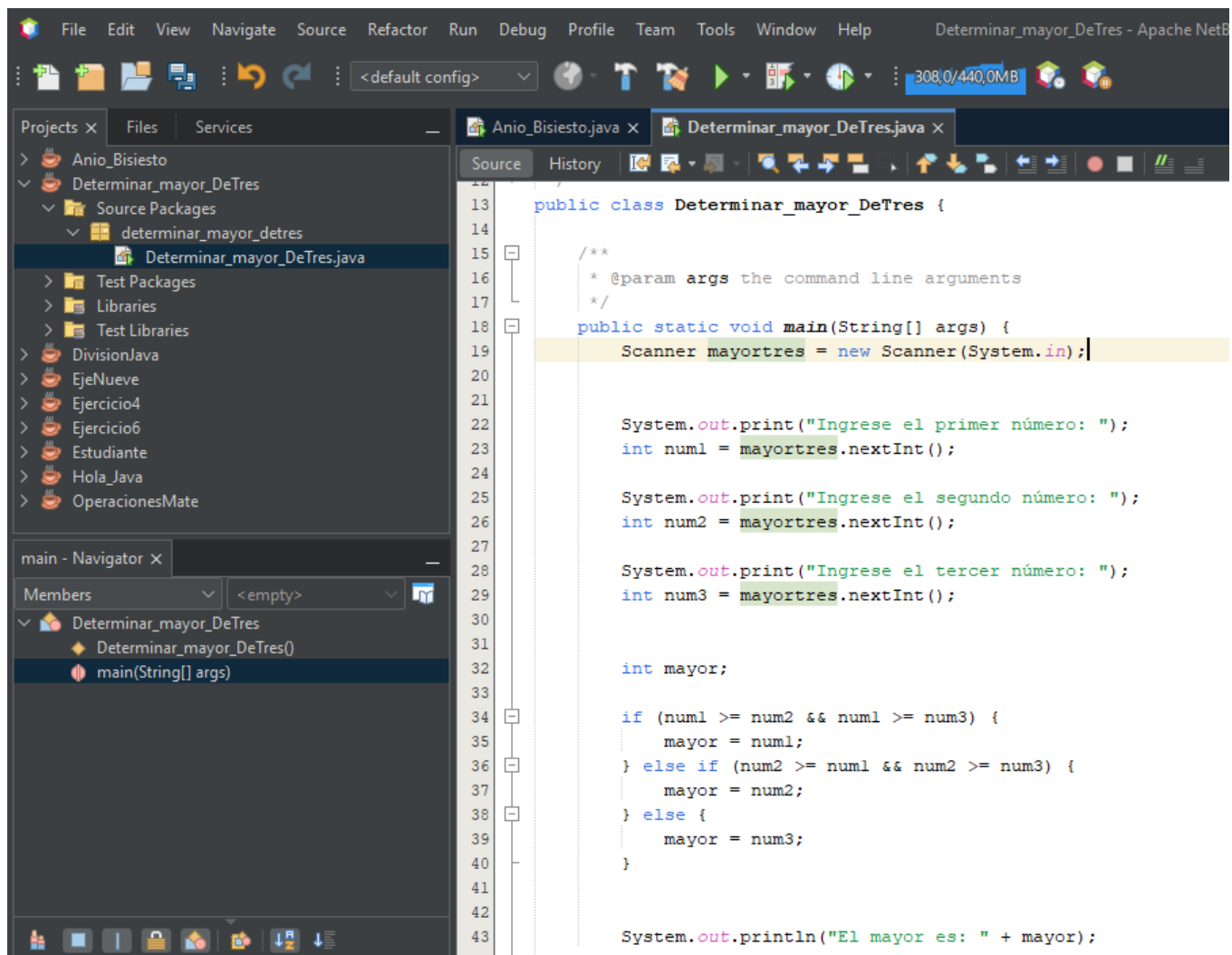
The screenshot shows the Apache NetBeans IDE interface. The main editor window displays the source code for 'Anio_Bisiesto.java'. The code is as follows:

```
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit
4   */
5  package anio_bisiesto;
6
7  import java.util.Scanner;
8
9  /**
10   *
11   * @author TRACITOS
12   */
13  public class Anio_Bisiesto {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          Scanner bisiesto = new Scanner(System.in);
20
21          System.out.print("Ingrese un año: ");
22          int anio = bisiesto.nextInt();
23
24          if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
25              System.out.println("El año " + anio + " es bisiesto.");
26          } else {
27              System.out.println("El año " + anio + " no es bisiesto.");
28          }
29      }
30  }
```

The left sidebar shows the project structure with 'Anio_Bisiesto.java' selected. The bottom sidebar shows the 'main' method in the 'Anio_Bisiesto' class.

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.



The screenshot shows an IDE with the following components:

- Top Bar:** Menus (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help) and a status bar showing memory usage (308,0/440,0MB).
- Left Panel:** A project tree showing a project named 'Anio_Bisiesto' with a sub-project 'Determinar_mayor_DeTres'. Inside this sub-project, there are 'Source Packages' and 'Test Packages'. The file 'Determinar_mayor_DeTres.java' is selected under 'Source Packages'. Below the project tree is a 'main - Navigator' panel showing the class 'Determinar_mayor_DeTres' with methods 'Determinar_mayor_DeTres()' and 'main(String[] args)'. The 'main' method is selected.
- Right Panel:** The source code of 'Determinar_mayor_DeTres.java'. The code is as follows:

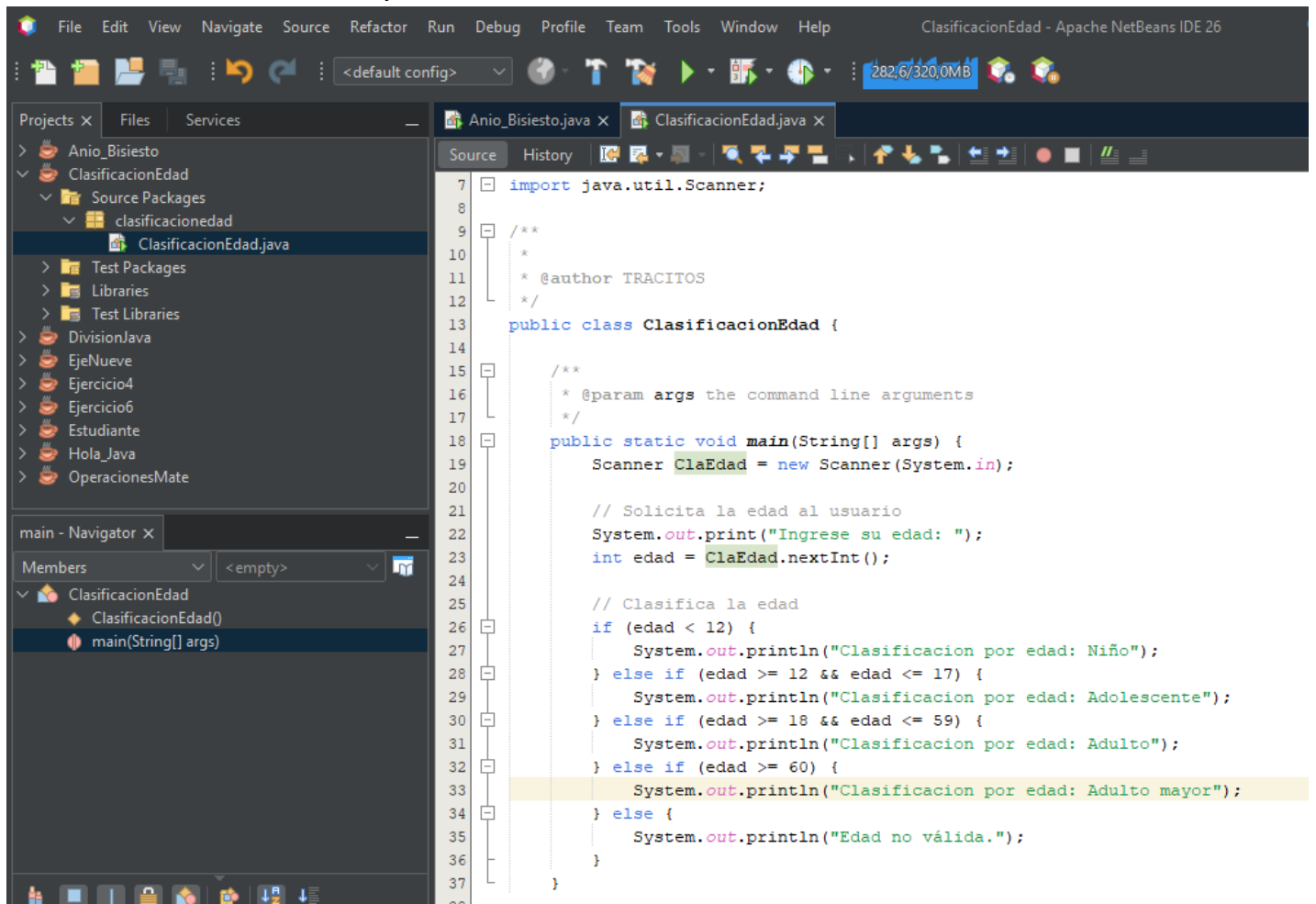
```
13 public class Determinar_mayor_DeTres {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         Scanner mayortres = new Scanner(System.in);
20
21         System.out.print("Ingrese el primer número: ");
22         int num1 = mayortres.nextInt();
23
24         System.out.print("Ingrese el segundo número: ");
25         int num2 = mayortres.nextInt();
26
27         System.out.print("Ingrese el tercer número: ");
28         int num3 = mayortres.nextInt();
29
30         int mayor;
31
32         if (num1 >= num2 && num1 >= num3) {
33             mayor = num1;
34         } else if (num2 >= num1 && num2 >= num3) {
35             mayor = num2;
36         } else {
37             mayor = num3;
38         }
39
40         System.out.println("El mayor es: " + mayor);
41     }
42 }
43
44 }
```

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".



```
7 import java.util.Scanner;
8
9 /**
10  *
11  * @author TRACITOS
12  */
13 public class ClasificacionEdad {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         Scanner ClaEdad = new Scanner(System.in);
20
21         // Solicita la edad al usuario
22         System.out.print("Ingrese su edad: ");
23         int edad = ClaEdad.nextInt();
24
25         // Clasifica la edad
26         if (edad < 12) {
27             System.out.println("Clasificacion por edad: Niño");
28         } else if (edad >= 12 && edad <= 17) {
29             System.out.println("Clasificacion por edad: Adolescente");
30         } else if (edad >= 18 && edad <= 59) {
31             System.out.println("Clasificacion por edad: Adulto");
32         } else if (edad >= 60) {
33             System.out.println("Clasificacion por edad: Adulto mayor");
34         } else {
35             System.out.println("Edad no válida.");
36         }
37     }
38 }
```

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:

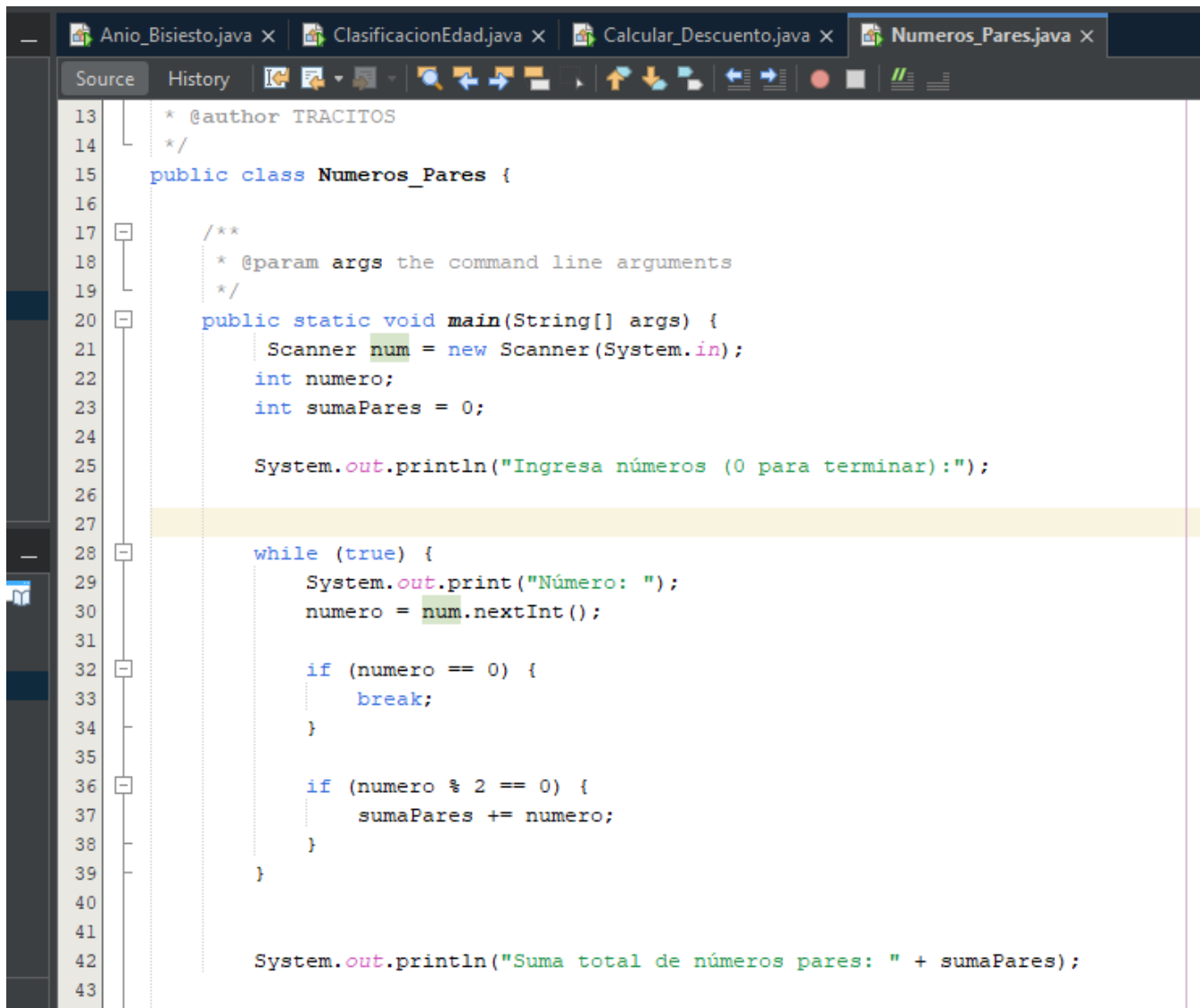
Categoría A: 10% de descuento. Categoría B: 15% de descuento. Categoría C: 20% de descuento. El programa debe mostrar el precio original, el descuento aplicado y el precio final.

```
config> 204.8/335.0MB
Anio_Bisiesto.java x ClasificacionEdad.java x Calcular_Descuento.java x
Source History
18 public static void main(String[] args) {
19     Scanner Precio = new Scanner(System.in);
20
21     System.out.print("Ingrese el precio del producto: ");
22     double precioOriginal = Precio.nextDouble();
23     Precio.nextLine();
24
25     System.out.print("Ingrese la categoría del producto (A, B o C): ");
26     String categoria = Precio.nextLine().toUpperCase();
27
28     double descuento = 0.0;
29
30
31     switch (categoria) {
32     case "A":
33         descuento = 0.10;
34         break;
35     case "B":
36         descuento = 0.15;
37         break;
38     case "C":
39         descuento = 0.20;
40         break;
41     default:
42         System.out.println("Categoría no válida. Debe ser A, B o C.");
43         Precio.close();
44         return;
45     }
46
47     double montoDescuento = precioOriginal * descuento;
48     double precioFinal = precioOriginal - montoDescuento;
49
50
51     System.out.printf("Precio original: $%.2f\n", precioOriginal);
52     System.out.printf("Descuento aplicado (%.0f%%): $%.2f\n", descuento * 100, montoDescuento);
53     System.out.printf("Precio final: $%.2f\n", precioFinal);
54
55
56 }
```

Estructuras de Repetición:

5. Suma de Números Pares (while).

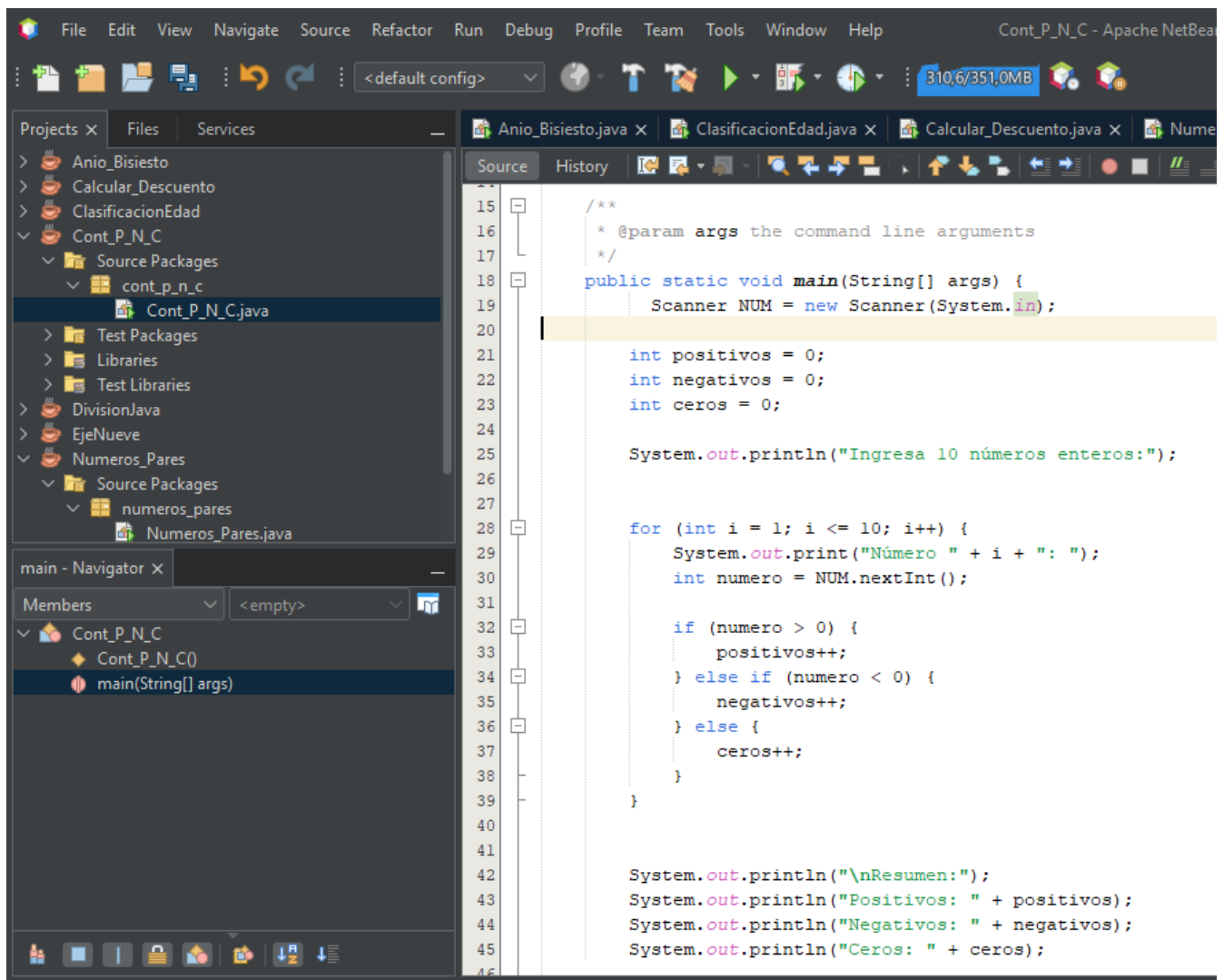
Escribe un programa que solicite números al usuario y sume solo los números pares. El ciclo debe continuar hasta que el usuario ingrese el número 0, momento en el que se debe mostrar la suma total de los pares ingresados.



```
13  * @author TRACITOS
14  */
15  public class Numeros_Pares {
16
17      /**
18       * @param args the command line arguments
19       */
20      public static void main(String[] args) {
21          Scanner num = new Scanner(System.in);
22          int numero;
23          int sumaPares = 0;
24
25          System.out.println("Ingresa números (0 para terminar):");
26
27
28          while (true) {
29              System.out.print("Número: ");
30              numero = num.nextInt();
31
32              if (numero == 0) {
33                  break;
34              }
35
36              if (numero % 2 == 0) {
37                  sumaPares += numero;
38              }
39          }
40
41
42          System.out.println("Suma total de números pares: " + sumaPares);
43  }
```

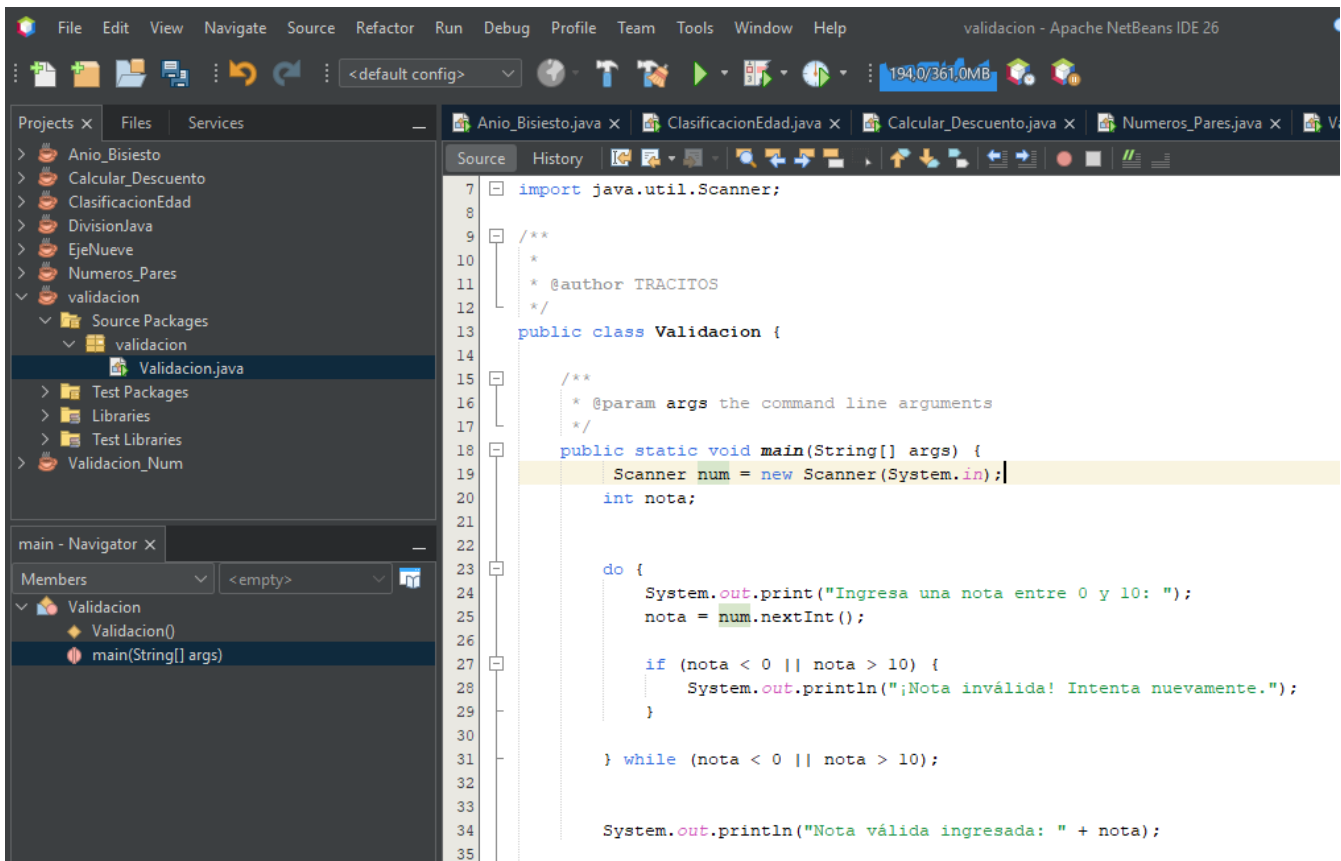
6. Contador de Positivos, Negativos y Ceros (for):

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.



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.

```
...va  Calculador_Descuento.java x  Numeros_Pares.java x  Validacion_Num.java x  Validacion.java x  Precio_impuesto.java x
Source History
13 public class Precio_impuesto {
14     public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento) {
15         double precioFinal = precioBase + (precioBase * impuesto) - (precioBase * descuento);
16         return precioFinal;
17     }
18
19     /**
20      * @param args the command line arguments
21      */
22     public static void main(String[] args) {
23         Scanner saldo = new Scanner(System.in);
24
25
26         System.out.print("Ingrese el precio base del producto: ");
27         double precioBase = saldo.nextDouble();
28
29
30         System.out.print("Ingrese el porcentaje de impuesto (por ejemplo, 18 para 18%): ");
31         double impuestoPorcentaje = saldo.nextDouble();
32
33
34         System.out.print("Ingrese el porcentaje de descuento (por ejemplo, 10 para 10%): ");
35         double descuentoPorcentaje = saldo.nextDouble();
36
37
38         double impuestoDecimal = impuestoPorcentaje / 100;
39         double descuentoDecimal = descuentoPorcentaje / 100;
40
41
42         double precioFinal = calcularPrecioFinal(precioBase, impuestoDecimal, descuentoDecimal);
43
44         System.out.printf("El precio final del producto es: $%.2f\n", precioFinal);
45     }
46 }
```

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

```
...va  Numeros_Pares.java x  Validacion_Num.java x  Validacion.java x  Precio_impuesto.java x  CostoEnvio_TotalCompra.java x
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5  package costoenvio_totalcompra;
6
7  import java.util.Scanner;
8
9  /**
10   *
11   * @author TRACITOS
12   */
13  public class CostoEnvio_TotalCompra {
14
15      public static double calcularCostoEnvio(double peso, String zona) {
16          double costoPorKg;
17
18
19          if (zona.equalsIgnoreCase("Nacional")) {
20              costoPorKg = 5.0;
21          } else if (zona.equalsIgnoreCase("Internacional")) {
22              costoPorKg = 10.0;
23          } else {
24
25              System.out.println("Zona no válida. Se asumirá costo de envío 0.");
26              return 0.0;
27          }
28
29          return peso * costoPorKg;
30      }
31  }
```



```
...va Numeros_Pares.java x Validacion_Num.java x Validacion.java x Precio_impuesto.java x CostoEnvio_TotalCompra.java x
Source History
33
34 public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
35     return precioProducto + costoEnvio;
36 }
37
38 /**
39  * @param args the command line arguments
40  */
41 public static void main(String[] args) {
42     Scanner num = new Scanner(System.in);
43
44     System.out.print("Ingrese el peso del paquete (en kg): ");
45     double peso = num.nextDouble();
46     num.nextLine();
47
48     System.out.print("Ingrese la zona de envío (Nacional o Internacional): ");
49     String zona = num.nextLine();
50
51     System.out.print("Ingrese el precio del producto: ");
52     double precioProducto = num.nextDouble();
53
54     double costoEnvio = calcularCostoEnvio(peso, zona);
55
56     double total = calcularTotalCompra(precioProducto, costoEnvio);
57
58     System.out.printf("Costo de envío: $%.2f\n", costoEnvio);
59     System.out.printf("Total a pagar: $%.2f\n", total);
60 }
61
62
63
64
```

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

```
...va Validacion_Num.java x Validacion.java x Precio_impuesto.java x CostoEnvio_TotalCompra.java x Stock_Ej10.java x
Source History
7 import java.util.Scanner;
8
9 /**
10  *
11  * @author TRACITOS
12  */
13 public class Stock_Ej10 {
14
15     public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
16         return stockActual - cantidadVendida + cantidadRecibida;
17     }
18
19     public static void main(String[] args) {
20         Scanner num = new Scanner(System.in);
21
22         // Solicitar datos al usuario
23         System.out.print("Ingrese el stock actual: ");
24         int stockActual = num.nextInt();
25
26         System.out.print("Ingrese la cantidad vendida: ");
27         int cantidadVendida = num.nextInt();
28
29         System.out.print("Ingrese la cantidad recibida: ");
30         int cantidadRecibida = num.nextInt();
31
32         // Calcular el nuevo stock
33         int nuevoStock = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);
34
35         // Mostrar el resultado
36         System.out.println("El stock actualizado es: " + nuevoStock);
37     }
38 }
```

11. Cálculo de descuento especial usando variable global.

```
...va  Validacion.java x  Precio_impuesto.java x  CostoEnvio_TotalCompra.java x  Stock_Ej10.java x  Descuento_Eje11.java x
Source  History  [Icons]

4  L  */
5  package descuento_eje11;
6
7  [ ] import java.util.Scanner;
8
9  [ ] /**
10  *
11  * @author TRACITOS
12  */
13  public class Descuento_Eje11 {
14
15      static double descuentoEspecial = 0.10;
16
17  [ ] public static void calcularDescuentoEspecial(double precio) {
18
19      double descuentoAplicado = precio * descuentoEspecial;
20      double precioFinal = precio - descuentoAplicado;
21
22      System.out.printf("Descuento aplicado: $%.2f%n", descuentoAplicado);
23      System.out.printf("Precio final con descuento: $%.2f%n", precioFinal);
24  }
25
26
27  [ ] public static void main(String[] args) {
28
29      Scanner num = new Scanner(System.in);
30
31      System.out.print("Ingrese el precio del producto: ");
32      double precio = num.nextDouble();
33
34      calcularDescuentoEspecial(precio);
35  }
```

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

```
...va  Precio_impuesto.java x  CostoEnvio_TotalCompra.java x  Stock_Ej10.java x  Descuento_Eje11.java x  Arrays_Eje12.java x
Source  History  [Icons]

13  public class Arrays_Eje12 {
14
15  [ ] /**
16  * @param args the command line arguments
17  */
18  [ ] public static void main(String[] args) {
19
20      Scanner num = new Scanner(System.in);
21
22      double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
23
24      System.out.println("Precios originales:");
25      mostrarArray(precios);
26
27
28      System.out.print("\nIngrese el índice del producto a modificar (0 a " + (precios.length - 1) + "): ");
29      int indice = num.nextInt();
30
31  [ ] if (indice >= 0 && indice < precios.length) {
32      System.out.print("Ingrese el nuevo precio: ");
33      double nuevoPrecio = num.nextDouble();
34      precios[indice] = nuevoPrecio; // Actualiza el precio
35  [ ] } else {
36      System.out.println("Índice inválido. No se modificó ningún precio.");
37  }
38
39
40      System.out.println("\nPrecios actualizados:");
41      mostrarArray(precios);
42
43  }
```

```
43     }
44
45     public static void mostrarArray(double[] array) {
46         for (int i = 0; i < array.length; i++) {
47             System.out.printf("Producto %d: $%.2f%n", i, array[i]);
48         }
49     }
50
51 }
52
```

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