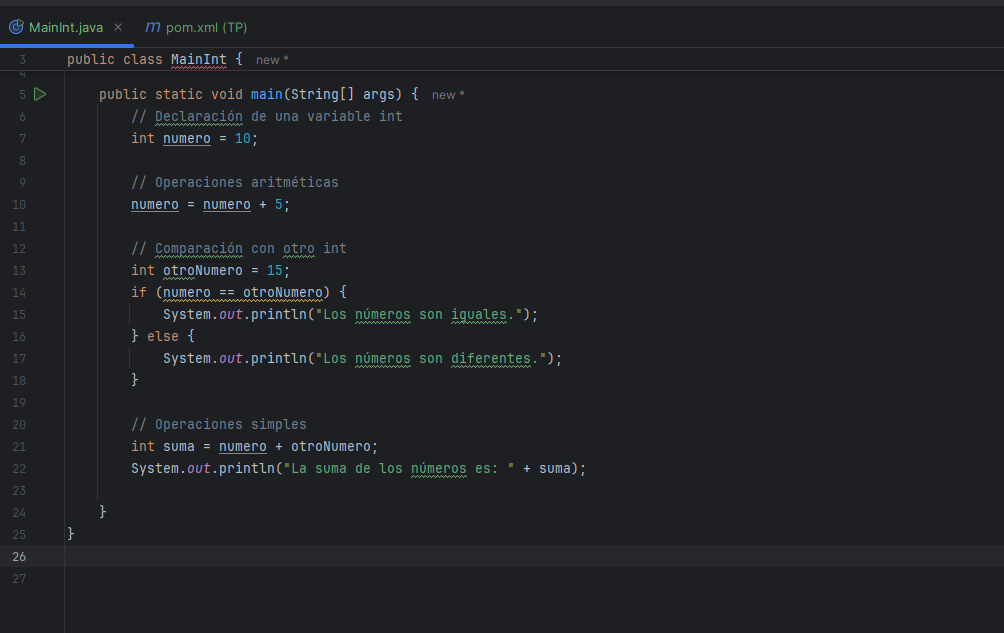
14- Indique cual es la diferencia entre los datos primitivos (por valor) ejemplo el int y su contraparte por referencia ejemplo Integer.



#### **Características del código con int:**

1. **Rendimiento:** El uso de int es más eficiente en términos de memoria y rendimiento, ya que los tipos primitivos no tienen la sobrecarga de un objeto.
2. **Operaciones básicas:** Se pueden realizar operaciones aritméticas y comparaciones directamente, sin necesidad de métodos adicionales.
3. **No admite valores nulos:** No se puede asignar null a una variable de tipo int.
4. **Paso de valor:** Cuando se pasa un int a un método, se pasa una copia del valor.

#### 

#### **Características del código con Integer:**

1. **Rendimiento:** Usar Integer es menos eficiente que int debido a la sobrecarga de crear y manipular un objeto.
2. **Autoboxing y Unboxing:** Java convierte automáticamente entre int y Integer cuando se realizan operaciones aritméticas o comparaciones, lo que se conoce como autoboxing y unboxing.
3. **Admite valores nulos:** A diferencia de int, una variable de tipo Integer puede contener el valor null. Esto puede ser útil en contextos donde un valor nulo tiene significado (por ejemplo, en estructuras de datos).
4. **Métodos adicionales:** Integer ofrece métodos como equals() para comparar valores, y otros métodos útiles como parseInt() y toString().

15- Indique que sucede si realizo la siguiente declaración de variable:

int numero = null;

¿Qué debo modificar para poder asignar null a la variable?

En Java, las variables de tipo primitivo, como int, no pueden tener el valor null. Esto se debe a que null es un valor especial que se utiliza para indicar la ausencia de un objeto en las variables de tipo referencia (como objetos), pero no tiene sentido para los tipos primitivos, que siempre deben contener un valor concreto.

Por lo que si declaro la variable “int numero = null;” tendre un error de compilacion porque int es un tipo primitivo y no puede almacenar el valor null.

Entonces para poder asignar null a una variable que representa un número, debes usar la clase envolvente, la cual explique en el punto 14, correspondiente al tipo primitivo. En el caso de int, la clase envolvente es Integer.

Y el código quedaría de la siguiente manera: “Integer numero = null;”

16- Como transformo una variable tipo Double a una variable tipo double.

Para convertir una variable de tipo Double (una clase envolvente) a double (un tipo primitivo) en Java, puedes hacer una conversión de un tipo a otro, que se conoce como "unboxing". El proceso de unboxing en Java es automático y implícito cuando se realiza una asignación o una operación que requiere un tipo primitivo.

17- Cree una clase FuncionesPrograma y codifique una función estática que reciba como parámetro una fecha de tipo Date y retorne la fecha como una cadena.

public static String getFechaString(Date fecha){

……….

}

Cree una clase Principal que contenga un método main y haga uso de la función getFechaString

18- En la clase FuncionesPrograma codifique una función estática que reciba como parámetro 3 valores enteros, día, mes, anio y retorne la fecha de tipo Date correspondiente.

public static Date getFechaDate(int día, int mes, int anio){

……….

}

En la clase Principal creada en el punto anterior haga uso de la función getFechaDate.