1.Cumple el principio de Inversión de dependencia. Justifica la respuesta.

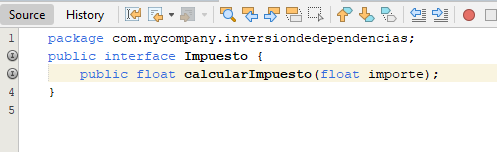
RTA: No, la implementación de la clase Factura no cumple con el principio de Inversión de Dependencia (IoD). El principio de IoD establece que los módulos de alto nivel no deben depender de módulos de bajo nivel, ambos deben depender de abstracciones.

En este caso, la clase Factura depende directamente de las clases Deduccion e Iva, lo cual no cumple con IoD, ya que son módulos de bajo nivel y no se han definido abstracciones para ellos.

Para cumplir con el principio de IoD, se podría definir una interfaz o clase abstracta que represente el concepto de "impuesto" y luego implementar las clases Deduccion e Iva para que implementen esa interfaz o extiendan esa clase abstracta. De esta forma, la clase Factura dependería solo de la abstracción "impuesto", en lugar de depender directamente de las clases Deduccion e Iva.

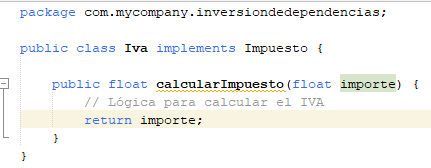
2. En caso negativo, refactoriza el código par que cumpla el principio.

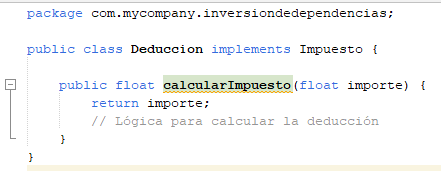
Crear una interfaz o clase abstracta que represente el concepto de "impuesto":



La interfaz define un método calcularImpuesto que toma como parámetro el importe base sobre el cual se calcula el impuesto y devuelve el importe del impuesto.

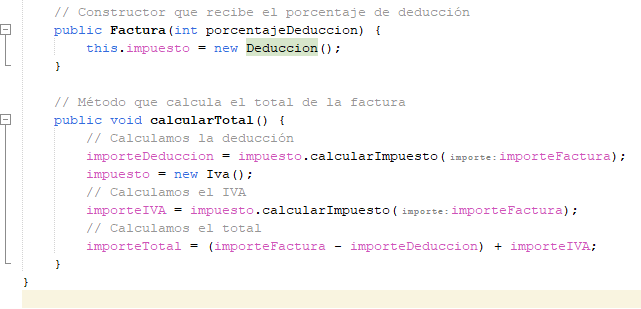
Refactorizar las clases Deduccion e Iva para que implementen la interfaz Impuesto:





mbas clases ahora implementan la interfaz Impuesto y proporcionan una implementación del método calcularImpuesto.

Refactorizar la clase Factura para que dependa de la abstracción Impuesto en lugar de depender directamente de las clases Deduccion e Iva. Además, se eliminará el atributo porcentajeDeduccion y se pasará este valor como parámetro al constructor de la clase Deduccion al crear una instancia de la misma:



Con esta refactorización, la clase Factura depende de la abstracción Impuesto en lugar de depender directamente de las clases Deduccion e Iva, lo que cumple con el principio de Inversión de Dependencia. Además, la eliminación del atributo porcentajeDeduccion hace que la clase Factura sea más flexible, ya que se puede cambiar el porcentaje de deducción al crear una nueva instancia de la clase Deduccion.