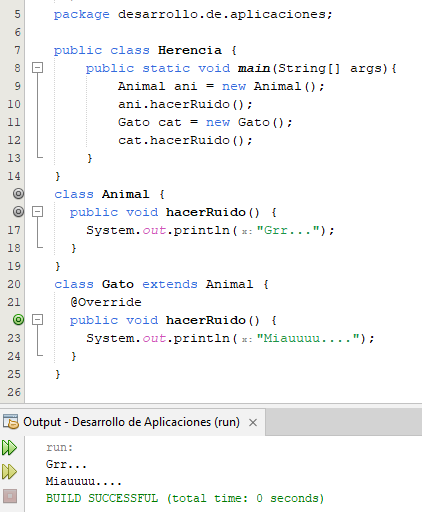
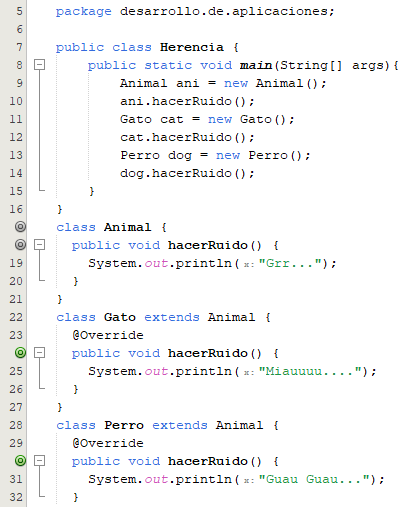
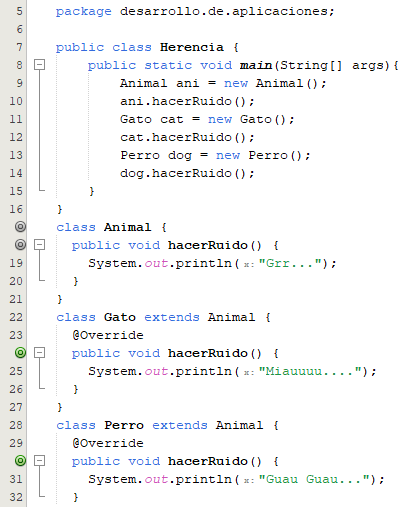
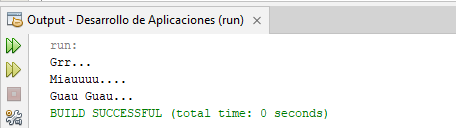
**Desarrollo Avanzado de Aplicaciones I**

**Ejercicio 1**. Desarrollar una aplicación que use una clase “Animal” y que tenga una subclase “Gato”. La clase “Animal” debe tener un método “hacerRuido” y en la clase “Gato” se debe sobreescribir ese método.

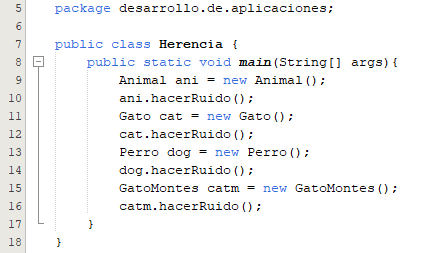


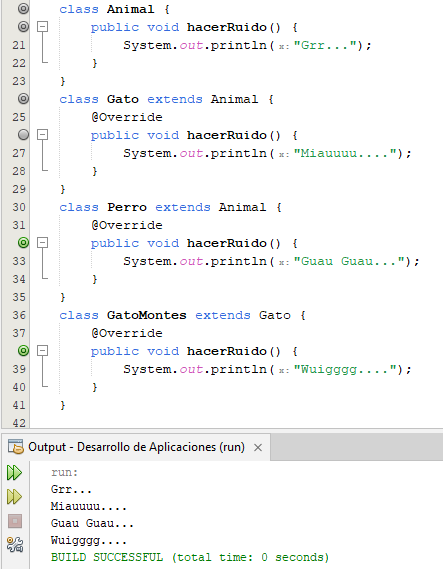
**Ejercicio 2**. Modificar el ejercicio anterior y agregar una subclase “Perro” que derive de la clase animal.



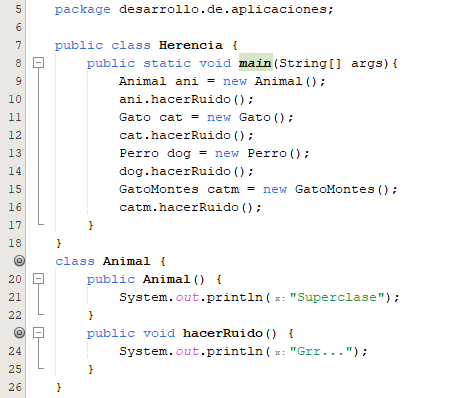


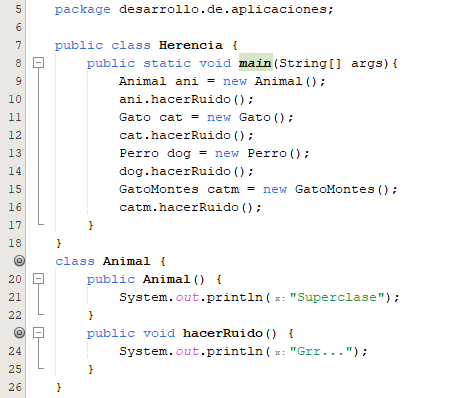
**Ejercicio 3**. Modificar el ejercicio anterior y agregar una subclase “GatoMontes” que derive de la clase “Gato”.

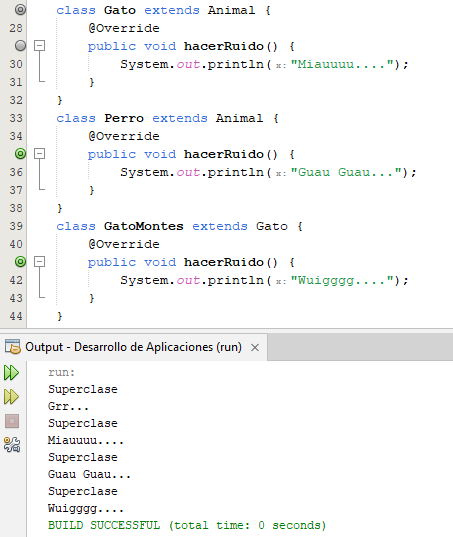




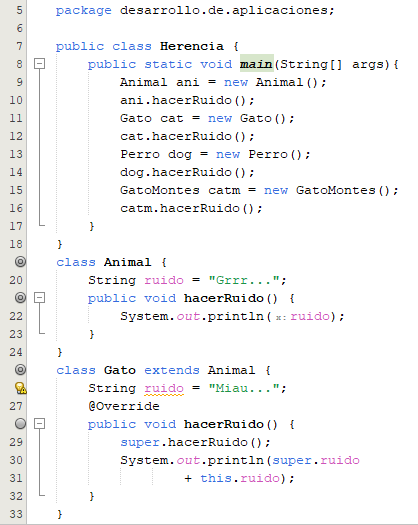
**Ejercicio 4**. Modificar el ejercicio anterior para que la clase “Animal” tenga un constructor que imprima el mensaje “Superclase” y se visualice el instanciamento de las superclases en cascada.

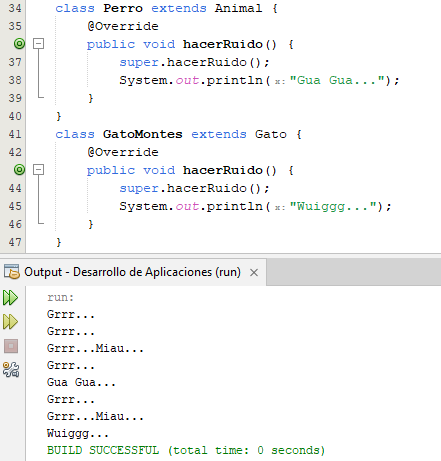




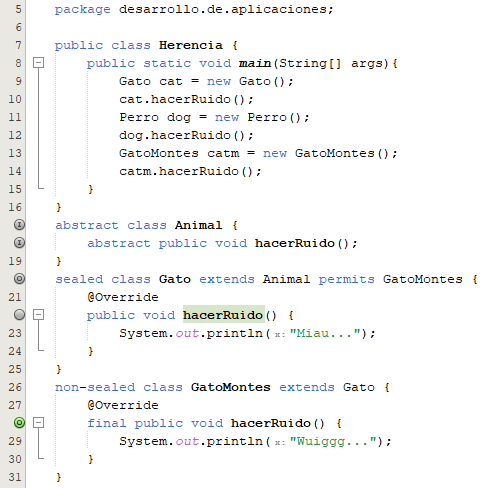


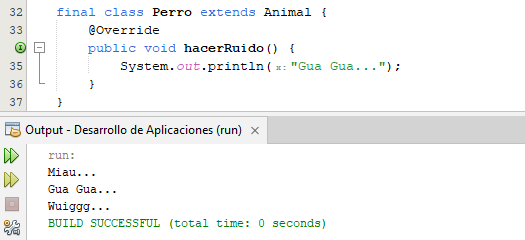
**Ejercicio 5**. Modificar el ejercicio anterior para que todas las clases derivadas llamen al método “hacerRuido” de su superclase. Eliminar el constructor de la clase “Animal”. Agregarle un atributo “ruido” de tipo “String” a la clase “Animal” y a la clase “Gato”, asignarle el valor de su sonido y usarlo en el método “hacerRuido”.



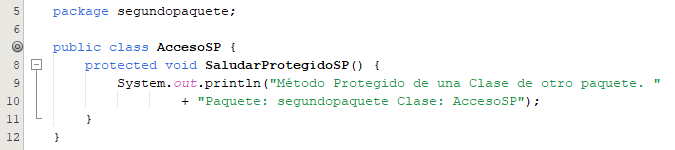


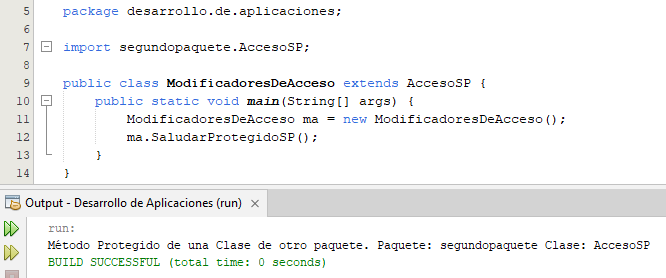
**Ejercicio 6**. Modificar el ejercicio anterior usando los diferentes modificadores de comportamiento aplicado a las clases y métodos usando herencia.





**Ejercicio 7**. Definir una clase en el paquete “segundopaquete” que se llame “AccesoSP”. En la clase “AccesoSP” definirle un método “SaludarProtegidoSP” con modificador de acceso protegido y que muestre un mensaje indicando que es un método protegido. Crear una clase “ModificadoresDeAcceso” en el paquete “desarrollo.de.aplicaciones” que sea derivada de la clase “AccesoSP” y que en su método main instancia su propia clase y llame al método protegido de la superclase o clase padre.



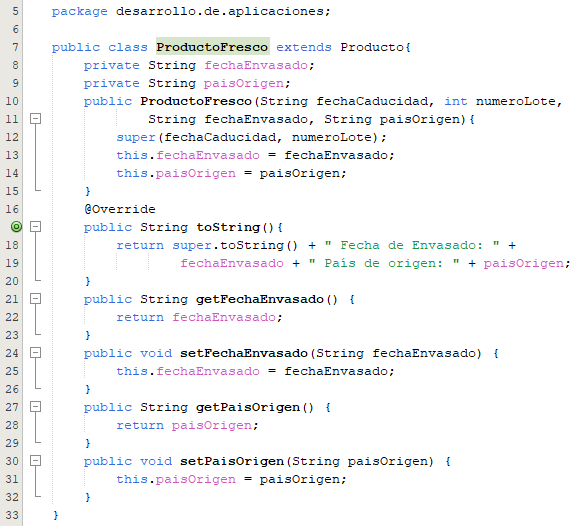


**Ejercicio 8**. Desarrollar una aplicación que permita la gestión de una empresa agroalimentaria que trabaja con tres tipos de productos: productos frescos, productos refrigerados y productos congelados. Todos los productos llevan esta información común: fecha de caducidad y número de lote. A su vez, cada tipo de producto lleva alguna información específica. Los productos frescos deben llevar la fecha de envasado y el país de origen. Los productos refrigerados deben llevar el código del organismo de supervisión alimentaria. Los productos congelados deben llevar la temperatura de congelación recomendada. Crear el código de las clases Java implementando una relación de herencia desde la superclase Producto hasta las subclases ProductoFresco, ProductoRefrigerado y ProductoCongelado. Cada clase debe disponer de constructor y permitir establecer (set) y recuperar (get) el valor de sus atributos y tener un método que permita mostrar la información del objeto. Crear una clase que tenga el método main donde se cree un objeto de cada tipo y se muestren los datos de cada uno de los objetos creados.

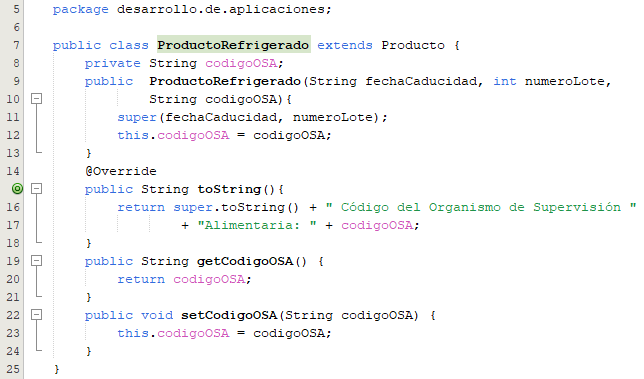
Archivo Producto.java



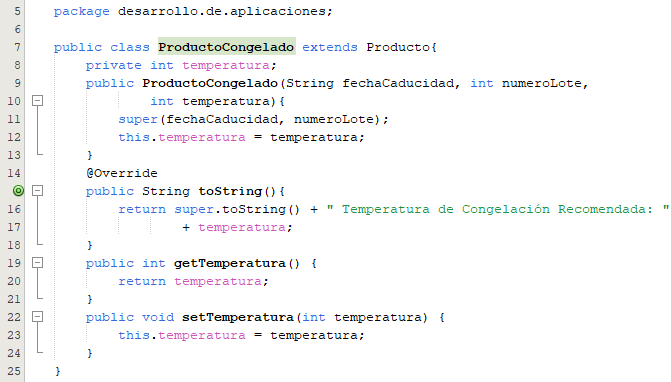
Archivo ProductoFresco.java



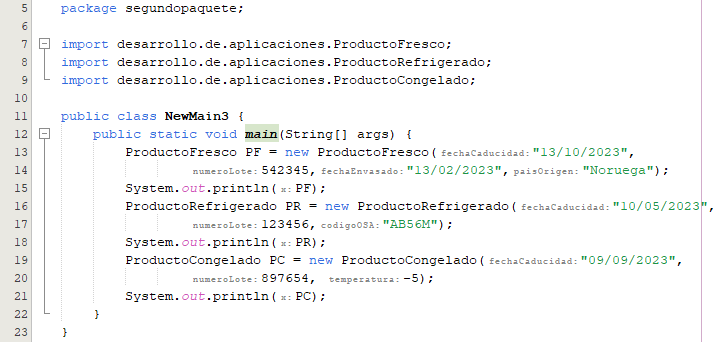
Archivo ProductoRefrigerado.java

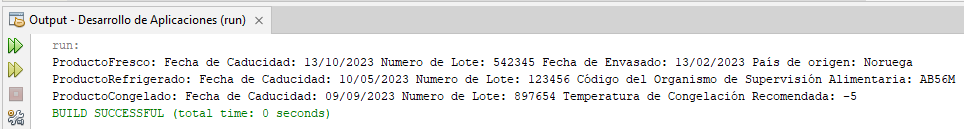


Archivo ProductoCongelado.java



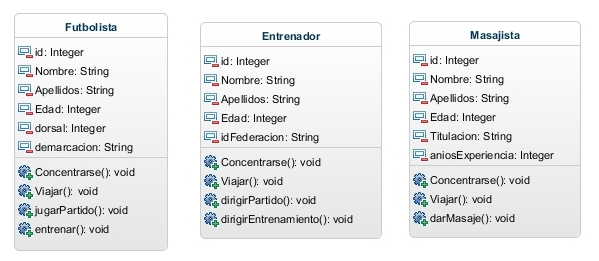
Archivo NewMain3.java





**Ejercicio 9**. A partir de las clases “Futbolista”, “Entrenador” y “Masajista”, crear una superclase, que permita generalizar los atributos y operaciones comunes y redefinir las 3 clases como subclases. Seguir las siguientes instrucciones:

* Definir los atributos de la superclase con modificador de acceso protegido.
* La superclase debe tener un constructor con parámetros que debe recibir todos los atributos de la superclase.
* Cada subclase debe tener su constructor con parámetros que invoque al constructor de la superclase.
* A las subclases agregarle los métodos getters y setters necesarios.
* Al método “Concentrarse” de la superclase agregarle código que muestre un mensaje utilizando los atributos “nombre” y “apellidos” indicando que está concentrando.
* Al método “Viajar” de la superclase agregarle código que muestre un mensaje utilizando los atributos “nombre” y “apellidos” indicando que está viajando.
* De forma similar implementar los métodos específicos de las subclases mostrando un mensaje de que actividad están realizando.
* Crear otra clase que tenga un método main donde se instancien las subclases y que se llamen a todos sus métodos.



Los resultados se deben mostrar de la siguiente manera:

Todos los integrantes comienzan una concentracion. (Todos ejecutan el mismo método)

Vicente Del Bosque está concentrando.

Andres Iniesta está concentrando.

Raúl Martinez está concentrando.

Todos los integrantes viajan para jugar un partido. (Todos ejecutan el mismo método)

Vicente Del Bosque está viajando.

Andres Iniesta está viajando.

Raúl Martinez está viajando

Entrenamiento: Solamente el entrenador y el futbolista tiene métodos para entrenar:

Vicente Del Bosque está dirigiendo un entrenamiento.

Andres Iniesta está entrenando.

Masaje: Solo el masajista tiene el método para dar un masaje:

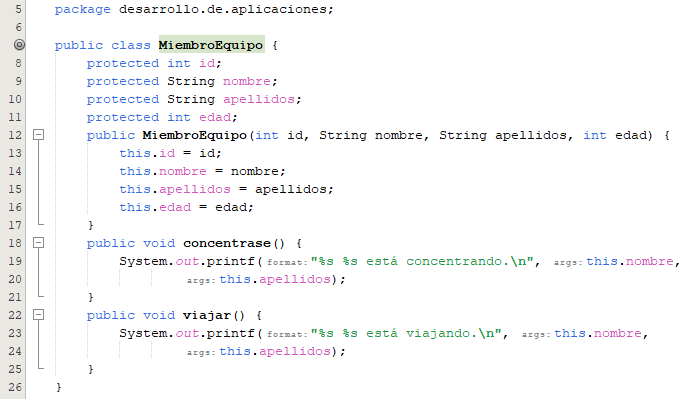
Raúl Martinez está realizando un masaje.

Partido de Fútbol: Solamente el entrenador y el futbolista tiene métodos para el partido de fútbol:

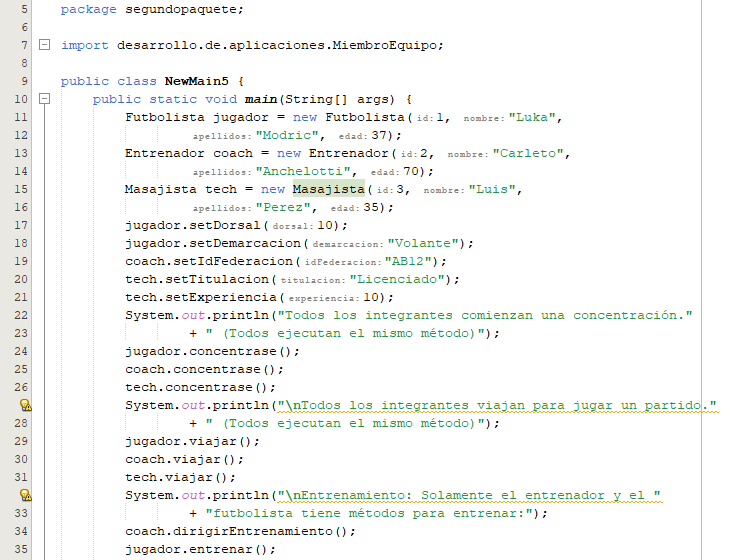
Vicente Del Bosque está dirigiendo un partido.

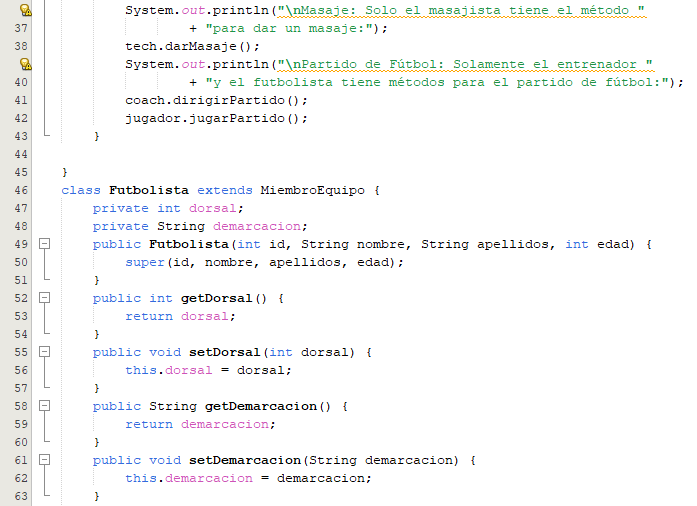
Andres Iniesta está jugando un partido.

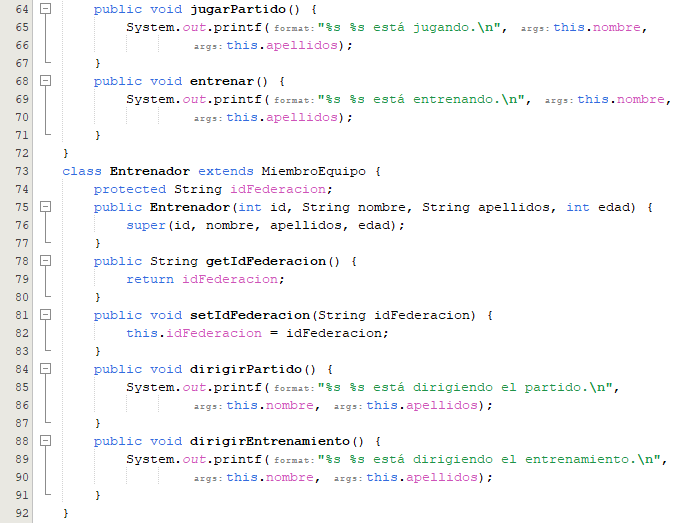
Archivo MiembroEquipo.java

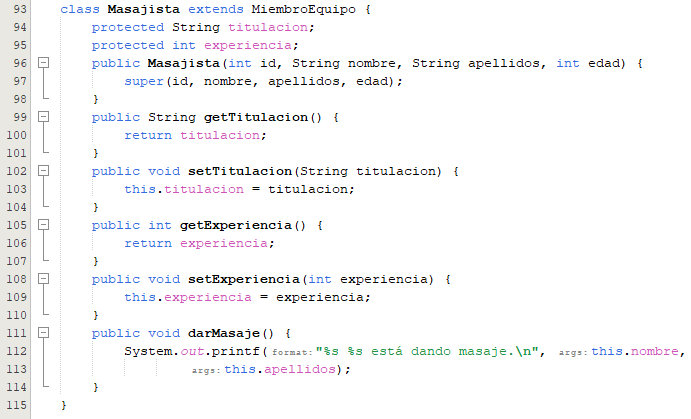


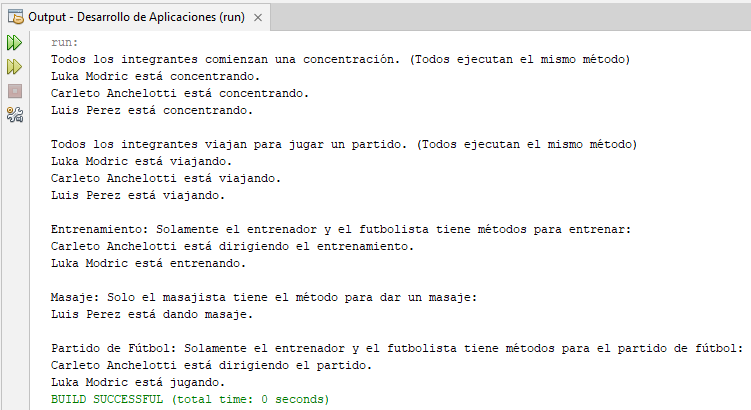
Archivo NewMain5.java





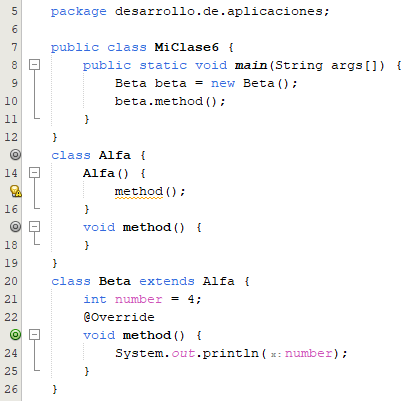






**Ejercicio 10**. Desarrollar una aplicación que aplique la herencia para calcular el área y perímetro de una figura geométrica. Las figuras geométricas son rectángulo, cuadrado, triangulo y círculo. Utilizar todos los conceptos revisados para hacer el código mas óptimo posible.

**Ejercicio 11**. ¿Cuál es el resultado de la ejecución del siguiente código?



1. 4
2. 0
3. 0

4

1. 4

0

1. Ninguna de las anteriores.