**Actividad 1.**

**1. Crear una clase llamada Círculo la cual contiene:**

a. Dos variables de instancia privada: radio (de tipo double) y color (de tipo String), con valor por defecto de 12.5 y "azul", respectivamente.

b. Dos constructores sobrecargados (overloaded): un constructor predeterminado o por default sin argumentos y otro constructor que tiene un argumento de input double para el valor del radio.

c. Dos métodos públicos: getRadio() y getArea(), que devuelven el radio y el área de esta instancia, respectivamente. Para calcular el área de un círculo, deberás seguir la siguiente fórmula: area = radius\*radius\*Math.PI



**2. Compilar y ejecutar la clase "Circulo.java". ¿Puedes ejecutar la clase “Circulo.java”? ¿Por**

**qué?**

La clase Circulo.java no se puede ejecutar, ya que no incluye el método main para hacer pruebas, lo que nos da la clase circulo es lo necesario para poder hacer pruebas en otra clase del mismo paquete con el método main para hacer pruebas creando objetos de la clase Circulo y para ejecutar los métodos de la clase Circulo en base al objeto de la clase Circulo que creemos en el main.

**3. Crear una clase CirculoTest, esta clase contendrá el método main() para poder visualizar**

**los siguientes datos:**

Utilizando un constructor por defecto para imprimir:

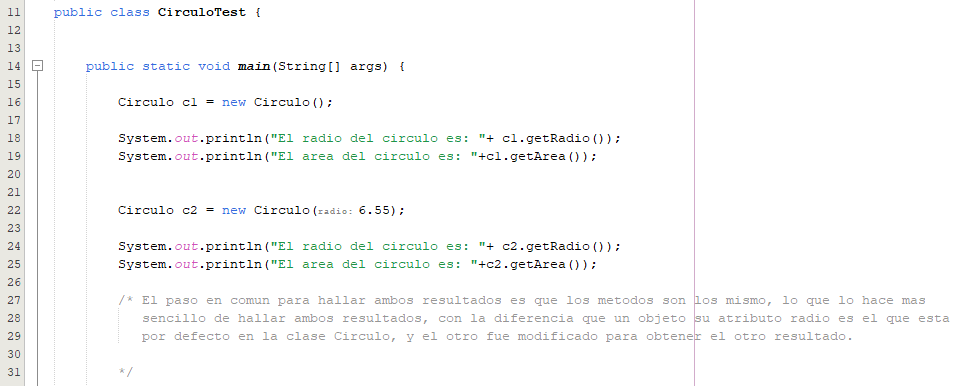
a. El radio de un círculo.

b. El área de un círculo.

Utilizando un constructor sobrecargado, donde radio es igual a 6.55, para imprimir:

a. El radio de un círculo.

b. El área de un círculo.



**¿Qué pasos en común debe realizar para ejecutar cada uno de los ejercicios? Describir**

**conceptos del paradigma orientada a objetos.**

El paso en común para hallar ambos resultados es que los métodos son los mismo, lo que lo hace más sencillo de hallar ambos resultados, con la diferencia que un objeto su atributo radio es el que esta por defecto en la clase Circulo, y el otro fue modificado para obtener el otro resultado.

**4. Ejecutar la clase “CirculoTest.java” y responde las siguientes preguntas:**

**a. ¿Qué es el constructor y para qué sirve?**

El constructor es la herramienta principal en la cual crearemos objetos, ya que al crear un objeto según los parámetros que queramos estamos llamando a un constructor.

**b. ¿Con qué método recupero el radio del círculo?**

Para obtener el radio, que es un atributo de la clase Circulo, se usan los getters, que deben ser declarados en la clase circulo para poder llamarlos en otra clase.

**c. ¿Por qué los atributos son privados y los métodos son públicos?**

Los atributos son privados ya que estamos aplicando encapsulamiento, para darle mayor seguridad a los atributos para que solo puedan ser accedidos a ellos por los getters y setters.

Los metodos son públicos, ya que podremos llamarlos en otras clases del mismo paquete, cuando creamos objetos nos sirve para poder aplicar los métodos en ese objeto que creemos.

**d. ¿Qué pasa si accedes directamente a los atributos del círculo desde el método**

**main()?**

Si son públicos no habría problemas, pero si son privados deben ser accedidos mediante getters y setters, ya que sino habrá errores.

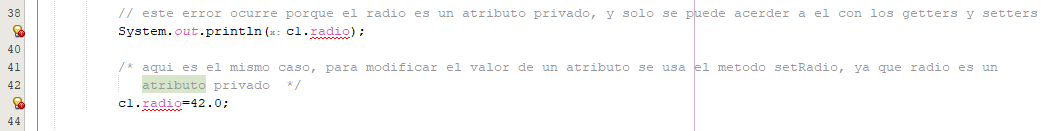
**e. ¿Qué pasa si modificas directamente el atributo área del círculo desde el método**

**main()?**

En el caso de este ejercicio, al ser atributos privados el radio, la forma de modificar su valor para que cuando se haga la función calcularArea() de un valor diferente es usar los setters, que para radio seria setRadio(), dentro de los paréntesis podríamos el nuevo valor para el atributo radio de la clase Circulo, pero si eso se hace directamente habrá errores.

**f. En la clase “CirculoTest.java”, ejecuta la siguiente instrucción:**

**System.out.println(c1.radio), donde c1 es una instancia del objeto Círculo. ¿Qué ocurrió? Explique el mensaje de error.**



Se esta queriendo acceder a ellos directamente y como radio es un atributo privado no se puede, solo se puede obtener su valor con los getters.

**g. En la clase “CirculoTest.java”, ejecuta la siguiente instrucción: c1.radio=42.0, donde**

**c1 es una instancia del objeto Círculo. ¿Qué ocurrió? Explique el mensaje de error.**

Se esta queriendo acceder a ellos directamente y como radio es un atributo privado no se puede, solo se puede modificar su valor con los setters.

**h. ¿Con qué método modificas el valor de un atributo de la clase “Circulo.java”?**

con el método setRadio() para el atributo radio y con el setColor() para el atributo color, dentro de los paréntesis se debe poder el valor al cual va cambiar, en caso del radio se debe poner otro Double, y en caso de color se debe poder otro String porque son los tipos de variable de esos atributos,

**i. Crear un objeto de tipo Círculo y modifica los valores por defecto.**

**j. ¿Para qué sirve la palabra reservada “this”? Modifique el constructor con argumentos (radio y color) y utiliza “this” y métodos setters en la clase “Circulo.java”.**

this es como que señalara al valor del atributo, en los metodos setter que su función es cambiar el valor de un atributo utiliza el this

ej: public double setRadio(10.00){

this.radio = radio;

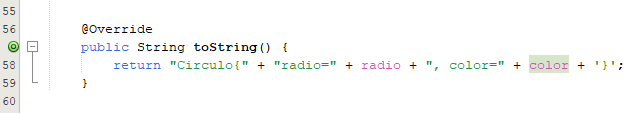
]

this.radio representa el valor del atributo, radio representa el valor que queremos que ahora tenga ese atributo, asi que lo que hace un setter es que el this.radio adquiera el valor que queremos que ahora tenga el atributo, radio es el parámetro que en ese ejemplo es 10.00, ya que al aplicar ese metodo debemos especificar a que valor queremos que cambie, eso con el fin de que this.radio tome ese valor.

**k. ¿Para qué sirve el método toString()?**

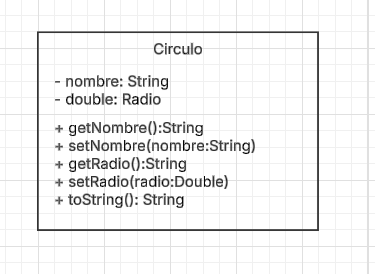
Retorna el nombre de la Clase con todos los valores de los atributos en orden, sirve para que nos dé una descripción de la clase del objeto que aplico este método.

**l. Incluir el método toString() en la clase “Circulo.java”.**



**5. Trabaja en el diagrama de clase para la clase Circulo utilizando la herramienta online Visio**

**o PlantUML (**[**https://plantuml.com/es/class-diagram**](https://plantuml.com/es/class-diagram)**).**

****

**6. Finalmente, subir este ejercicio a su repositorio GitHub con el siguiente nombre POOIIGRUPOX-S1-1, este repositorio debe ser público. Enviar el repositorio de esta primera a actividad.**

https://github.com/Josephtellolaiza/POOII-GRUPOXS1-2.git

**Actividad 2.**

**1. Se tiene las siguientes clases:**

• Crear la clase Punto.

• Crear la clase Triángulo.

• Crear la clase Figura Geométrica. Completar con 2 métodos abstractos: calcular el

área y determinar si es una figura geométrica regular o no. Adicionalmente,

incorporar métodos de accesos y manipulación.

• Crear la clase Superficie Plana. Esta clase tendrá un método que devolverá las áreas

de cada figura geométrica. Completarlo.

• Crear una clase genérica para invocar a todas las clases mencionadas. Esta clase

genérica debe contener el método main().

**2. Para la clase Punto considerar:**

a. El método toString()devolverá lo siguiente: "El punto tiene las siguientes

coordenadas: " + x + "," + y .

b. Un constructor sin parámetros y otro con parámetros (coordenaX, coordenaY).

c. Incluir todos sus métodos de acceso y manipulación.

d. La clase Punto tiene un método calcular distancia que se sobreescriben, el que no

cuenta con parámetros retornará el resultado del otro método con parámetro.

**3. Para la clase Triángulo considerar:**

a. El método toString() devolverá lo siguiente: "Triángulo: " + getNombre() + "

tiene 3 Puntos: "+ getPunto1().toString() + “,” + getPunto2().toString()+ “,” +

getPunto3().toString();

b. Un constructor sin parámetros y otro con parámetros pero éste debe invocar al constructor de la superclase.

c. Incluir todos sus métodos de acceso y manipulación.

d. Sobreescribir los métodos hereados de la clase Figura Geométrica.

**4. Identificar las clases que heredan, ¿qué atributos y métodos heredan?**

La clase Triangulo hereda de la clase FiguraGeometrica, sus métodos abstractos para luego usar polimorfismo para usarlos a su manera.

**5. Identificar las clases que conforman una composición.**

La Clase Punto con la Clase Triangulo, ya que si no existe la Clase Punto tampoco existiera la del Triangulo al tener atributos objetos de la Clase Punto

**6. ¿Qué es una superclase y una subclase?**

Una super clase es una clase que hereda algunos o todos de sus atributos a subclases, también heredan sus metodos, las subclases son clases que reciben atributos de la superclase, las subclases pueden tener atributos propios también, además de poder heradarlos también a otras clases.

**7. ¿Por qué usamos abstract? ¿Se puede dejar de heredar un método de una clase abstracta?**

Usamos abstract para declarar métodos que se sobrescribirán en otras clases.

**8. ¿Qué anotación utilizo para sobreescribir métodos?**

@override, señala el uso de polimorfismo para que el método se adapte a lo que se quiera hallar en la clase en la que se sobreescriba.

**9. Los atributos de la clase Figura Geométrica conviértalas en protected. ¿En qué condición convierte a los atributos? ¿Es posible acceder a los atributos protegidos sin utilizar una invocación a super() o sin método get?.**

Los atributos protected solo pueden ser accedidos a ellos por las clases hijas, si pueden ser accedidos pero si es en una clase hija.

**10. ¿Cómo aplicarías polimorfismo?**

El polimorfismo es sobrescribir uno o varios métodos, aunque los métodos se llamen igual, depende de que en que objeto de clase se aplique el método, un ejemplo en el ejercicio es que en la clase abstracta se declararon métodos pero esos métodos no hacían nada, estaban para que la clase hija Triangulo la sobrescribiera y la use de acuerdo a sus necesidades.

**11.Finalmente, subir este ejercicio a su repositorio GitHub con el siguiente nombre POOII-GRUPOXS1-2, este repositorio debe ser público. Enviar el repositorio de esta primera a actividad.**

<https://github.com/Josephtellolaiza/POOII-GRUPOXS1-2.git>