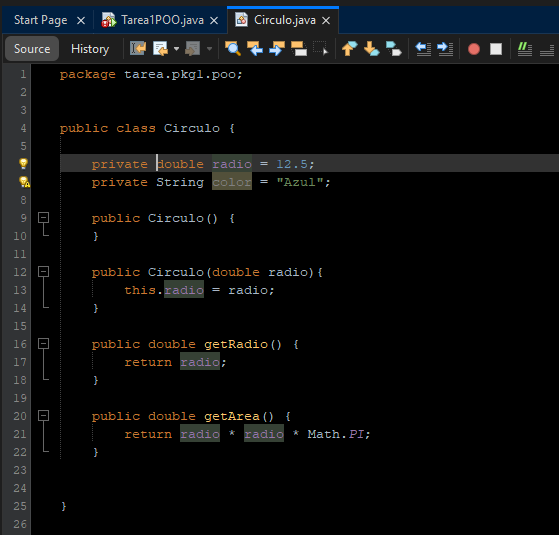
# Actividad de Aprendizaje Semana 01

**Indicaciones Generales:**

1. El plagio se sanciona con la suspensión o expulsión del estudiante de la Universidad. Reglamento General de Estudios.
2. Sea cuidadoso con su redacción, la cual formará parte de su calificación (aspectos léxicos, sintácticos, semánticos).
3. Lea bien la pregunta o enunciado antes de proceder a su desarrollo, administre su tiempo eficazmente.

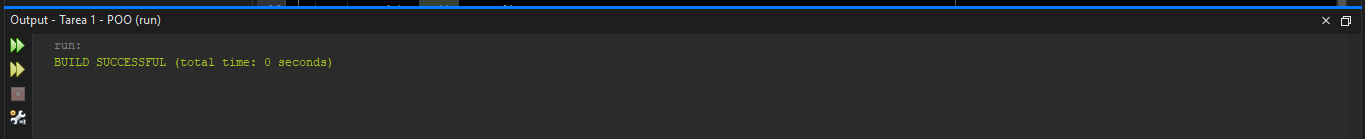
**Actividad 1.**

* 1. Crear una clase llamada Círculo la cual contiene:
     1. Dos variables de instancia privada: radio (de tipo double) y color (de tipo String), con valor por defecto de 12.5 y "azul", respectivamente.
     2. 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.
     3. 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



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

El código fuente de la clase si compila, pero no es ejecutable, puesto que para que el código sea ejecutable se deben instanciar los objetos y llamar a los métodos en una clase que tenga el método main().



* 1. Crear una clase CirculoTest, esta clase contendrá el método *main*() para poder visualizar los siguientes datos:

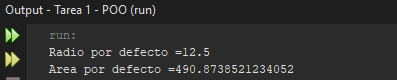
Utilizando un constructor por defecto para imprimir:

* + 1. El radio de un círculo.
    2. El área de un círculo.

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

* + 1. El radio de un círculo.
    2. 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.

Para cada uno de los ejercicios en primer lugar se debe instanciar el objeto de la clase Circulo. La diferencia es en la forma de cómo obtener los datos, en el caso del constructor por defecto, no tiene parámetros por lo cual coje los valores por defecto del atributo radio el cual es 12.5 y en el método getArea() trabaja con este valor por defecto.

En el segundo ejercicio nos pide instanciar el objeto circulo con el valor de 6.55. En este ejercicio la diferencia es que estamos trabajando con el constructor por argumentos ya este nos permite agregar valores a diferencia de un constructor por defecto.

1. Ejecutar la clase “CirculoTest.java” y responde las siguientes preguntas:
   * 1. ¿Qué es el constructor y para qué sirve?

Es un método sin retorno de datos el cual inicializa los atributos de un objeto.

* + 1. ¿Con qué método recupero el radio del círculo?

Con el método getRadio().

* + 1. ¿Por qué los atributos son privados y los métodos son públicos?

Por que los atributos son de uso exclusivo de la clase, en cambio los métodos realizan su lógica utilizando los atributos, pero son públicos para que puedan ser llamados en otras clases.

* + 1. ¿Qué pasa si accedes directamente a los atributos del círculo desde el método *main*()?

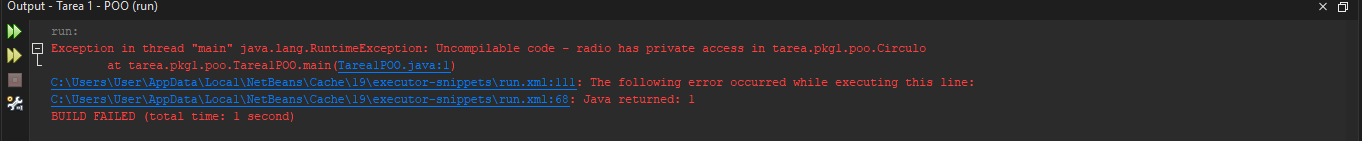
Para que pase eso el atributo no debería estar encapsulado, pero lo que ocasionaría es que cualquier clase pueda modificar los atributos y pierdan su lógica.

* + 1. ¿Qué pasa si modificas directamente el atributo área del círculo desde el método *main*()?

De igual manera que la pregunta anterior, esto ocasionaría fallos en la lógica de los atributos, puesto que desde otra clase se puede definir valores para el área que no tienen nada que ver, por ejemplo, se puede asignar un valor negativo al área, siendo esto ilógico.

* + 1. 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.

Lo que ocurre aquí es un error de modificador de acceso del atributo, no podemos acceder al valor del radio puesto que el atributo esta en modo privado.

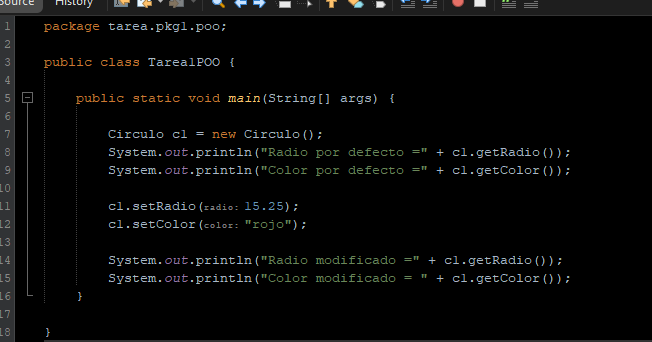
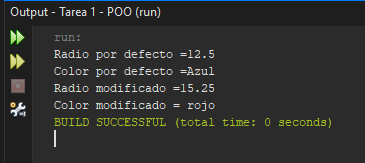


* + 1. En la clase “CirculoTest.java”, ejecuta la siguiente instrucción: c1.radio=42.0, donde c1 es una instancia del objeto Círculo. ¿Qué ocurrió? Explique el mensaje de error.

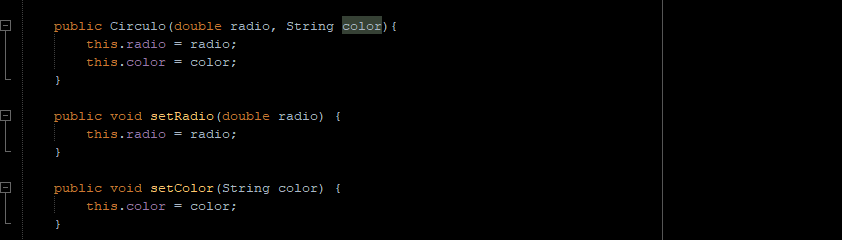
No se puede modificar o asignar este valor, puesto que el atributo radio es privado.

* + 1. ¿Con qué método modificas el valor de un atributo de la clase “Circulo.java”?

Con el método set() se pueden modificar los valores de un atributo, este método puede ser llamado desde cualquier otra clase si es que su modificador de acceso es publico.

* + 1. Crear un objeto de tipo Círculo y modifica los valores por defecto.
    2. ¿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”.

Se utiliza para referirse al objeto actual.

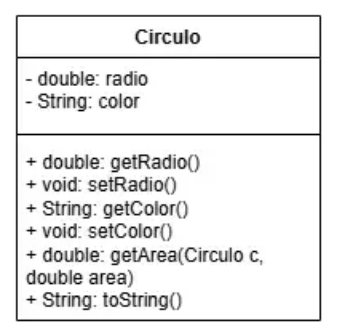


* + 1. ¿Para qué sirve el método toString()?

Sirve para mostrar todos los valores de los atributos de una clase.

* + 1. Incluir el método toString() en la clase “Circulo.java”.

1. Trabaja en el diagrama de clase para la clase Circulo utilizando la herramienta online Visio o PlantUML (https://plantuml.com/es/class-diagram).



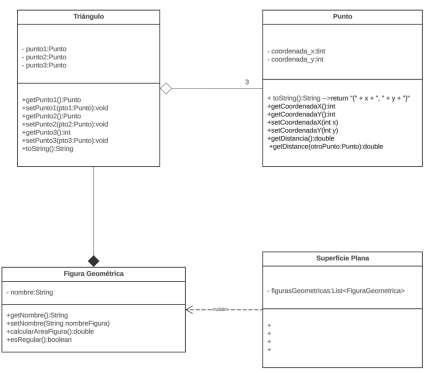
1. Finalmente, subir este ejercicio a su repositorio GitHub con el siguiente nombre POOIIGRUPOX-S1-1, este repositorio debe ser público. Enviar el repositorio de esta primera a actividad.

<https://github.com/Robodepredador/s01-Poo2>

|  |  |
| --- | --- |
|  | **Fecha límite de presentación: En clase**  **Integrante: 3**  **Tiempo de resolución 30min** |

**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().



1. Para la clase Punto considerar:
   1. El método toString()devolverá lo siguiente: "El punto tiene las siguientes coordenadas: " + x + "," + y .
   2. Un constructor sin parámetros y otro con parámetros (coordenaX, coordenaY).
   3. Incluir todos sus métodos de acceso y manipulación.
   4. 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.

1. Para la clase Triángulo considerar:
   1. El método toString() devolverá lo siguiente: "Triángulo: " + getNombre() + " tiene 3 Puntos: "+ getPunto1().toString() + “,” + getPunto2().toString()+ “,” + getPunto3().toString();
   2. Un constructor sin parámetros y otro con parámetros pero éste debe invocar al constructor de la superclase.
   3. Incluir todos sus métodos de acceso y manipulación.
   4. Sobreescribir los métodos hereados de la clase Figura Geométrica.

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

La clase triangulo hereda los atributos de la clase FiguraGeometrica. Esta clase hereda el atributo nombre y los métodos que hereda son los getNombre() y setNombre().

1. Identificar las clases que conforman una composición.

Las clases que tienen composición es FiguraGeometrica con Triangulo, en donde Triangulo es una Figura Geometrica.

1. ¿Qué es una superclase y una subclase?

Una superclase es una clase “padre” que es una clase general y la super clase es una clase “hija” la cual hereda atributos y métodos según conveniencia.

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

Para poder heredar métodos a una subclase, si se puede heredar métodos de una super clase abstracta.

1. ¿Qué anotación utilizo para sobreescribir métodos?

@override

1. 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 son accesibles desde la misma clase, subclases y clases del mismo paquete. Si es posible acceder a los atributos protegidos desde sub clases.

1. ¿Cómo aplicarías polimorfismo?

En este ejercicio se usa cuando hacemos la composición (List<Figurasgeométricas> puede contener Triángulos ), es decir cuando tratamos super clases y sub clases.

1. 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/Robodepredador/s01-Poo2>

|  |  |
| --- | --- |
|  | **Fecha límite de presentación: En clase Integrantes: 3 estudiantes.**  **Tiempo de resolución 30min** |