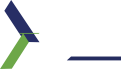
Actividad de Aprendizaje Semana 01



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

qué?

* Si, se puede compilar y ejecutar la clase Circulo.java, pero no se visualiza nada porque el main está en CirculoTest
  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.

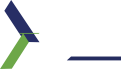
* Primero se crea la instancia de la clase Circulo utilizando los constructores, la única diferencia es que en el constructor sobrecargado es necesario enviar un parámetro. Luego, utilizamos los métodos creados en la clase Circulo para obtener el radio y el área.
  1. Ejecutar la clase “CirculoTest.java” y responde las siguientes preguntas:
     1. **¿Qué es el constructor y para qué sirve?**
* El constructor es lo que permite crear una instancia de una clase
  + 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?**
* Para garantizar el encapsulamiento
  + 1. **¿Qué pasa si accedes directamente a los atributos del círculo desde el método**

***main*()?**

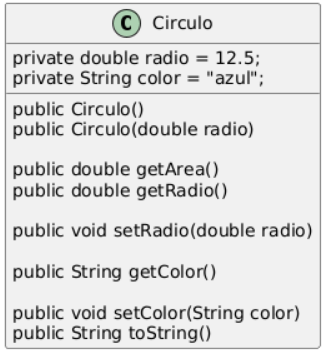
* No te va a permitir acceder ya que están declarados como atributos privados
  + 1. **¿Qué pasa si modificas directamente el atributo área del círculo desde el método**

***main*()?**

* No será posible ya que área no es un atributo



* + 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.**
* No mostrará el radio debido a que el atributo radio es 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.**
* Generará un error debido a que no se puede modificar un atributo privado directamente
  + 1. **¿Con qué método modificas el valor de un atributo de la clase “Circulo.java”?**
* Mediante el uso de setters como setRadio() o setColor()
  + 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”.**
* Sirve para referenciar la instancia actual y que no se confunda con el parámetro del mismo nombre
  + 1. **¿Para qué sirve el método toString()?**
* Para representar el objeto como una cadena de texto
  + 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 POOII- GRUPOX-S1-1, este repositorio debe ser público. Enviar el repositorio de esta primera a actividad.

https://github.com/SebastianTapiaGarcia/POO2

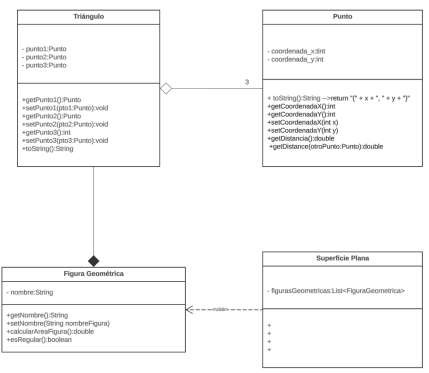
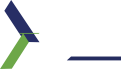


**Fecha límite de presentación: En clase Integrante: 3**

**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:
2. El método toString()devolverá lo siguiente: "El punto tiene las siguientes coordenadas: " + x + "," + y .
3. Un constructor sin parámetros y otro con parámetros (coordenaX, coordenaY).
4. Incluir todos sus métodos de acceso y manipulación.
5. 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.
6. Para la clase Triángulo considerar:
7. El método toString() devolverá lo siguiente: "Triángulo: " + getNombre() + " tiene 3 Puntos: "+ getPunto1().toString() + “,” + getPunto2().toString()+ “,” + getPunto3().toString();
8. Un constructor sin parámetros y otro con parámetros pero éste debe invocar al constructor de la superclase.
9. Incluir todos sus métodos de acceso y manipulación.
10. Sobreescribir los métodos hereados de la clase Figura Geométrica.
11. Identificar las clases que heredan, ¿qué atributos y métodos heredan?

* La clase Triangulo hereda de la clase FiguraGeometrica el atributo nombre, los métodos getNombre(), setNombre(), calcularAreaFigura() y esRegular()

1. Identificar las clases que conforman una composición.

* La clase triangulo tiene una composición con la clase Punto ya que un triangulo siempre debe tener 3 puntos
* La clase SuperficiePlana tiene una composición con FiguraGeometrica ya que contiene una lista de figuras geometricas

1. ¿Qué es una superclase y una subclase?

* Superclase es una clase que proporciona atributos y métodos a otra clase mediante la herencia
* Subclase es una clase que hereda atributos y/o métodos de otra

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

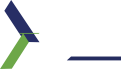
* Se usa abstract ára definir que métodos queremos que hereden las subclases pero sin desarrollarlos ya que cada figura los calculará de forma diferente
* No se puede dejar de heredar un método de una clase abstracta a menos que la subclase también sea abstracta

1. ¿Qué anotación utilizo para sobreescribir métodos?

* Se usa la anotación @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?.

* Al convertir nombre en protected hace que las subclases puedan acceder directamente a él sin la necesidad de los getters y setters.



1. ¿Cómo aplicarías polimorfismo?

* El polimorfismo se aplica en el método calcularAreaFigura(), ya que cualquier figura puede llamar a ese método y desarrollarlo de manera distinta.

1. Finalmente, subir este ejercicio a su repositorio GitHub con el siguiente nombre POOII-GRUPOX- S1-2, este repositorio debe ser público. Enviar el repositoriode esta primera a actividad.