**Icono

El contenido generado por IA puede ser incorrecto.**

**S01 – Laboratorio (POO)**

**Curso:**

**Programación Orientada a Objetos II**

**Docente:**

**Alfaro Gutierrez, Gianny Romie**

**Bloque:**

**FC-PREAGRO07A01N**

**Integrante:**

**Agramonte Sanchez, José Fernando**

**Lima - Perú**

**2025-1**

**Solo preguntas - Actividad 1**

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

No se puede ejecutar la clase Circulo porque no tiene un main() como método.

3. ¿Qué pasos en común debe realizar para ejecutar cada uno de los ejercicios? Describir conceptos del paradigma orientada a objetos.

Se creó clases .java para ingresar el código y se usó un compilador de java para ejecutar el código.

En la clase Circulo, se usaron dos constructores para inicializar los atributos del círculo y se crearon los métodos getArea y getRadio.

En la clase CirculoTest, se usaron los dos constructores para crear dos círculos y los métodos de getArea y getRadio fueron mostrados.

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

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

El constructor sirve para poder inicializar los atributos de los objetos antes de crearlos.

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

En este caso, para recuperar el radio se debe usar el nombre del objeto creado y luego un “.getRadio()”.

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

Para esto se usó encapsulamiento, los atributos son privados para que no se puedan modificar después de haberlos inicializados y los métodos son públicos para en este caso, la clase CirculoTest pueda acceder sin problema y mostrarlos en el main.

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

En mi caso, si quiero acceder directamente al radio, me saldrá un error y me pedirá que elimine la palabra “private” y solo quede “double radio”.

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

Saldrá un error porque el área no es un atributo sino un valor que se calcula, si se quiere modificar se debe crear el método “setRadio()” para darle un nuevo valor.

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.

Saldrá un error, debido a que solo se puede acceder al radio en la clase Circulo, para poder mostrar el radio se tendría que modificar a ” System.out.println(c1.getRadio());”.

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.

El error es principalmente porque se debe crear un método “setRadio()” para poder modificar el radio directamente de la clase Circulo.

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

Con el método “set” se puede modificar un atributo que esté privado en otra clase. Aunque, si el atributo fuese protected, se podría modificar sin tener que usar un set, siempre y cuando sea una subclase o esté en el mismo package.

j. ¿Para qué sirve la palabra reservada “this”?

La palabra “this” sirve para que Java pueda diferenciar las variables que se llaman igual. Por ejemplo, en el caso de “this.radio = radio”, significa que al radio del objeto que vamos a crear, le asignamos el radio que pasamos como parámetro.

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

Sirve para poder ver el estado del objeto o la información del objeto, si no usáramos el “toString()” y escríbimos, por ejemplo, “System.out.println(circulo1);”, solo veríamos la información de “circulo1” que sería la referencia hacia los objetos.

**Solo preguntas - Actividad 2**

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

La clase triángulo hereda: Métodos de FiguraGeometrica, calcularAreaFigura(), esRegular(), getNombre() y setNombre(). Y el nombre de FiguraGeometrica como atributo.

5. Identificar las clases que conforman una composición.

La clase de Triangulo no podría existir sin la clase Punto porque un triángulo necesita si o si tres vértices, cuando se destruye un triángulo, sus puntos dejan de tener sentido.

6. ¿Qué es una superclase y una subclase?

Una superclase es la clase principal en donde otras clases heredan sus atributos o métodos y las subclases son las clases que heredan de la superclase. En este caso, la clase Triangulo seria la subclase y FiguraGeometrica sería la superclase.

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

Se usa para no poder instanciar una clase y que otras clases heredan de ella. No se puede dejar de heredar un método abstract de una clase abstracta. Si una subclase hereda de una clase abstracta, debe implementar todos los métodos abstractos o convertirse en clase abstracta.

8. ¿Qué anotación utilizo para sobreescribir métodos?

Para sobreescribir métodos en Java, se utiliza @override. Se usa principalmente para evitar errores de escritura, para ver la información del objeto y tener una mejor legibilidad del código.

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?.

Cuando se cambian los atributos de private a protected, solo se pueden acceder desde las subclases y no son accesibles desde clases externas. Si, las subclases pueden acceder directamente a los atributos protected sin necesidad de invocar a super() ni utilizar el get, por ejemplo, en el método mostrarNombre(), nombre accede directamente en la subclase Triangulo sin super() ni getNombre().

10. ¿Cómo aplicarías polimorfismo?

Se podría usar en la lista de figuras geométricas, donde almacena diferentes tipos de figuras. En la clase SuperficiePlana, el método mostrarAreas(), muestra el área sin importar la figura que sea.