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

La clase triángulo hereda la clase FiguraGeometrica, y por lo tanto, hereda el atributo de nombre los métodos getNombre, setNombre, calcularAreaFigura y esRegular.

5. Identificar las clases que conforman una composición.

La clase FiguraGeometrica se compone de Triangulo, es decir que la FiguraGeometrica no puede existir sin un Triangulo, son dependientes entre sí.

6. ¿Qué es una superclase y una subclase?

Cada clase solo puede tener una clase padre, a esta se le denomina superclase; la clase hija de una superclase es una subclase.

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

Usamos una clase o métodos abstractos para declarar estos métodos en una superclase sin la necesidad de implementarlo, y que sus hijos estén obligados a tener su propia implementación de estos métodos. No, una subclase está obligada a heredar los métodos abstractos de su padre.

8. ¿Qué anotación utilizo para sobreescribir métodos?

@Override

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 cambiamos el encapsulamiento a protected, los atributos pasan a ser solo reconocibles por las clases que estén en el mismo paquete. En este caso las demás clases pueden acceder a sus atributos protegidos porque están todas en el mismo paquete Figura.

10. ¿Cómo aplicarías polimorfismo?

Usaría polimorfismo para la clase FiguraGeometrica y sus hijos, ya que pueden ser cualquier figura (Rectángulo, cuadrado, etc.), cada figura tiene puntos, pero diferente forma de calcular su área y determinar si es regular, con el polimorfismo podemos lograr que cada clase hija tenga su propia implementación de estos métodos.