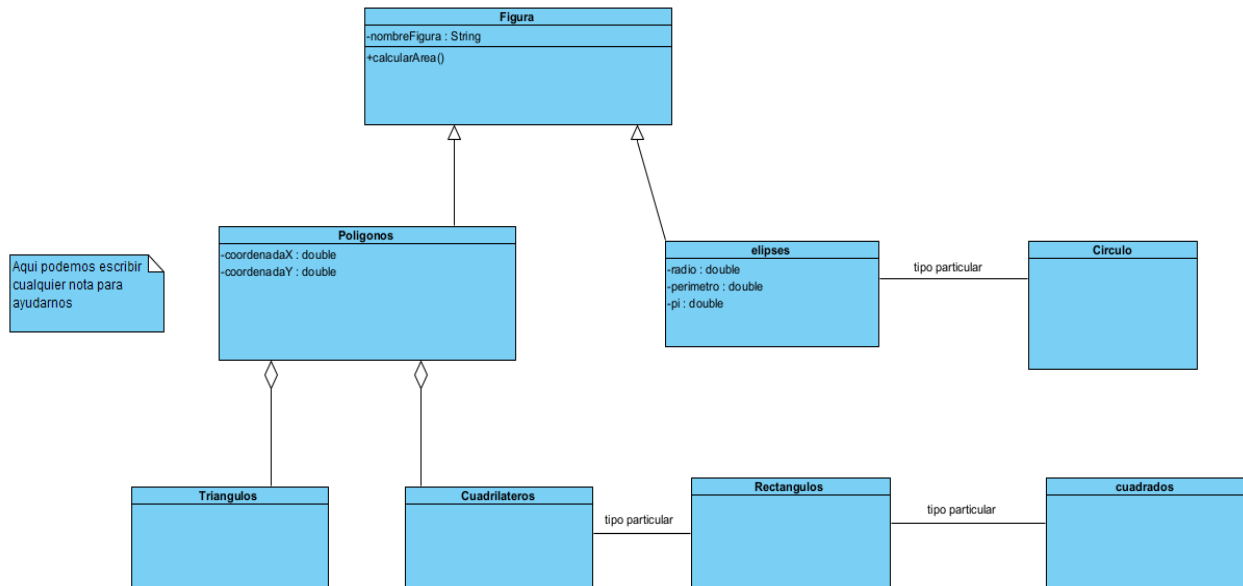


ED05

ANTONIO JIMÉNEZ SEVILLA

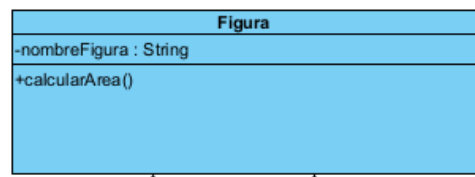


Los diagramas de clases son uno de los tipos de diagramas más útiles en UML, ya que trazan claramente la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos.

En un diagrama de clases podemos encontrar los siguientes elementos: **clases, relaciones, notas, elementos de agrupación.**

Empecemos por las clases:

En la imagen de la derecha vemos como se representa una clase. Se representa en el diagrama como un rectángulo dividido en tres filas. En la primera escribimos el nombre de la clase Figura si fuera abstracta la escribiríamos en cursiva.



Una instancia de esa clase sería el nombre de alguna figura, como cuadrado, circulo etc.. para identificarlos usaremos los **atributos**, que son un dato importante que contiene valores que describen cada instancia de esa clase. Se colocan en la segunda fila. En nuestro caso hemos puesto una cadena de texto String, para el nombre de la figura. También podríamos utilizar integer o double etc dependiendo del tipo de atributo que queramos utilizar para nuestro programa. Según los apuntes del curso se coloca primero la visibilidad el símbolo negativo para que sea privado (más adelante explico que significa) el nombreFigura dos puntos y el tipo de dato en este caso una cadena de texto” String”.

En la tercera fila es donde se colocan los métodos(se les conoce también como operaciones o funciones), estos permiten especificar cualquier característica de

comportamiento de una clase. En nuestra tarea no hace falta poner ningún método ya que se especifica así en el enunciado, pero yo he puesto el método `calcularArea`, así sabremos el área de cada figura. El formato para escribir el método es la visibilidad en este caso un símbolo `+` para decir que es público (lo explico más adelante), el nombre de la función o método, y paréntesis `()` para indicar la función que se va a programar.

Vamos a explicar que es la **visibilidad**, la cual define la accesibilidad para ese atributo o método.

Público: se usa el símbolo `+`. Se pueden acceder desde cualquier clase y cualquier parte del programa.

Privado: se usa el símbolo `-`. Ninguna otra clase o subclase puede acceder a ellos.

Protegido: se usa el símbolo `#`. Sólo se pueden acceder desde operaciones de la clase o de clases derivadas en cualquier nivel.

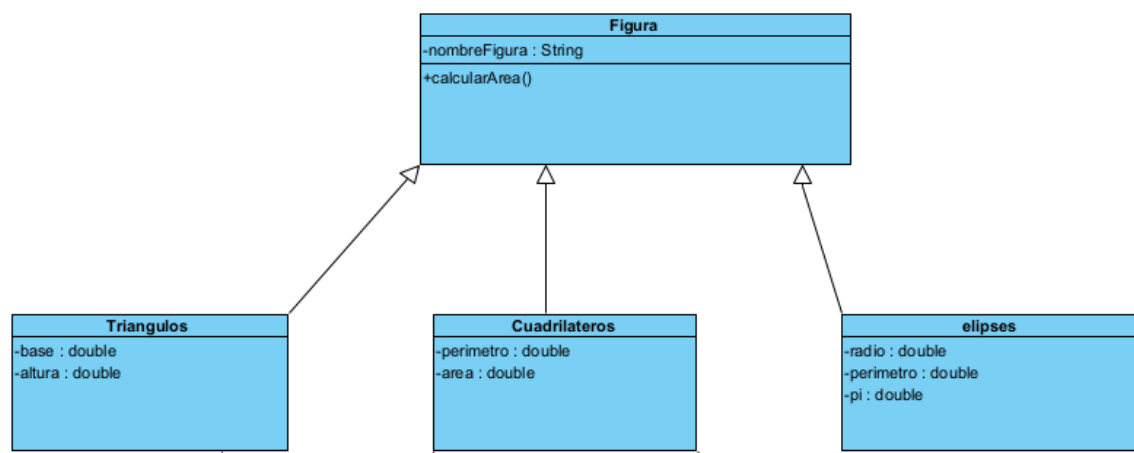
Paquete: se usa el símbolo `~`. Se puede acceder desde las operaciones de las clases que pertenecen al mismo paquete que la clase que estamos definiendo. Se usa cuando el lenguaje de implementación es Java.

Una vez explicado lo que es una clase, continuamos el ejercicio.

Relaciones: en el diagrama representan relaciones reales entre los elementos del sistema a los que hacen referencia las clases. Pueden ser de asociación, agregación y herencia.

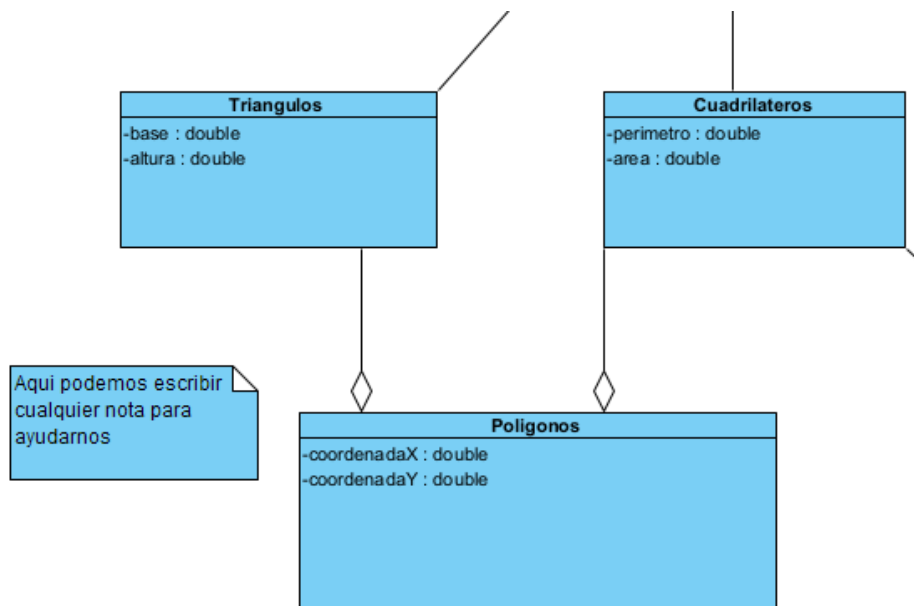
Hemos utilizado la herencia de forma que la clase **Figura** es la superclase(clase padre, clase base) y **triángulos**, **cuadriláteros** y **elipses** son las subclases(clases hijas, clase derivada). El objetivo principal de la herencia es la reutilización, poder utilizar código desarrollado con anterioridad. Los atributos y métodos de la superclase no se repiten en las subclases, salvo que se hayan redefinido.

En nuestro caso estamos hablando de herencia simple ya que viene solo de una superclase, clase padre o clase base.

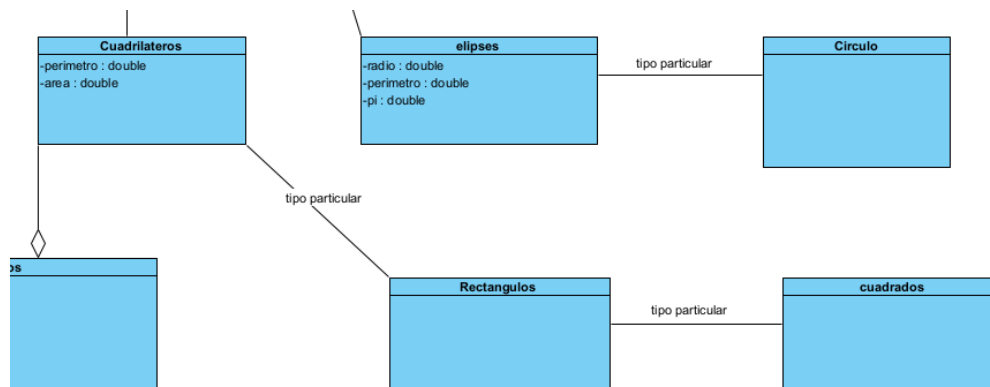


Continuamos explicando nuestro diagrama de clases. El enunciado dice que los **cuadriláteros** y los **triángulos** son tipos de **polígonos** para ello hemos utilizado la **agregación**. Que son una asociación binaria que representa una relación todo-parte (pertenece a, tiene un, es parte de). Los elementos parte pueden existir sin el elemento contenedor y no son propiedad suya. Esto significa que hemos creado una clase **Polígonos**.

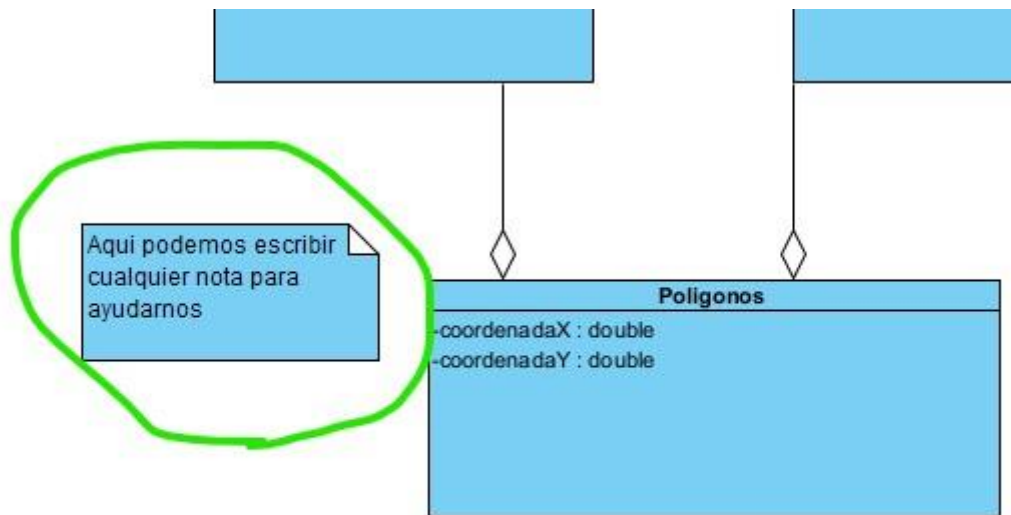
Un polígono es la región del plano limitada por tres o más segmentos, con lo que triángulos y cuadriláteros hexágonos etc... podrían estar en la clase polígonos pero no tiene por que ser así, un triángulo podría estar fuera de polígonos y seguir existiendo por su cuenta.



Otro tipo de relación que he utilizado en la tarea es **la asociación**, que es una relación estructural que describe una conexión entre objetos. En nuestro caso tenemos, un tipo particular de elipse es el círculo; mientras que un tipo particular de cuadrilátero es el rectángulo. A su vez, un tipo particular de rectángulo es el cuadrado.



En nuestro diagrama de clase también tenemos notas que se representan como un cuadro donde podemos escribir comentarios que nos ayuden a entender algún concepto que queramos representar.



Para terminar podríamos usar también **elementos de agrupación** que se utilizan cuando hay que modelar un sistema grande, entonces las clases y sus relaciones se agrupan en paquetes, que a su vez se relacionan entre sí. Pero al ser tan pequeño el programa no haría falta.