**Enunciado: clase Planeta**

Se requiere un programa que modele el concepto de un planeta del sistema solar. Un ***Planeta*** tiene los siguientes **atributos**:

▶ Un **nombre** de tipo *String* con valor inicial de *null*.

▶ **Cantidad de satélites** de tipo *int* con valor inicial de cero.

▶ **Masa en kilogramos** de tipo *double* con valor inicial de cero.

▶ **Volumen** en kilómetros cúbicos de tipo *double* con valor inicial de cero.

▶ **Diámetro** en kilómetros de tipo *int* con valor inicial de cero.

▶ **Distancia** media al Sol en millones de kilómetros, de tipo *int* con valor inicial de cero.

▶ Tipo de planeta de acuerdo con su tamaño, **de tipo enumerado** con los siguientes valores posibles: GASEOSO, TERRESTRE y ENANO.

▶ **Observable** a simple vista, de tipo booleano con valor inicial *false*.

La clase debe incluir los siguientes **métodos**:

▶ La clase debe tener un **constructor** que inicialice los valores de sus respectivos atributos.

▶ Definir un método que **imprima en pantalla** los valores de los atributos de un planeta.

▶ **Calcular la densidad** un planeta, como el cociente entre su masa y su volumen.

▶ **Determinar** si un planeta del sistema solar se considera exterior. Un planeta exterior está situado más allá del cinturón de asteroides. El cinturón de asteroides se encuentra entre 2.1 y 3.4 UA. Una unidad astronómica (UA) es la distancia entre la Tierra y el Sol= 149 597 870 Km.

▶ En un ***método main***se deben ***crear dos planetas y mostrar los valores de sus atributos en pantalla. Además, se debe imprimir la densidad de cada planeta y si el planeta es un planeta exterior del sistema solar.***

Diagrama de clases

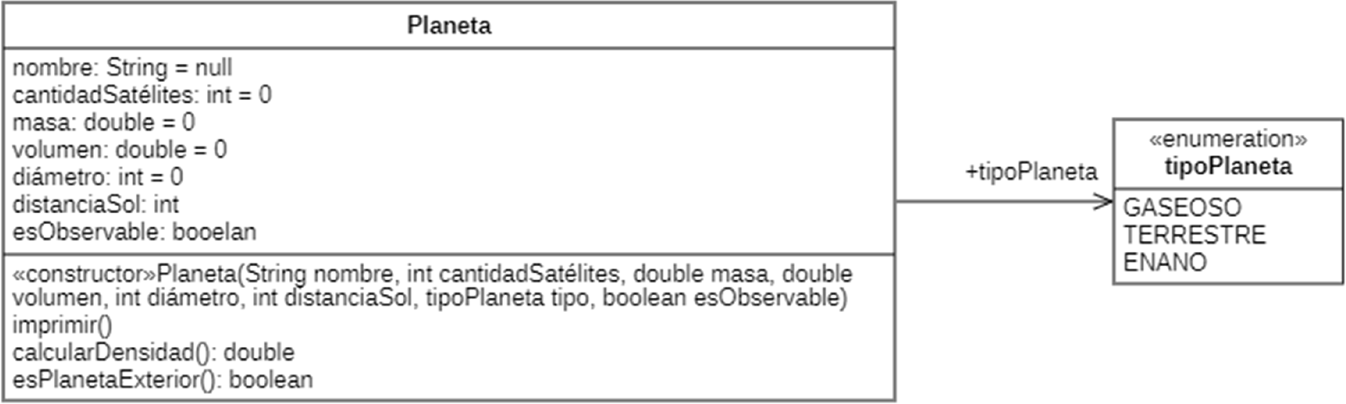


Diagrama de clases

Explicación del diagrama de clases

Se ha definido una clase denominada Planeta con sus respectivos atributos que representan el nombre del planeta, cantidad de satélites que tiene, su masa, volumen, diámetro, distancia al Sol y si es observable o no. Para cada atributo, además de su tipo, se ha agregado su correspondiente valor inicial o por defecto. También se han definido sus métodos respectivos: un constructor que inicializa los valores de sus atributos, un método para imprimir los valores de sus atributos en pantalla, otro para calcular la densidad (que devuelve un valor de tipo *double*) y un último método para determinar si es un planeta exterior (que devuelve un valor booleano).

El atributo para identificar el tipo de planeta se expresa como una asociación entre la clase Planeta y la clase TipoPlaneta cuyo nombre es el nombre del atributo. En UML, las variables *enum* se identifican con el este- reotipo <<enumeration>>, que representan un conjunto de valores constantes identificados como atributos en su segundo compartimiento.

Ejercicios propuestos

▶ Agregar dos atributos a la clase Planeta. El primero debe representar el periodo orbital del planeta (en años). El segundo atributo representa el periodo de rotación (en días).

▶ Modificar el constructor de la clase para que inicialice los valores de estos dos nuevos atributos.

▶ Modificar el método imprimir para que muestre en pantalla los valores de los nuevos atributos.

Enunciado: clases sobre figuras geométricas

Se requiere un programa que modele varias figuras geométricas: el círculo, el rectángulo, el cuadrado y el triángulo rectángulo.

▶ El círculo tiene como atributo su radio en centímetros.

▶ El rectángulo, su base y altura en centímetros.

▶ El cuadrado, la longitud de sus lados en centímetros.

▶ El triángulo, su base y altura en centímetros.

Se requieren métodos para determinar el área y el perímetro de cada figura geométrica. Además, para el triángulo rectángulo se requiere:

▶ Un método que calcule la hipotenusa del rectángulo.

▶ Un método para determinar qué tipo de triángulo es:

Equilátero: todos sus lados son iguales. Isósceles: tiene dos lados iguales.

Escaleno: todos sus lados son diferentes.

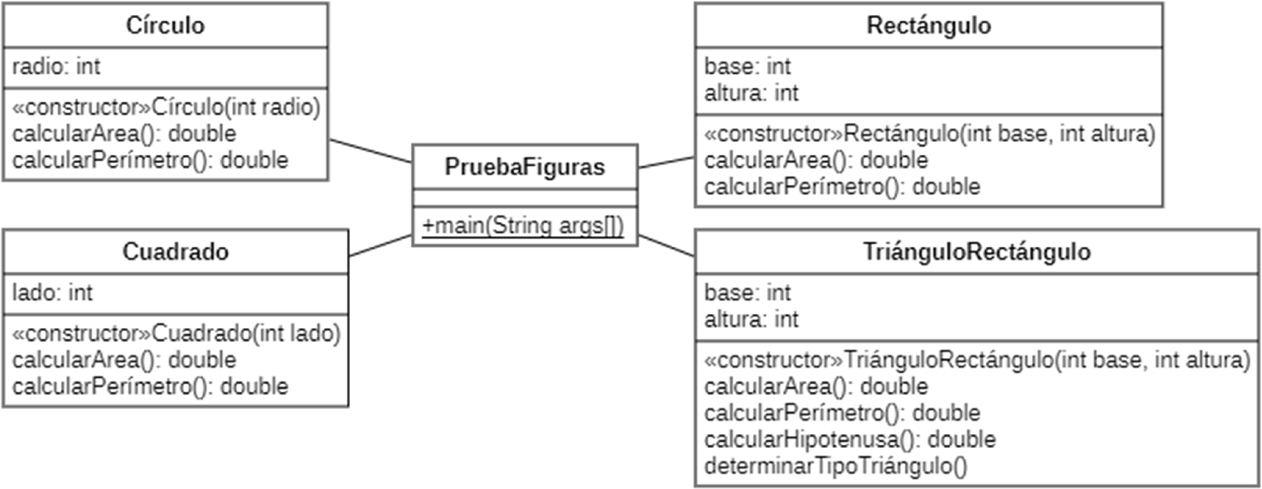
Se debe desarrollar una clase de prueba con un método *main* para crear las cuatro figuras y probar los métodos respectivos.

Instrucciones Java del ejercicio

**Tabla 2.4.** Instrucciones Java del ejercicio 2.4.

|  |  |  |
| --- | --- | --- |
| Instrucción | Descripción | Formato |
| ***Math.PI*** | Un valor *double* que representa la variable matemática pi = 3.1416, la relación entre el radio de la circunferencia y su diámetro. | Math.PI |
| ***Math.pow*** | Método que devuelve el valor del primer argumento elevado a la potencia del segundo argumento. | *Math.pow*(argumento1, argumento2); |
| && | Operador lógico AND. Devuelve solamente un valor verdadero cuando las dos expresiones son verdaderas, en caso contrario devuelve falso. | (expresión) &&  (expresión2) |

Diagrama de clases



**Figura 2.10.** Diagrama de clases del ejercicio 2.4.

Explicación del diagrama de clases

Se ha definido una clase para cada figura geométrica (Círculo, Cuadrado, Rectángulo y TriánguloRectángulo). Cada clase contiene sus atributos respectivos. La clase Círculo tiene su radio como atributo (de tipo *int*). La clase Cuadrado tiene la longitud de sus lados como atributo (de tipo *int*). La clase Rectángulo tiene como atributos su base y altura (ambos de tipo *int*). La clase TriánguloRectángulo, su base y altura (ambos de tipo *int*). La hipotenusa no es atributo, en este caso será un valor que será calculado por medio de un método.

Todas las clases poseen los mismos métodos: un constructor y métodos para calcular el área y calcular el perímetro que devuelven valores de tipo *double*. La clase TriánguloRectángulo tiene dos métodos adicionales para calcular el valor de la hipotenusa y para determinar qué tipo de triángulo es (de acuerdo con la longitud de sus lados).

La clase PruebaFiguras es una clase que no tiene significado en el do- minio conceptual del problema. Como su nombre lo indica se ha definido para definir un único método *main* que sirva como punto de entrada para la ejecución del programa. El método *main* es un método estático cuya re- presentación en UML es subrayando la signatura del método.

Ejercicios propuestos

▶ Agregar una nueva clase denominada Rombo. Definir los métodos para calcular el área y el perímetro de esta nueva figura geométrica.

▶ Agregar una nueva clase denominada Trapecio. Definir los métodos para calcular el área y el perímetro de esta nueva figura geométrica.