

Java – Programación Orientada a Objetos (II)

Ejercicios a realizar

Ejercicio 1

Implementa la jerarquía de la Figura 1

1. Escribir las clases descritas en la Figura 1 con la implementación de sus métodos.
2. Crear una clase `Test` con un método `public static void main(String args[])` en el que debéis implementar lo siguiente:
 - a. Crear un *array* de `Geometry`

- b. Rellenar el *array* con instancias de un objeto de tipo `Rectangle`, otro de tipo `Circle` y otro de tipo `Triangle`.
- c. Iterar sobre el *array* para dibujar en la pantalla el perímetro y el área de sus geometrías.

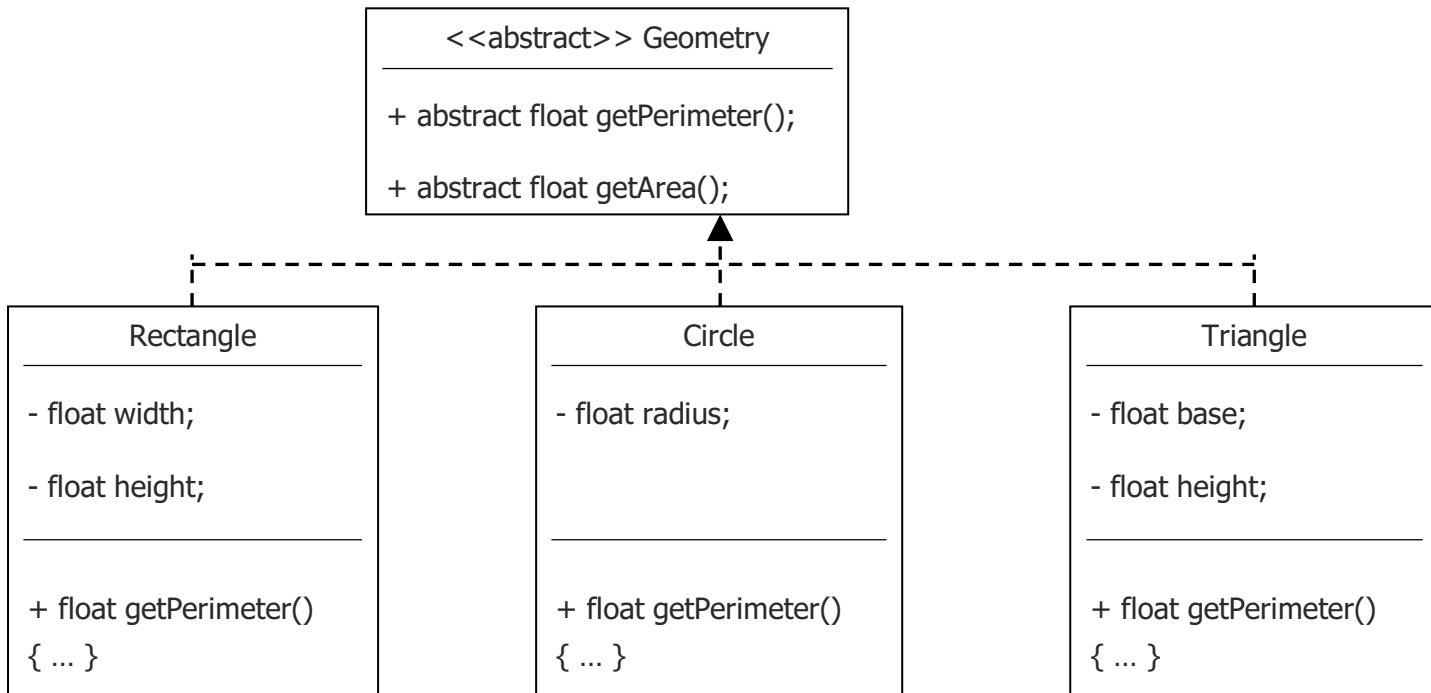


Figura 1. Jerarquía de geometrías

Ejercicio 2

Implementa la jerarquía de la Figura 2 con las siguientes observaciones:

1. No se puede implementar jerarquía múltiple con Java. La solución es usar interfaces.
 - a. Convierte `Geometry` en un interfaz.
 - b. Sobrecarga los métodos `getPerimeter` y `getArea` de las clases `Box`, `Sphere` y `Tetrahedron`
 - c. Implementar el método `getVolume`.
2. Implementar una clase aplicación (`TryGeometry`) en el que el método `main` cree un *array* en el que se puedan añadir instancias de cada una de esas nuevas clases al *array*. Extender el recorrido del *array* averiguando si cada objeto es instancia de `Geometry3D` (usar `instanceof`) para representar también el volumen.

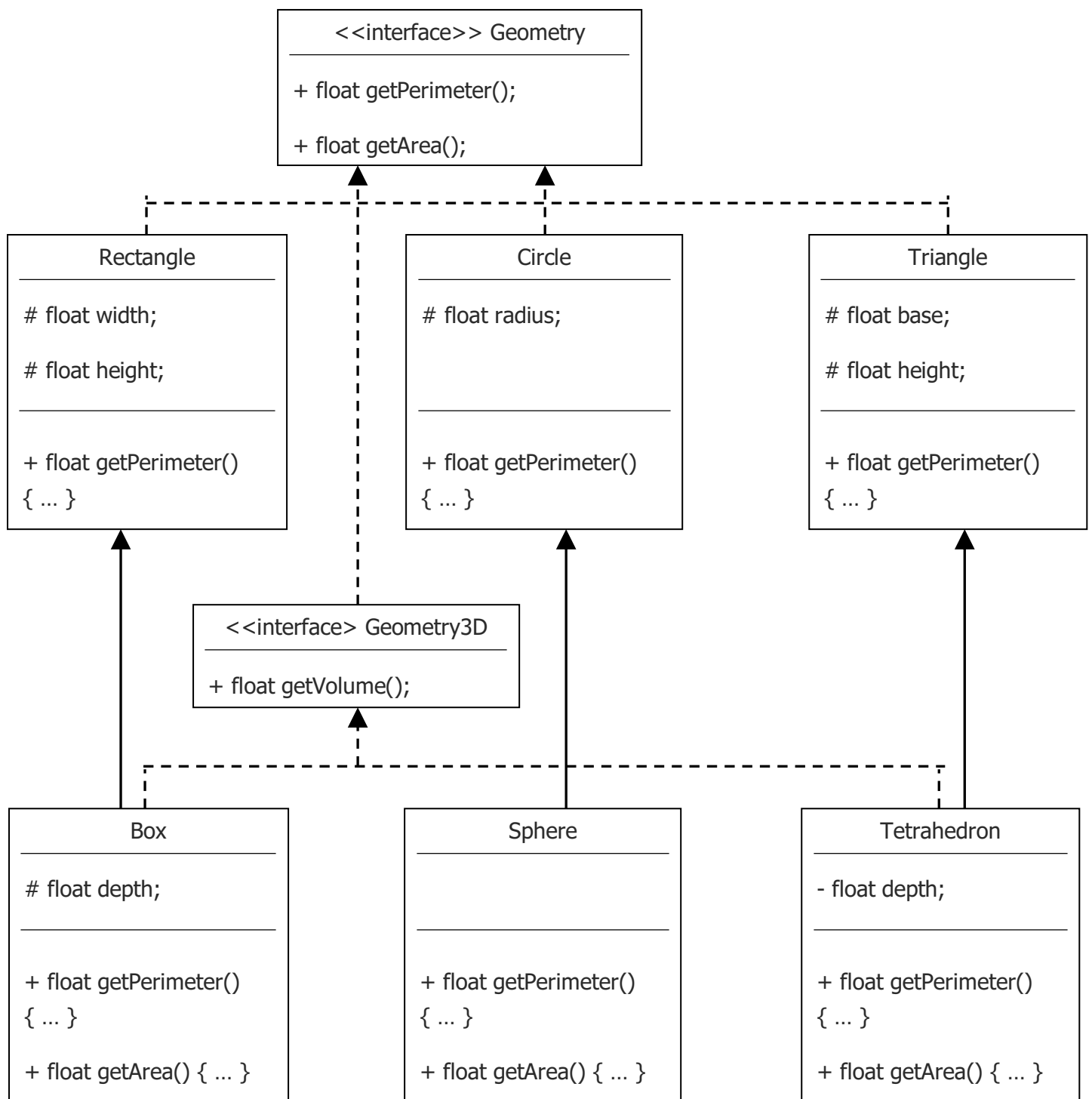


Figura 2. Jerarquía de geometrías en 3D

Ejercicio 3

Implementa la jerarquía que se indica en la Figura 3. Debes tener en cuenta que:

1. Escribir las clases descritas en la figura 3 con la implementación de sus métodos
2. Geometry2DFactory leerá los dos primeros parámetros
3. Geometry3DFactory leerá los tres primeros parámetros

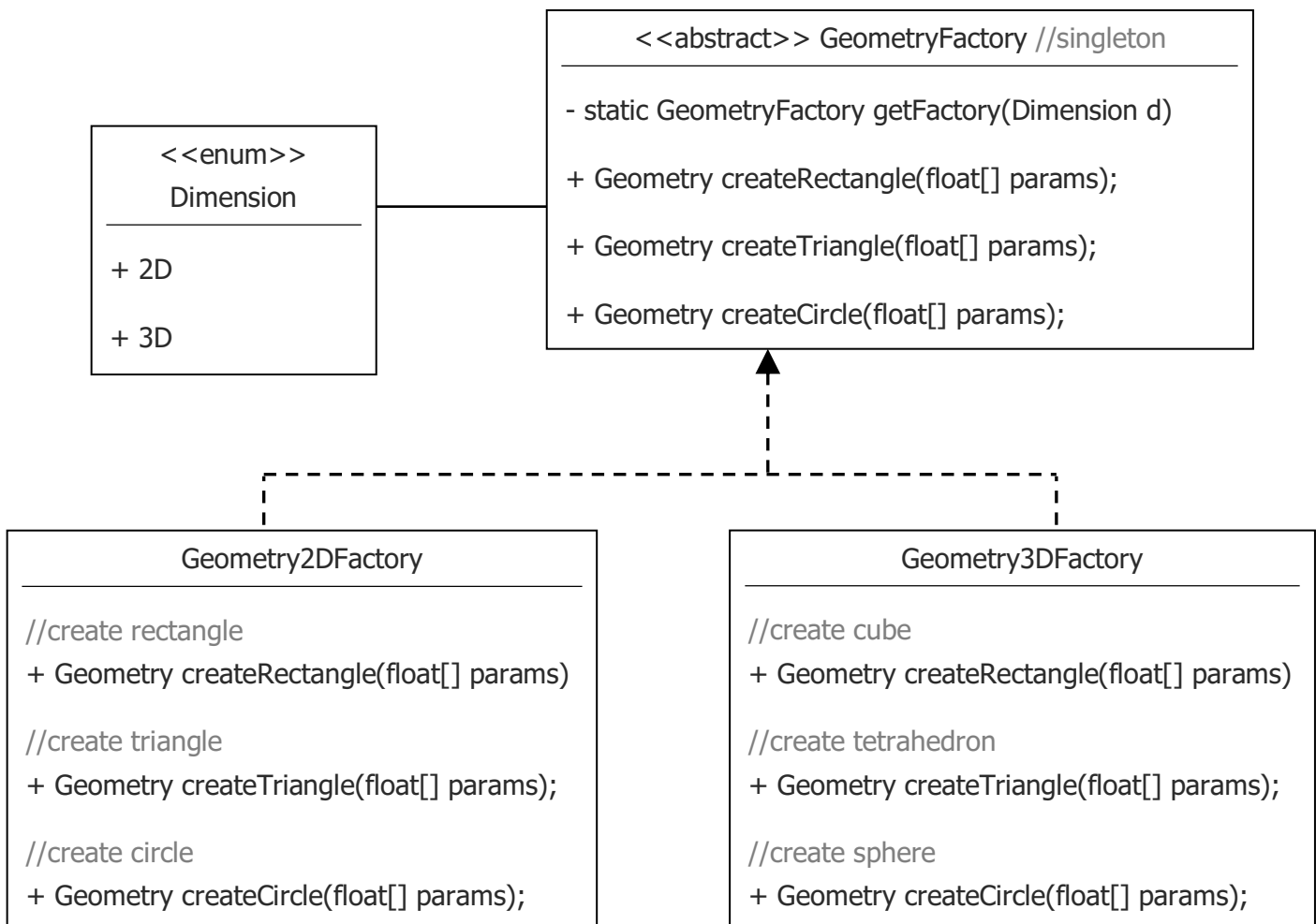


Figura 3. Factoría abstracta de geometrías