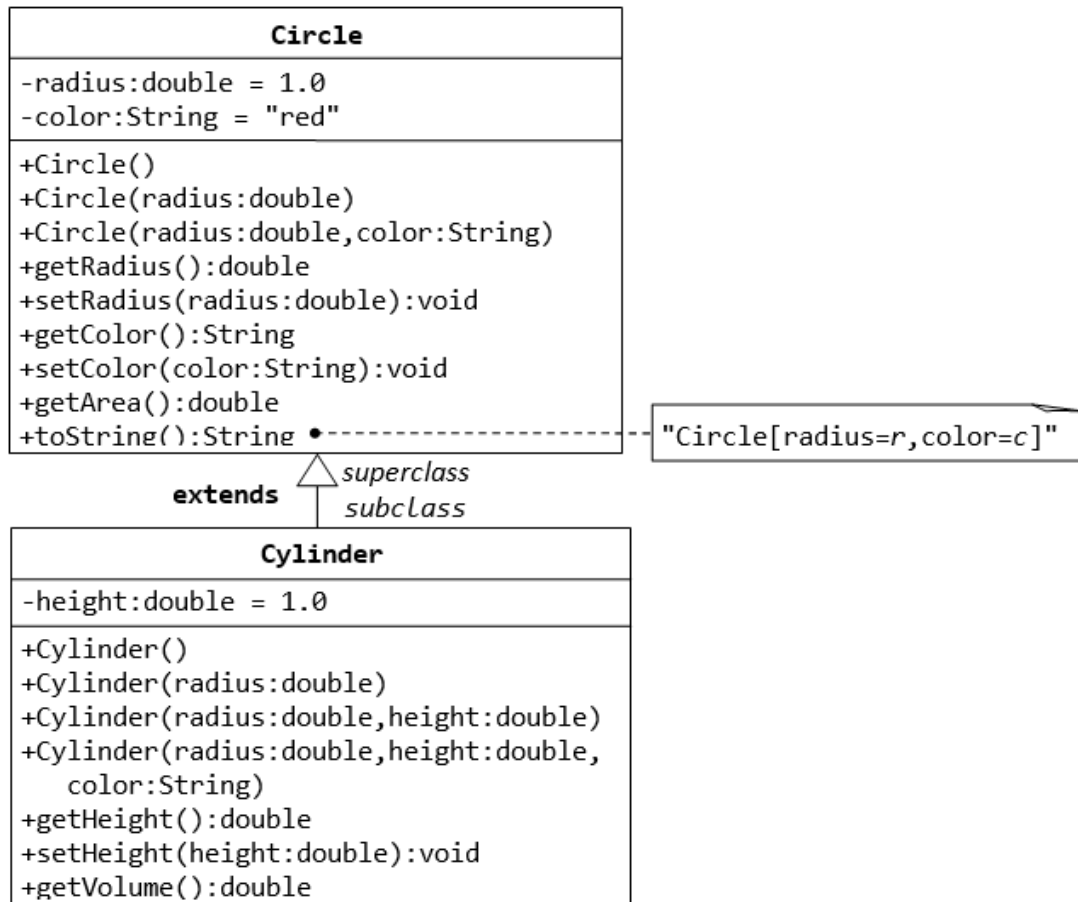


EJERCICIOS DE HERENCIA Y POLIMORFISMO

1. CÍRCULO Y CILINDRO

Escribe la clase `Circle` y la clase `Cylinder`, que hereda de `Circle`. Utiliza el siguiente diagrama de clases para concretar atributos y métodos:

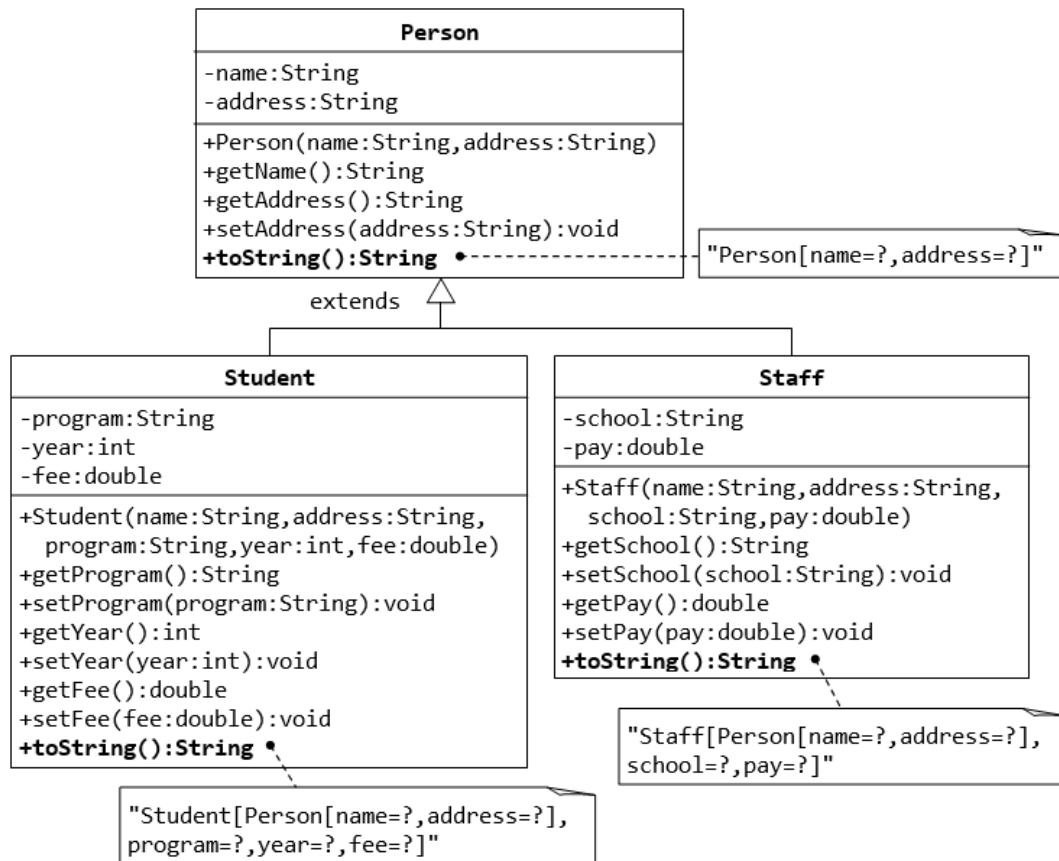


La subclase `Cylinder` debe invocar los constructores de la superclase `Circle` (mediante `super()` y `super(radius)`) y hereda los atributos y los métodos de la superclase.

Escribe un programa de prueba para verificar el correcto funcionamiento de las clases definidas.

2. PERSONA, ESTUDIANTE Y EMPLEADO

Escribe la clase `Person` y las clases `Student` y `Staff`, que heredan de `Person`. Utiliza el siguiente diagrama de clases para concretar atributos y métodos:



Escribe un programa de prueba para verificar el correcto funcionamiento de las clases definidas.

3. FORMA, CÍRCULO, RECTÁNGULO, CUADRADO

Escribe la clase abstracta `Shape` y sus subclases `Circle`, `Rectangle` y `Square`. Utiliza el siguiente diagrama de clases para concretar atributos y métodos:

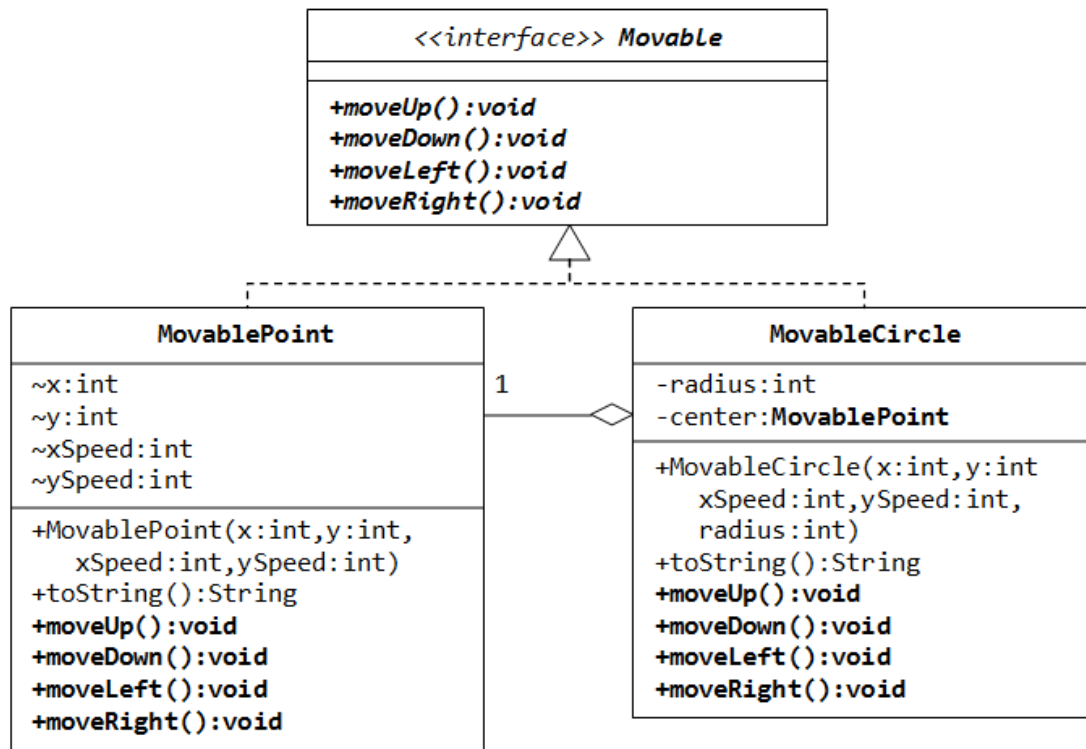


Las subclases `Circle` y `Rectangle` deberán sobrescribir los métodos abstractos `getArea()` y `getPerimeter()` y proporcionar una implementación adecuada. Además, deberán sobrescribir el método `toString()`.

Escribe un programa de prueba para verificar el correcto funcionamiento que muestre ejemplos de polimorfismo y explica las salidas.

4. MÓVIL, PUNTO MÓVIL Y CÍRCULO MÓVIL

Escribe la interfaz `Movable`, que contiene varios métodos abstractos, y las clases concretas `MovablePoint` y `MovableCircle`, que proporcionan la implementación de dichos métodos de la interfaz. Utiliza el siguiente diagrama de clases para concretar atributos y métodos:



En la clase `MovablePoint`, declara los atributos como variables instancia con acceso de paquete tal como se muestran con `'~'` en el diagrama de clases (las clases del mismo paquete pueden acceder a estos atributos directamente).

En la clase `MovableCircle`, utiliza un `MovablePoint` para representar el centro.

Escribe un programa de prueba para verificar el correcto funcionamiento que incluya las siguientes sentencias:

```
Movable m1 = new MovablePoint(5, 6, 10);
System.out.println(m1);
m1.moveLeft();
System.out.println(m1);

Movable m2 = new MovableCircle(2, 1, 2, 20);
System.out.println(m2);
m2.moveRight();
System.out.println(m2);
```

Explica las salidas producidas por este programa de prueba.