

## プログラミングⅡ 2022 第11回 演習課題 「抽象クラス」

1. 下記は、立体図形の直方体を表すクラス Cuboid と三角柱を表すクラス TriangularPrism である。これらのクラスから共通部分である立体の高さ length, 体積 volume (計算処理), 底面積 baseArea (名前だけ) を取り出し、共通のスーパークラスとして「角柱」を表す**抽象クラス Prism**を作成せよ。さらに、それを継承して Cuboid と TriangularPrism を定義しなおし、適当な main メソッドを作成して動作を確認せよ。

```
/* Cuboid.java */
// 直方体クラス
public class Cuboid {

    private double rectWidth, rectHeight; // 底面の長方形の横・縦
    private double length;                // 立体の高さ

    // コンストラクタ
    public Cuboid(double w, double h, double l) {
        this.rectWidth = w;
        this.rectHeight = h;
        this.length = l;
    }

    // 底面積を求める
    public double baseArea() {
        return this.rectWidth * this.rectHeight;
    }

    // 体積を求める
    public double volume() {
        return this.baseArea() * this.length;
    }
}
```

```
/* TriangularPrism.java */
// 三角柱クラス
public class TriangularPrism {

    private double triBase, triHeight; // 底面の三角形の底辺・高さ
    private double length;            // 立体の高さ

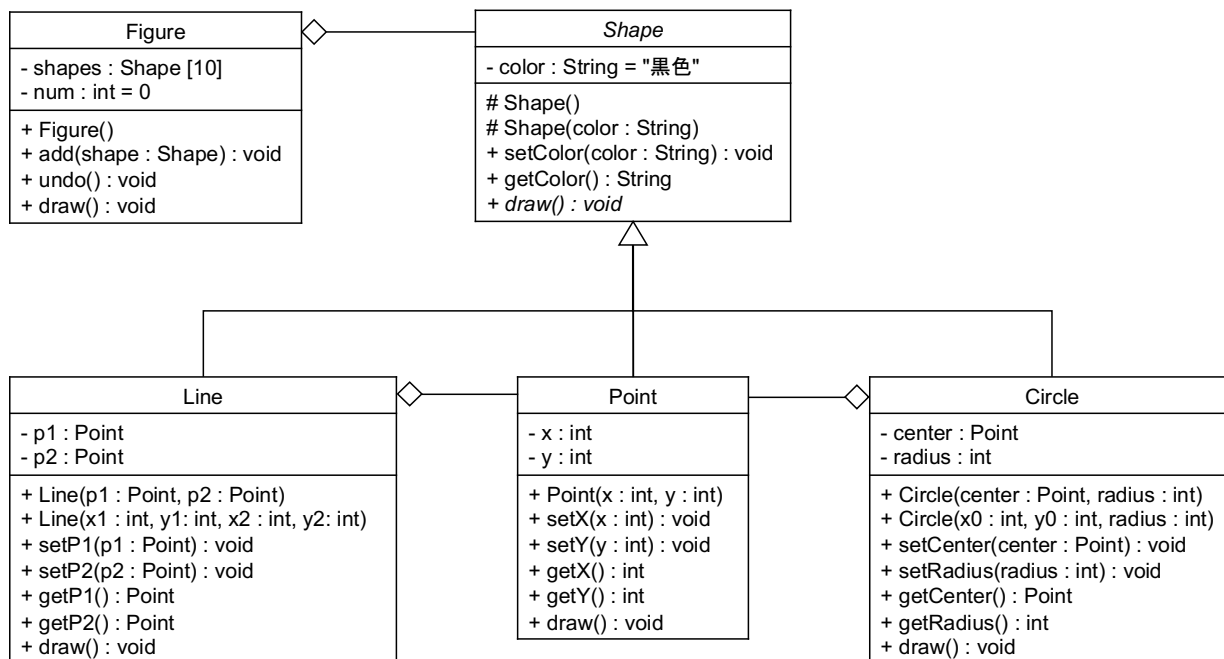
    // コンストラクタ
    public TriangularPrism(double b, double h, double l) {
        this.triBase = b;
        this.triHeight = h;
        this.length = l;
    }

    // 底面積を求める
    public double baseArea() {
        return this.triBase * this.triHeight / 2.0;
    }

    // 体積を求める
    public double volume() {
        return this.baseArea() * this.length;
    }
}
```

2. 前問で作成した Prism, Cuboid, TriangularPrism の関係を UML のクラス図で示せ。

3. 下記はお絵描きソフトを想定したクラスの設計である。Figure は図, Shape は図形, Point は点, Line は線, Circle は円を表すクラスである。Shape は抽象クラスであり, 抽象メソッド draw() を持つ。Figure クラスと Main クラスの実装例と実行例を示したので, 他のクラスを作成してプログラムを完成させよ。



```

/* Figure.java */
public class Figure {
    private Shape[] shapes;
    private int num = 0;

    public Figure() {
        this.shapes = new Shape[10];
    }

    public void add(Shape shape) {
        if (this.num < shapes.length) {
            this.shapes[this.num++] = shape;
        }
    }

    public void undo() {
        if (this.num > 0) {
            this.shapes[--this.num] = null;
            System.out.println("1つ取り消し");
        }
    }

    public void draw() {
        System.out.println("描画開始");
        for (int i = 0; i < num; i++) {
            this.shapes[i].draw();
        }
    }
}

```

```

/* Main.java */
public class Main {
    public static void main(String[] args) {
        Figure fig = new Figure();
        Point p1 = new Point(0, 10);
        fig.add(p1);
    }
}

```

```

Point p2 = new Point(20, 30);
p2.setColor("赤色");
fig.add(p2);
Circle c1 = new Circle(p2, 40);
c1.setColor("青色");
fig.add(c1);
fig.add(new Line(20, 30, 40, 50));
fig.draw();

```

```

fig.undo();
fig.undo();
fig.draw();

```

```

Line l1 = new Line(p1, p2);
l1.setColor("黄色");
fig.add(l1);
fig.draw();
}
}

```

==== Main による実行例 ====

描画開始  
 点(0,10)を黒色で描きます  
 点(20,30)を赤色で描きます  
 中心(20,30)で半径 40 の円を青色で描きます  
 点(20,30)と点(40,50)を結ぶ線を黒色で描きます  
 1つ取り消し  
 1つ取り消し  
 描画開始  
 点(0,10)を黒色で描きます  
 点(20,30)を赤色で描きます  
 描画開始  
 点(0,10)を黒色で描きます  
 点(20,30)を赤色で描きます  
 点(0,10)と点(20,30)を結ぶ線を黄色で描きます