オブジェクト指向プログラミング　No.7

17173033 後藤 亘

1.

ソースコード

Traiangle.java

```
public class Traiangle {

	private double b = 1.0;

	private double h = 1.0;



	public Traiangle(){

	}



	public Traiangle(double b, double h){

		this.b = b;

		this.h = h;

	}
```

```
public double getB(){
        return b;
}

public double getH(){
        return h;
}

public void setB(double b){
        this.b = b;
}

public void setH(double h){
        this.h = h;
}
```

```
    public double calcArea(){

        return (b*h)/2;

    }

}
```

Pyramid.java

```
class Traiangle {

    private double b = 1.0;

    private double h = 1.0;



    Traiangle(){

    }



    Traiangle(double b, double h){

        this.b = b;

        this.h = h;
```

```
}

double getB(){
        return b;
}

double getH(){
        return h;
}

void setB(double b){
        this.b = b;
}

void setH(double h){
        this.h = h;
```

```
    }

    double calcArea(){
        return (b*h)/2;
    }
}

public class Pyramid extends Traiangle{
    private double k = 1.0;
    //側面積は底辺がbで高さがhの半分とkを合わせたものの平方根と
する

    public Pyramid(){
    }

    public Pyramid(double b, double h, double k){
        super(b, h);

        this.k = k;

    }



    public void setK(double k){

        this.k = k;

    }



    public double getK(){

        return k;

    }



    public double calcVol(){

        return super.getB()*super.getH()*k/3;

    }
```

```
        public double calcArea(){
                double sum1 = super.calcArea();
                double sum2 = 3 * super.getB() * Math.sqrt(super.getH() *
super.getH() + k * k);
                return sum1 + sum2;
        }




}
```

calcTester.java

```
public class calcTester{
        public static void main(String[] args){
                Traiangle T1 = new Traiangle();
                Traiangle T2 = new Traiangle(2.0, 4.5);
                Pyramid P1 = new Pyramid();
}
```

```
        Pyramid P2 = new Pyramid(2.0, 4.5, 7.5);
        double sample1 = T1.calcArea();
        double sample2 = T2.calcArea();
        double sample3 = P1.calcVol();
        double sample4 = P1.calcArea();
        double sample5 = P2.calcVol();
        double sample6 = P2.calcArea();

        System.out.println(sample1);
        System.out.println(sample2);
        System.out.println(sample3);
        System.out.println(sample4);
        System.out.println(sample5);
        System.out.println(sample6);
    }
}
```

実行結果

```
gotouwatarusMBP:No.7 gotouwataru$ java calcTester

0.5

4.5

0.3333333333333333

4.742640687119286

22.5

56.978567053607705
```

2.

(a)3 行目は Point クラスの distance(Point b)を実行しており、4 行目は distance(Point2D b)を実行している。4 行目のようにするにはダウンキャストを行わなければならない。

PointTester.java

```
class Point3D extends Point2D {

	private int z = 1;
```

```
Point3D(){}
Point3D(int x, int y, int z){
        super(x, y);
        this.z = z;
}
double distance(){
        return Math.sqrt(x*x+y*y+z*z);
}
double distance(Point b){
        if( b instanceof Point3D){
                Point3D a = (Point3D)b;
                int dx, dy, dz;
                dx = this.x - a.x;
                dy = this.y - a.y;
                dz = this.z - a.z;
                return Math.sqrt(dx*dx+dy*dy+dz*dz);
        } else {
            int dx, dy, dz;
            dx = this.x - b.x;
            dy = this.y - 1;
            dz = this.z - 1;
            return Math.sqrt(dx*dx+dy*dy+dz*dz);
        }
    }
    double distance(Point2D b){
        if( b instanceof Point3D){
            Point3D a = (Point3D)b;
            int dx, dy, dz;
            dx = this.x - a.x;
            dy = this.y - a.y;
            dz = this.z - a.z;
            return Math.sqrt(dx*dx+dy*dy+dz*dz);
        } else {
            int dx, dy, dz;
            dx = this.x - b.x;
            dy = this.y - b.y;
            dz = this.z - 1;
            return Math.sqrt(dx*dx+dy*dy+dz*dz);
        }
    }
    double distance(Point3D b){
        int dx, dy, dz;
        dx = this.x - b.x;
        dy = this.y - b.y;
        dz = this.z - b.z;
        return Math.sqrt(dx*dx+dy*dy+dz*dz);
    }
    public String toString(){
```

```
                return String.format("(%d, %d, %d)", x, y, z);
        }
}

class Point2D extends Point {
        protected int y = 1;
        Point2D(){}
        Point2D(int x, int y){
                super(x);
                this.y = y;
        }
        double distance(){
                return Math.sqrt(x*x+y*y);
        }

        double distance(Point b){
```

```
        if( b instanceof Point2D){
                Point2D a = (Point2D)b;
                int dx, dy;
                dx = this.x - a.x;
                dy = this.y - a.y;
                return Math.sqrt(dx*dx+dy*dy);
        } else {
                int dx, dy;
                dx = this.x - b.x;
                dy = this.y - 1;
                return Math.sqrt(dx*dx+dy*dy);
        }
}

double distance(Point2D b){
        int dx, dy;
        dx = this.x - b.x;
        dy = this.y - b.y;
        return Math.sqrt(dx*dx+dy*dy);
    }


    public String toString(){
        return String.format("(%d, %d)", x, y);
    }
}

class Point {
    protected int x = 1;
    Point(){}
    Point(int x){
        this.x = x;
        }
        double distance(){
                return Math.sqrt(x*x);
        }
        double distance(Point b){
                int dx;
                dx = this.x - b.x;
                return Math.sqrt(x*x);
        }
        public String toString(){
                return String.format("(%d)", x);
        }
}



class PointTester {
        public static void main(String[] args){

                Point p1 = new Point(2);

                Point p2 = new Point(-3);

                System.out.println("点 p1 から原点からの距離は" +

p1.distance());

                System.out.println("点 p1 から p2 の距離は" +

p1.distance(p2));



                Point2D q1 = new Point2D(1, -1);

                Point q2 = new Point2D(0, 1);

                Point3D o1 = new Point3D(0, 0, 0);

                Point3D o2 = new Point3D(3, 4, 0);

                System.out.println("点 q1 と q2 の距離は" +

q1.distance(q2));

                System.out.println("点 q1 と q2 の距離は" +

q1.distance((Point2D)q2));
```

```
                    System.out.println("点 o1 と o2 の距離は" + o1.distance(o2));

                    System.out.println("点 q1 と o1 の距離は" +

o1.distance(q1));

                    System.out.println("点 q1 と p1 の距離は" +

q1.distance(p1));



          }

}
```

実行結果

```
gotouwatarusMBP:No.7 gotouwataru$ java PointTester

点 p1 から原点からの距離は 2.0

点 p1 から p2 の距離は 2.0

点 q1 と q2 の距離は 2.23606797749979

点 q1 と q2 の距離は 2.23606797749979

点 o1 と o2 の距離は 5.0

点 q1 と o1 の距離は 1.7320508075688772
```

```
点 q1 と p1 の距離は 2.23606797749979
```