```
------------------------------------------------------------------------------------
public abstract class Solid_Figure {

        protected double volume
        protected double surface

        public double get_v() {
                return volume
        }

        public double get_s() {
                return surface
        }

        public abstract void print_info();

}
```

```
-----------------------------------------------------------------------------
public class Rect_Para extends Solid_Figure {
        private double length_x
        private double length_y
        private double length_z

        public boolean set_x (double length) {
                if (length > 0) {
                        length_x = length;
                        if (is_valid()) {
                                volume = calc_v();
                                surface = calc_s();
                        }
                        return true
                }
                else
                        return false
        }

        public boolean set_y (double length) {
                if (length > 0) {
                        length_y = length;

                        if (is_valid()) {
                                volume = calc_v();
                                surface = calc_s();
                        }
                        return true
                }
                else
                        return false
        }

        public boolean set_z (double length) {
                if (length > 0) {
                        length_z = length;

                        if (is_valid()) {
                                volume = calc_v();
                                surface = calc_s();
                        }
                        return true
                }
                else
                        return false
        }


        public double get_x() {
                return length_x
        }

        public double get_y() {
                return length_y
        }

        public double get_z() {
                return length_z
        }

        public boolean is_valid() {
                return (length_x > 0 && length_y > 0 && length_z > 0);
        }
```

```
    public void print_info() {
        System.out.print("x : " + length_x + " y : " +
                length_y + " z : " + length_z);
        System.out.println(" v : " + volume + " s : " + surface);
    }

    private double calc_v() {
        return length_x * length_y * length_z
    }

    private double calc_s() {
        return 2 * (length_x * length_y + length_y *
                length_z + length_z * length_x);
    }
}
```

```java
public class Sphere extends Solid_Figure {
        private double radius

        public boolean set_r (double length) {
                if (length > 0) {
                        radius = length;
                        volume = calc_v();
                        surface = calc_s();
                        return true
                }
                else
                        return false
        }

        public double get_r() {
                return radius
        }

        public void print_info() {
                System.out.print("r : " + radius);
                System.out.println(" v : " + volume + " s : " + surface);
        }

        private double calc_v() {
                return 4. / 3. * Math.PI * radius * radius * radius
        }

        private double calc_s() {
                return 4. * Math.PI * radius * radius ;
        }
}
```

```
----------------------------------------------------------------------------
public class Test_Solid_Figure {
    int max_v_index(Solid_Figure [] sf_arr) {
        double max_v = 0;
        int max_index = -1;

        for (int i = 0; i < sf_arr.length i++) {
            if (sf_arr[i].get_v() > max_v) {
                max_v = sf_arr[i].get_v();
                max_index = i;
            }
        }
        return max_index;
    }

    int max_s_index(Solid_Figure [] sf_arr) {
        double max_s = 0;
        int max_index = -1;

        for (int i = 0; i < sf_arr.length i++) {
            if (sf_arr[i].get_v() > max_s) {
                max_s = sf_arr[i].get_s();
                max_index = i;
            }
        }
        return max_index;
    }
}
```

```
---------------------------------------------------------------------------

public class Testdrive_Solid_Figure {

        public static void main(String[] args) {

                Solid_Figure [] sf_arr = new Solid_Figure[5];
                Test_Solid_Figure refree = new Test_Solid_Figure();

                Rect_Para rp_a = new Rect_Para();
                rp_a.set_x(1.0);
                rp_a.set_y(1.0);
                rp_a.set_z(1.0);

                sf_arr[0] = (Solid_Figure) rp_a;

                Rect_Para rp_b = new Rect_Para();
                rp_b.set_x(12.0);
                rp_b.set_y(5.0);
                rp_b.set_z(3.0);

                sf_arr[1] = (Solid_Figure) rp_b;

                Rect_Para rp_c = new Rect_Para();
                rp_c.set_x(13.0);
                rp_c.set_y(5.0);
                rp_c.set_z(3.0);

                sf_arr[2] = (Solid_Figure) rp_c;

                Sphere g1 = new Sphere();
                g1.set_r(2.0);

                sf_arr[3] = (Solid_Figure) g1;

                Sphere g2 = new Sphere();
                g2.set_r(4.0);

                sf_arr[4] = (Solid_Figure) g2;

                System.out.println("가장부피가큰도형의정보);
                sf_arr[refree.max_v_index(sf_arr)].print_info();
                System.out.println("가장표면적이큰도형의정보);
                sf_arr[refree.max_s_index(sf_arr)].print_info();

        }

}
```