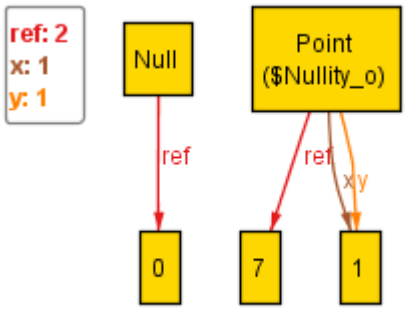
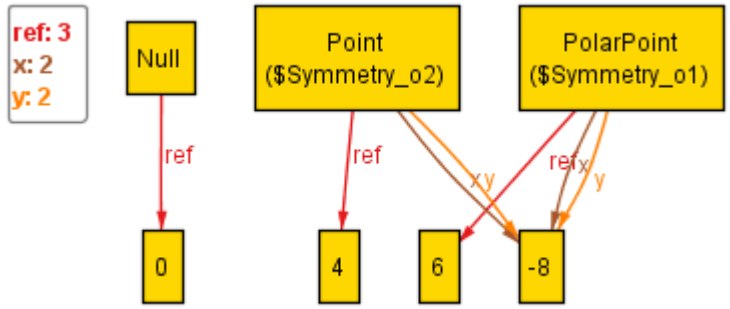
0.



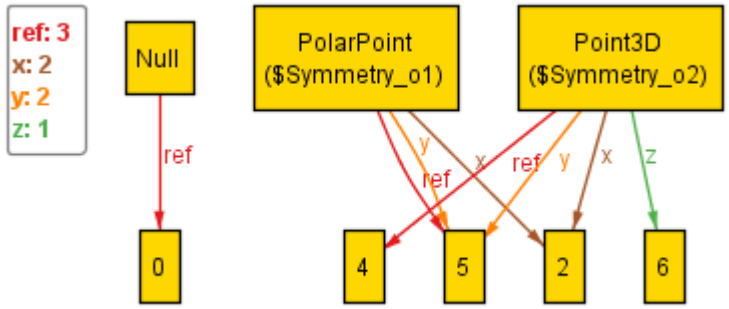

```
public class ErrorReproducer {
    public static void main(String[] args) throws Exception {
        Point p = new Point(1, 1);

        boolean expected = false;
        boolean actual = p.equals(null);

        System.out.printf("Expected %b Actual %b", expected, actual);
    }
}
```

1.



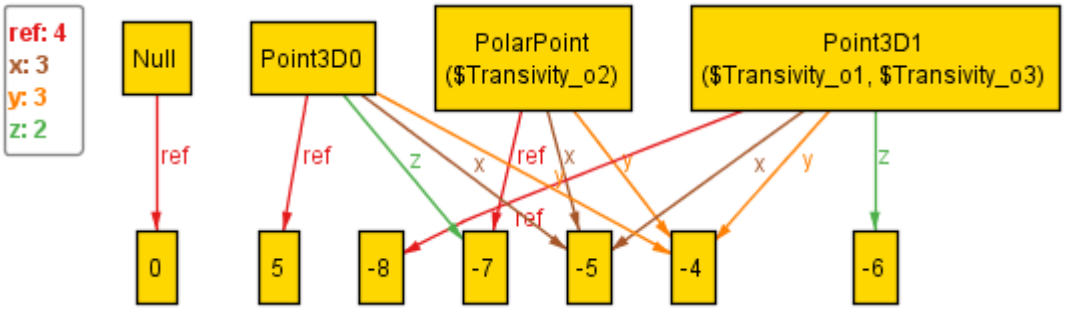

```java
public class ErrorReproducer {
    public static void main(String[] args) {
        //reflexive
        PolarPoint p = new PolarPoint(new Point(1, 1));

        boolean reflexive = p.equals(p);
        System.out.printf("Reflexive test: expected true, got %s\n", reflexive);

        //symmetric
        Point pnt = new Point(-8, -8);
        p = new PolarPoint(pnt);
        boolean pntp = pnt.equals(p);
        boolean ppnt = p.equals(pnt);

        System.out.printf("symmetry\n\tpoint=polar: %s\n\tpolar=point: %s", pntp, ppnt);
    }
}
```

2.




```
public class ErrorReproducer {
    public static void main(String[] args) {
        //symmetry
        PolarPoint pp = new PolarPoint(new Point(2, 5));
        Point3D p3d = new Point3D(2, 5, 6);

        boolean forward = pp.equals(p3d);
        boolean back = p3d.equals(pp);

        System.out.printf("polar=3d:%s\n3d=polar:%s\n", forward, back);

    }
}
```

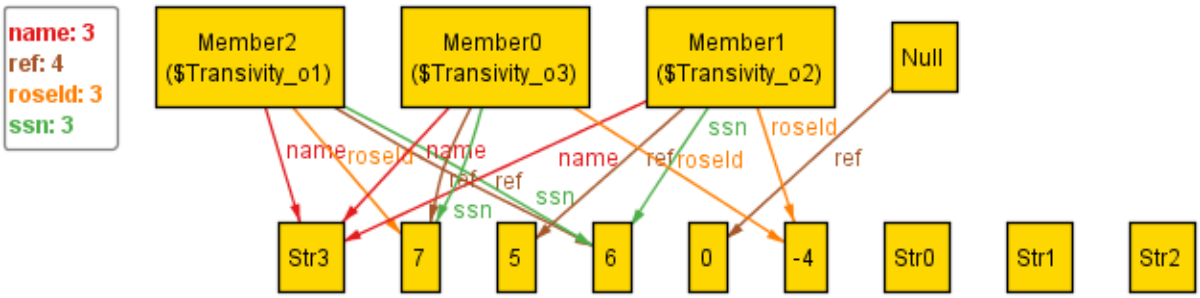
3.




```
public class ErrorReproducer {
    public static void main(String[] args) {
        Point3D p1 = new Point3D(-5, -4, -7);
        PolarPoint p2 = new PolarPoint(new Point(-5, -4));
        Point3D p3 = new Point3D(-5, -4, -6);

        boolean p1p2 = p1.equals(p2);
        boolean p2p3 = p2.equals(p3);
        boolean p1p3 = p1.equals(p3);

        System.out.printf("p1=p2: %s\n"
                        + "p2=p3: %s\n"
                        + "p1=p3: %s\n", p1p2, p2p3,p1p3);

    }
}
```

4. No Counter Examples found

5.




```
public class ErrorReproducer {
    public static void main(String[] args) {
        String name = "";
        Member p1 = new Member(name, -4, 7);
        Member p2 = new Member(name, -4, 6);
        Member p3 = new Member(name, 7, 6);


        boolean p1p2 = p1.equals(p2);
        boolean p2p3 = p2.equals(p3);
        boolean p1p3 = p3.equals(p1);

        System.out.printf("p1=p2: %s\n"
                        + "p2=p3: %s\n"
                        + "p1=p3: %s\n", p1p2, p2p3,p1p3);
    }
}
```