Test of Point.intersect(Point) 1

Case 1: `Point intersects point`

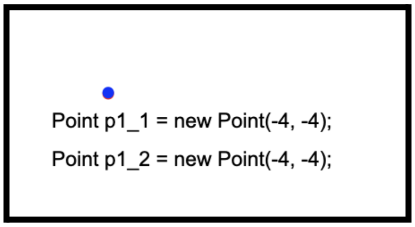
Point p1_1 = new Point(-4, -4);

Point p1_2 = new Point(-4, -4);

Expected Result: true

Code Result: true

Test Result: pass



Point p1_1 = new Point(-4, -4);

Point p1_2 = new Point(-4, -4);

Case 2: `Point does not intersect point`

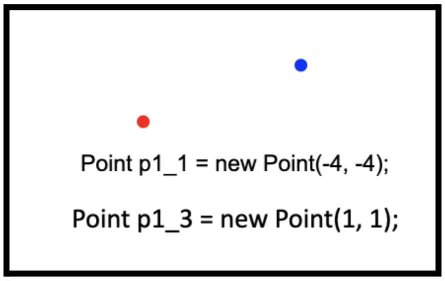
Point p1_1 = new Point(-4, -4);

Point p1_3 = new Point(1, 1);

Expected Result: false

Code Result: false

Test Result: pass



Point p1_1 = new Point(-4, -4);

Point p1_3 = new Point(1, 1);

Test of Point.intersect(Circle) 2

Case 3: `Point is in circle`

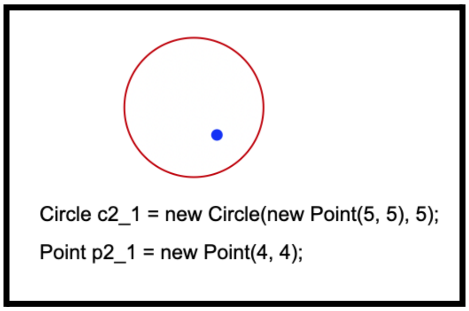
Circle c2_1 = new Circle(new Point(5, 5), 5);

Point p2_1 = new Point(4, 4);

Expected Result: true

Code Result: true

Test Result: pass

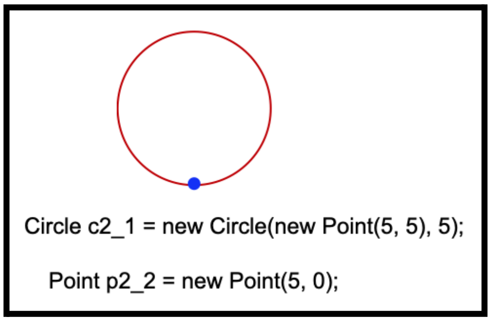


Circle c2_1 = new Circle(new Point(5, 5), 5);

Point p2_1 = new Point(4, 4);

Case 4: Point is on circle's edge

Circle c2_1 = new Circle(new Point(5, 5), 5);
Point p2_2 = new Point(5, 0);
Expected Result: true
Code Result: true
Test Result: pass

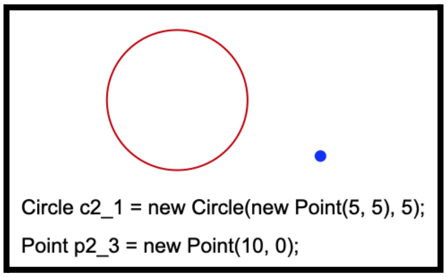


Circle c2_1 = new Circle(new Point(5, 5), 5);

Point p2_2 = new Point(5, 0);

Case 5: Point is outside circle

Circle c2_1 = new Circle(new Point(5, 5), 5);
Point p2_3 = new Point(10, 0);
Expected Result: false
Code Result: false
Test Result: pass



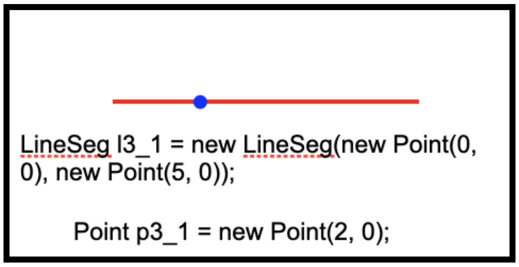
Circle c2_1 = new Circle(new Point(5, 5), 5);

Point p2_3 = new Point(10, 0);

Test of Point.intersect(LineSeg) 3

Case 6: Point is on line

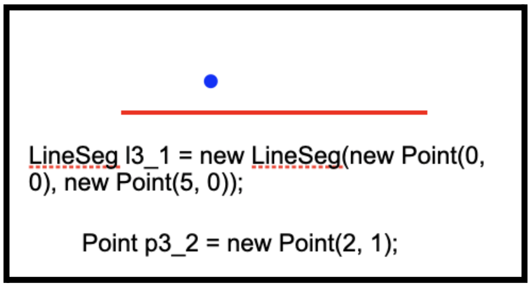
LineSeg l3_1 = new LineSeg(new Point(0, 0), new Point(5, 0));
Point p3_1 = new Point(2, 0);
Expected Result: true
Code Result: true
Test Result: pass



LineSeg l3_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Point p3_1 = new Point(2, 0);

Case 7: Point is not on line
LineSeg l3_1 = new LineSeg(new Point(0, 0), new Point(5, 0));
Point p3_2 = new Point(2, 1);
Expected Result: false
Code Result: false
Test Result: pass

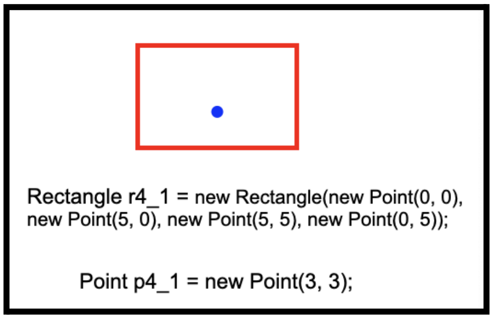


LineSeg l3_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Point p3_2 = new Point(2, 1);

Test of Point.intersect(Rectangle) 4

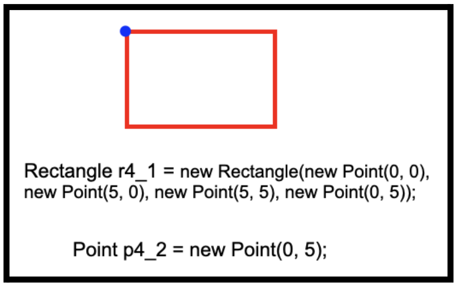
Case 8: Point is in rectangle
Rectangle r4_1 = new Rectangle(new Point(0, 0), new Point(5, 0), new Point(5, 5), new Point(0, 5));
Point p4_1 = new Point(3, 3);
Expected Result: true
Code Result: true
Test Result: pass



Rectangle r4_1 = new Rectangle(new Point(0, 0), new Point(5, 0), new Point(5, 5), new Point(0, 5));

Point p4_1 = new Point(3, 3);

Case 9: Point is on rectangle line
Rectangle r4_1 = new Rectangle(new Point(0, 0), new Point(5, 0), new Point(5, 5), new Point(0, 5));
Point p4_2 = new Point(0, 5);
Expected Result: true
Code Result: true
Test Result: pass



Rectangle r4_1 = new Rectangle(new Point(0, 0), new Point(5, 0), new Point(5, 5), new Point(0, 5));

Point p4_2 = new Point(0, 5);

Case 10: `Point is outside rectangle`

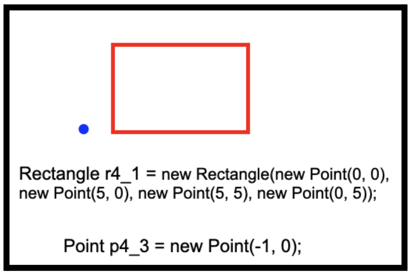
Rectangle r4_1 = new Rectangle(new Point(0, 0), new Point(5, 0), new Point(5, 5), new Point(0, 5));

Point p4_3 = new Point(-1, 0);

Expected Result: false

Code Result: false

Test Result: pass



Test of LineSeg.intersect(Point) 5

Case 11: `Point is on line`

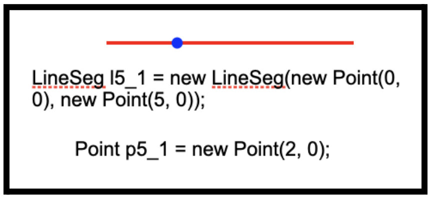
LineSeg l5_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Point p5_1 = new Point(2, 0);

Expected Result: true

Code Result: true

Test Result: pass



Case 12: `Point is not on line`

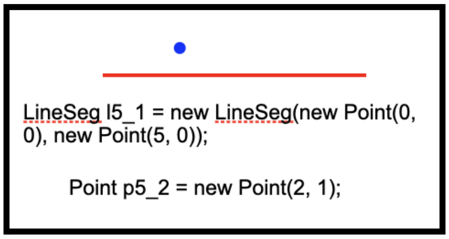
LineSeg l5_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Point p5_2 = new Point(2, 1);

Expected Result: false

Code Result: false

Test Result: pass



Test of LineSeg.intersect(Circle) 6

Case 13: Line intersects circle

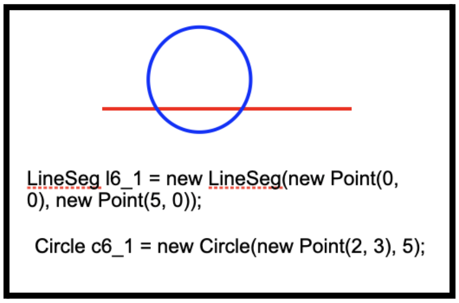
LineSeg l6_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Circle c6_1 = new Circle(new Point(2, 3), 5);

Expected Result: true

Code Result: true

Test Result: pass



```
LineSeg l6_1 = new LineSeg(new Point(0,
0), new Point(5, 0));

 Circle c6_1 = new Circle(new Point(2, 3), 5);
```

Case 14: Line and circle edge touching

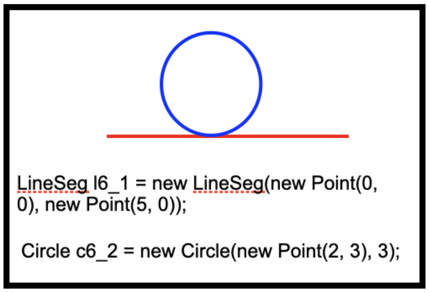
LineSeg l6_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Circle c6_2 = new Circle(new Point(2, 3), 3);

Expected Result: true

Code Result: true

Test Result: pass



```
LineSeg l6_1 = new LineSeg(new Point(0,
0), new Point(5, 0));

 Circle c6_2 = new Circle(new Point(2, 3), 3);
```

Case 15: Line and circle do not intersect

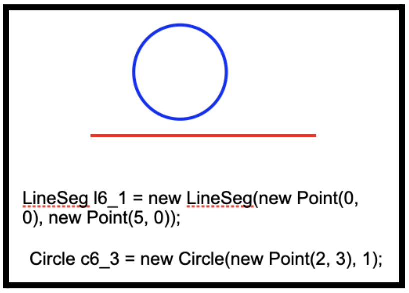
LineSeg l6_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Circle c6_3 = new Circle(new Point(2, 3), 1);

Expected Result: false

Code Result: false

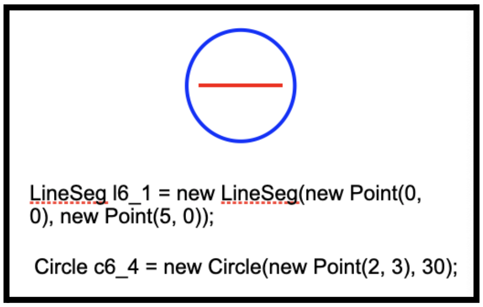
Test Result: pass



```
LineSeg l6_1 = new LineSeg(new Point(0,
0), new Point(5, 0));

 Circle c6_3 = new Circle(new Point(2, 3), 1);
```

Case 16: Line is completely in circle
LineSeg l6_1 = new LineSeg(new Point(0, 0), new Point(5, 0));
Circle c6_4 = new Circle(new Point(2, 3), 30);
Expected Result: true
Code Result: true
Test Result: pass

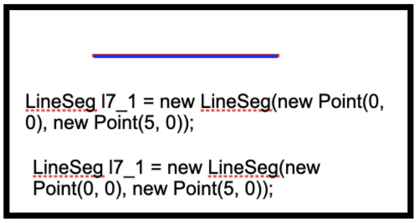


LineSeg l6_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Circle c6_4 = new Circle(new Point(2, 3), 30);

Test of LineSeg.intersect(LineSeg) 7

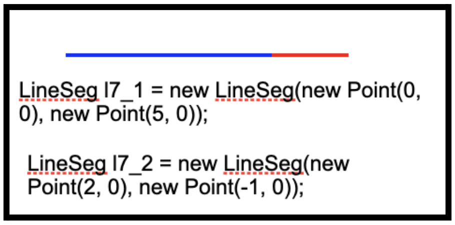
Case 17: Line intersects with itself
LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));
LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));
Expected Result: true
Code Result: true
Test Result: pass



LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

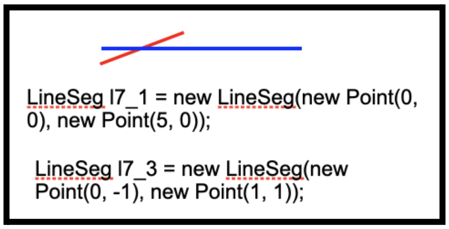
Case 18: Line intersects multiple times with another line
LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));
LineSeg l7_2 = new LineSeg(new Point(2, 0), new Point(-1, 0));
Expected Result: true
Code Result: true
Test Result: pass



LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

LineSeg l7_2 = new LineSeg(new Point(2, 0), new Point(-1, 0));

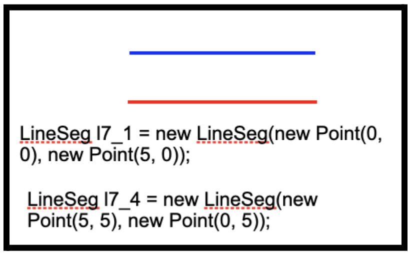
Case 19: Line intersects with other line once
LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));
LineSeg l7_3 = new LineSeg(new Point(0, -1), new Point(1, 1));
Expected Result: true
Code Result: true
Test Result: pass



LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

LineSeg l7_3 = new LineSeg(new Point(0, -1), new Point(1, 1));

Case 20: Lines do not intersect
LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));
LineSeg l7_4 = new LineSeg(new Point(5, 5), new Point(0, 5));
Expected Result: false
Code Result: false
Test Result: pass

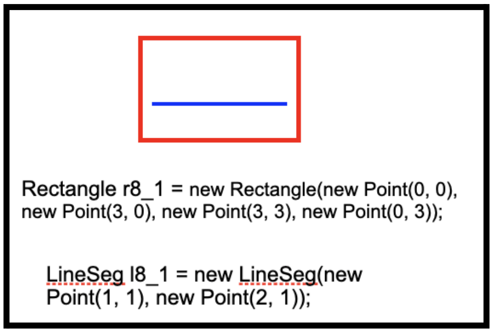


LineSeg l7_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

LineSeg l7_4 = new LineSeg(new Point(5, 5), new Point(0, 5));

Test of LineSeg.intersect(Rectangle) 8

Case 21: Line is completely in rectangle
Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));
LineSeg l8_1 = new LineSeg(new Point(1, 1), new Point(2, 1));
Expected Result: true
Code Result: true
Test Result: pass



Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_1 = new LineSeg(new Point(1, 1), new Point(2, 1));

Case 22: Line intersects one of rectangle's lines

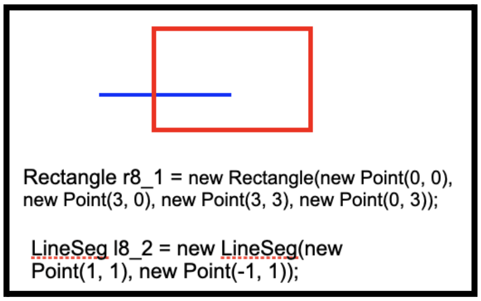
Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_2 = new LineSeg(new Point(1, 1), new Point(-1, 1));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_2 = new LineSeg(new Point(1, 1), new Point(-1, 1));

Case 23: Line is on one of rectangle's lines

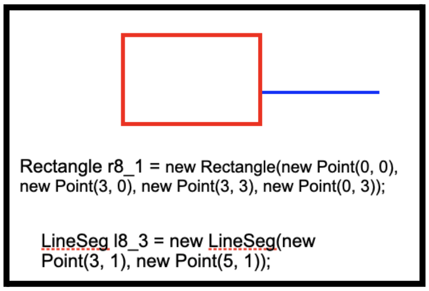
Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_3 = new LineSeg(new Point(3, 1), new Point(5, 1));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_3 = new LineSeg(new Point(3, 1), new Point(5, 1));

Case 24: Line intersects two lines from rectangle

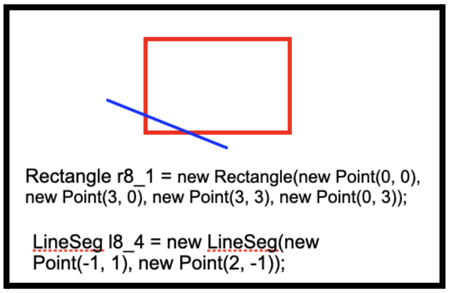
Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_4 = new LineSeg(new Point(-1, 1), new Point(2, -1));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_4 = new LineSeg(new Point(-1, 1), new Point(2, -1));

Case 25: Line is not in rectangle

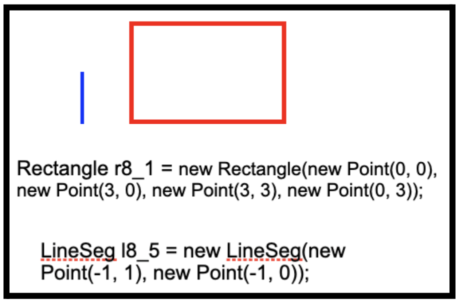
Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_5 = new LineSeg(new Point(-1, 1), new Point(-1, 0));

Expected Result: false

Code Result: false

Test Result: pass



Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_5 = new LineSeg(new Point(-1, 1), new Point(-1, 0));

Test of Circle.intersect(Point) 9

Case 26: Point is in circle

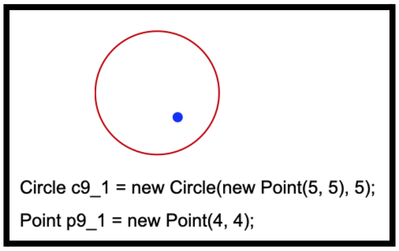
Circle c9_1 = new Circle(new Point(5, 5), 5);

Point p9_1 = new Point(4, 4);

Expected Result: true

Code Result: true

Test Result: pass



Circle c9_1 = new Circle(new Point(5, 5), 5);

Point p9_1 = new Point(4, 4);

Case 27: Point is on circle's edge

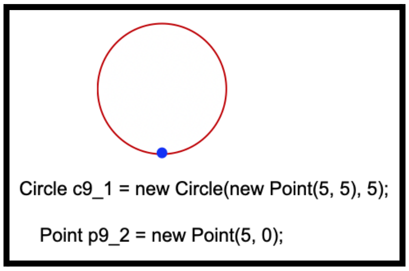
Circle c9_1 = new Circle(new Point(5, 5), 5);

Point p9_2 = new Point(5, 0);

Expected Result: true

Code Result: true

Test Result: pass



Circle c9_1 = new Circle(new Point(5, 5), 5);

Point p9_2 = new Point(5, 0);

Case 28: Point is outside circle

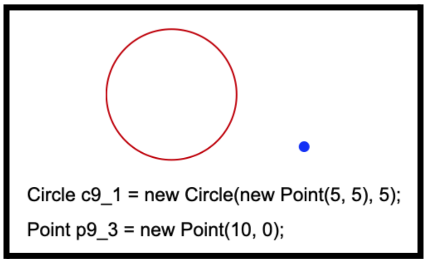
Circle c9_1 = new Circle(new Point(5, 5), 5);

Point p9_3 = new Point(10, 0);

Expected Result: false

Code Result: false

Test Result: pass



```
Circle c9_1 = new Circle(new Point(5, 5), 5);
Point p9_3 = new Point(10, 0);
```

Test of Circle.intersect(Circle) 10

Case 29: Circle is completely inside another circle

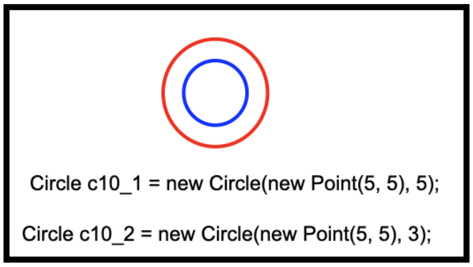
Circle c10_1 = new Circle(new Point(5, 5), 5);

Circle c10_2 = new Circle(new Point(5, 5), 3);

Expected Result: true

Code Result: true

Test Result: pass



```
Circle c10_1 = new Circle(new Point(5, 5), 5);
Circle c10_2 = new Circle(new Point(5, 5), 3);
```

Case 30: Circle intersects with another circle

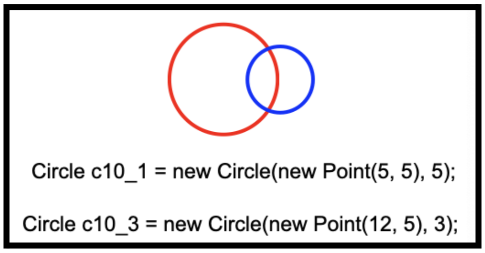
Circle c10_1 = new Circle(new Point(5, 5), 5);

Circle c10_3 = new Circle(new Point(12, 5), 3);

Expected Result: true

Code Result: true

Test Result: pass



```
Circle c10_1 = new Circle(new Point(5, 5), 5);
Circle c10_3 = new Circle(new Point(12, 5), 3);
```

Case 31: Circles edges touch

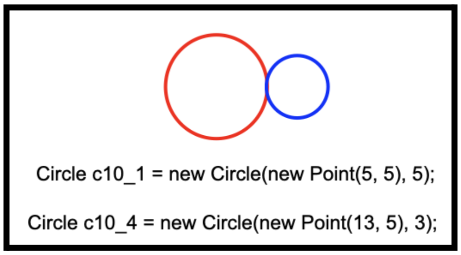
Circle c10_1 = new Circle(new Point(5, 5), 5);

Circle c10_4 = new Circle(new Point(13, 5), 3);

Expected Result: true

Code Result: true

Test Result: pass



Circle c10_1 = new Circle(new Point(5, 5), 5);

Circle c10_4 = new Circle(new Point(13, 5), 3);

Case 32: Circles do not intersect

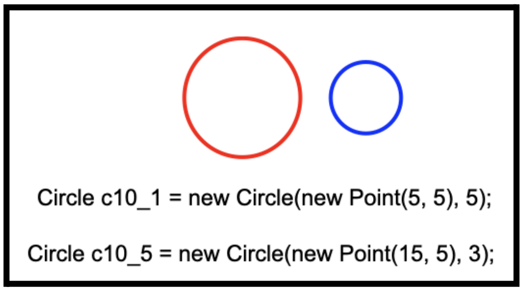
Circle c10_1 = new Circle(new Point(5, 5), 5);

Circle c10_5 = new Circle(new Point(15, 5), 3);

Expected Result: false

Code Result: false

Test Result: pass



Circle c10_1 = new Circle(new Point(5, 5), 5);

Circle c10_5 = new Circle(new Point(15, 5), 3);

Test of Circle.intersect(LineSeg) 11

Case 33: Line intersects circle

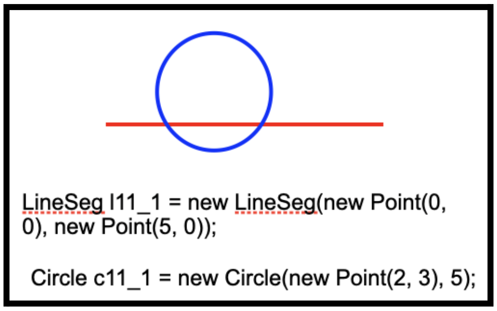
Circle c11_1 = new Circle(new Point(2, 3), 5);

LineSeg l11_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Expected Result: true

Code Result: true

Test Result: pass



LineSeg l11_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Circle c11_1 = new Circle(new Point(2, 3), 5);

Case 34: Line and circle edge touching

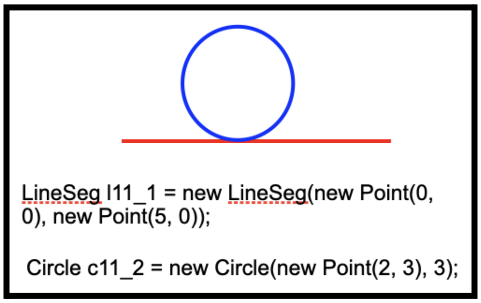
Circle c11_2 = new Circle(new Point(2, 3), 3);

LineSeg l11_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Expected Result: true

Code Result: true

Test Result: pass



```
LineSeg l11_1 = new LineSeg(new Point(0,
0), new Point(5, 0));

Circle c11_2 = new Circle(new Point(2, 3), 3);
```

Case 35: Line and circle do not intersect

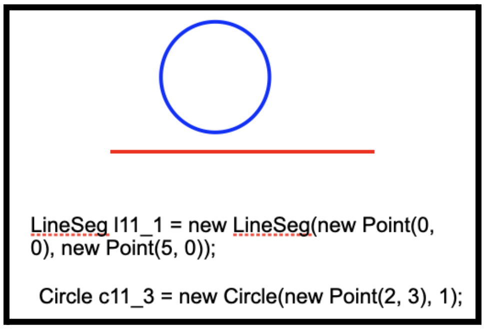
Circle c11_3 = new Circle(new Point(2, 3), 1);

LineSeg l11_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Expected Result: false

Code Result: false

Test Result: pass



```
LineSeg l11_1 = new LineSeg(new Point(0,
0), new Point(5, 0));

Circle c11_3 = new Circle(new Point(2, 3), 1);
```

Case 36: Line is completely in circle

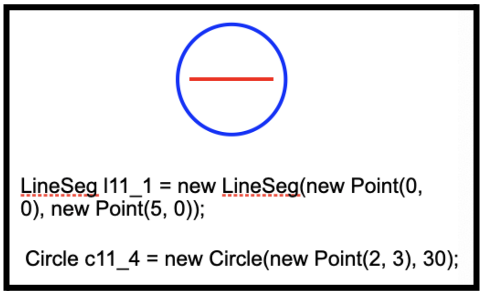
Circle c11_4 = new Circle(new Point(2, 3), 30);

LineSeg l11_1 = new LineSeg(new Point(0, 0), new Point(5, 0));

Expected Result: true

Code Result: true

Test Result: pass



```
LineSeg l11_1 = new LineSeg(new Point(0,
0), new Point(5, 0));

Circle c11_4 = new Circle(new Point(2, 3), 30);
```

Test of Circle.intersect(Rectangle) 12

Case 37:`Circle is in rectangle`

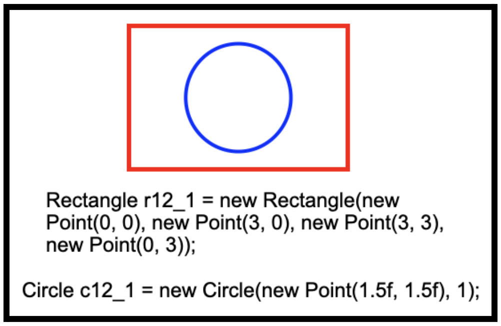
Circle c12_1 = new Circle(new Point(1.5f, 1.5f), 1);

Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: true

Code Result: true

Test Result: pass



```
Rectangle r12_1 = new Rectangle(new
Point(0, 0), new Point(3, 0), new Point(3, 3),
new Point(0, 3));

Circle c12_1 = new Circle(new Point(1.5f, 1.5f), 1);
```

Case 38: `Rectangle is in circle`

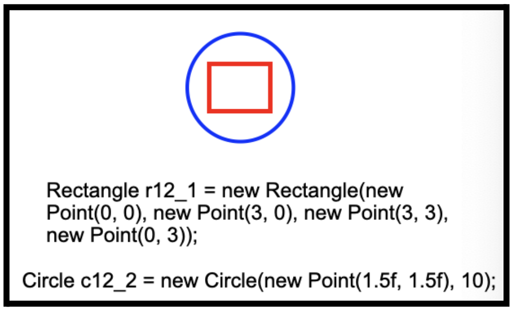
Circle c12_2 = new Circle(new Point(1.5f, 1.5f), 10);

Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: true

Code Result: true

Test Result: pass



```
Rectangle r12_1 = new Rectangle(new
Point(0, 0), new Point(3, 0), new Point(3, 3),
new Point(0, 3));

Circle c12_2 = new Circle(new Point(1.5f, 1.5f), 10);
```

Case 39: `Circle intersects with rectangle`

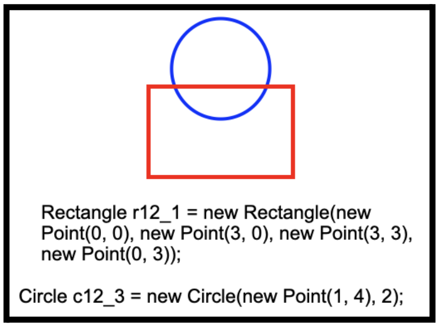
Circle c12_3 = new Circle(new Point(1, 4), 2);

Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: true

Code Result: true

Test Result: pass


Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));
Circle c12_3 = new Circle(new Point(1, 4), 2);

Case 40: `Circle's and rectangle's edge touch`

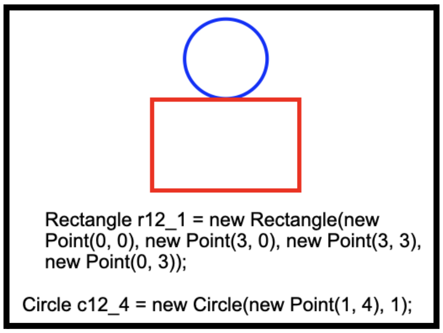
Circle c12_4 = new Circle(new Point(1, 4), 1);

Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: true

Code Result: true

Test Result: pass


Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));
Circle c12_4 = new Circle(new Point(1, 4), 1);

Case 41: Circle and rectangle do not intersect

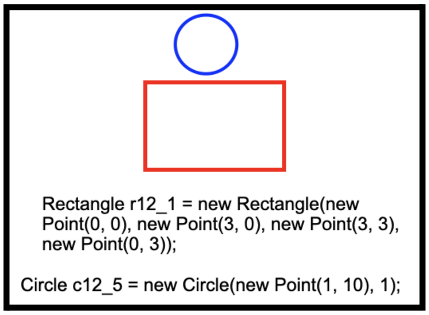
Circle c12_5 = new Circle(new Point(1, 10), 1);

Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: false

Code Result: false

Test Result: pass



Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Circle c12_5 = new Circle(new Point(1, 10), 1);

Test of Rectangle.intersect(Point) 13

Case 42: Point is in rectangle

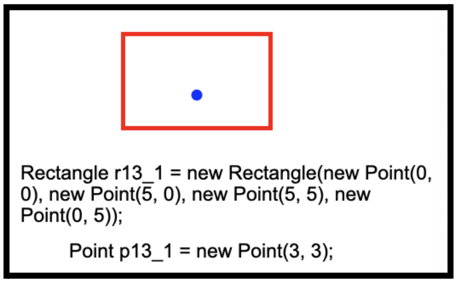
Rectangle r13_1 = new Rectangle(new Point(0, 0), new Point(5, 0), new Point(5, 5), new Point(0, 5));

Point p13_1 = new Point(3, 3);

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r13_1 = new Rectangle(new Point(0, 0), new Point(5, 0), new Point(5, 5), new Point(0, 5));

Point p13_1 = new Point(3, 3);

Case 43: `Point is on rectangle line`

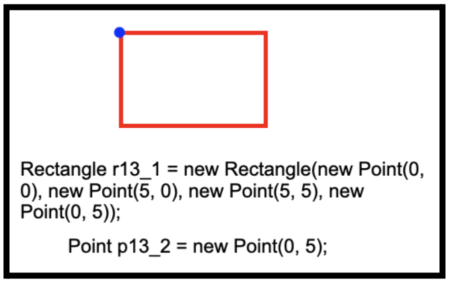
Rectangle r13_1 = new Rectangle(new Point(0, 0), new Point(5, 0), new Point(5, 5), new Point(0, 5));

Point p13_2 = new Point(0, 5);

Expected Result: true

Code Result: true

Test Result: pass



```
Rectangle r13_1 = new Rectangle(new Point(0,
0), new Point(5, 0), new Point(5, 5), new
Point(0, 5));
    Point p13_2 = new Point(0, 5);
```

Case 44: `Point is outside rectangle`

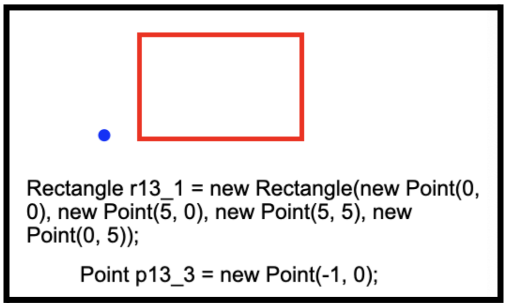
Rectangle r13_1 = new Rectangle(new Point(0, 0), new Point(5, 0), new Point(5, 5), new Point(0, 5));

Point p13_3 = new Point(-1, 0);

Expected Result: false

Code Result: false

Test Result: pass



```
Rectangle r13_1 = new Rectangle(new Point(0,
0), new Point(5, 0), new Point(5, 5), new
Point(0, 5));
    Point p13_3 = new Point(-1, 0);
```

Test of Rectangle.intersect(Circle) 14

Case 45: Circle is in rectangle

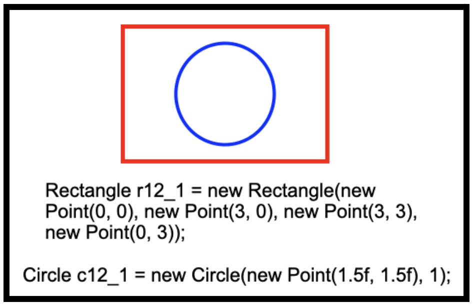
Circle c14_1 = new Circle(new Point(1.5f, 1.5f), 1);

Rectangle r14_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Circle c12_1 = new Circle(new Point(1.5f, 1.5f), 1);

Case 46: Rectangle is in circle

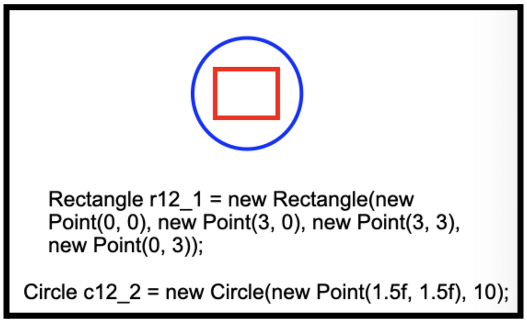
Circle c14_2 = new Circle(new Point(1.5f, 1.5f), 10);

Rectangle r14_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Circle c12_2 = new Circle(new Point(1.5f, 1.5f), 10);

Case 47: Circle intersects with rectangle

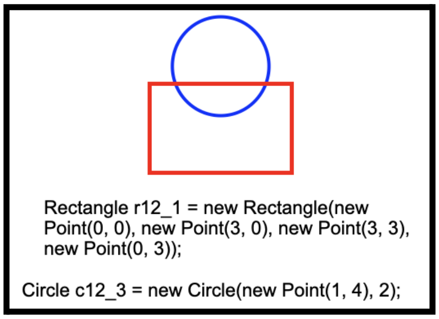
Circle c14_3 = new Circle(new Point(1, 4), 2);

Rectangle r14_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Circle c12_3 = new Circle(new Point(1, 4), 2);

Case 48: Circle's and rectangle's edge touch

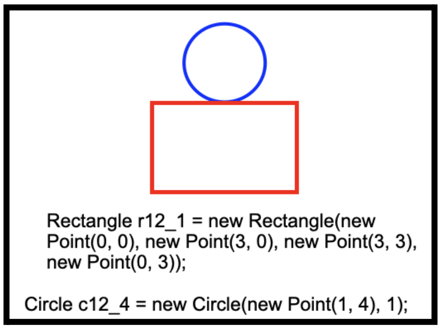
Circle c14_4 = new Circle(new Point(1, 4), 1);

Rectangle r14_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Circle c12_4 = new Circle(new Point(1, 4), 1);

Case 49: Circle and rectangle do not intersect

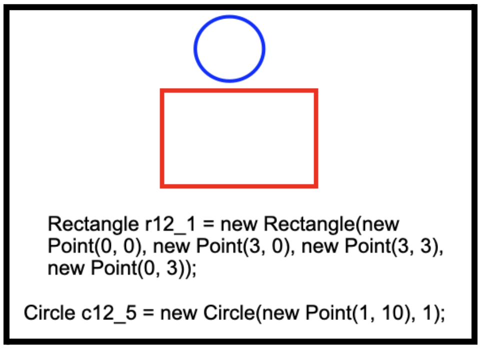
Circle c14_5 = new Circle(new Point(1, 10), 1);

Rectangle r14_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Expected Result: false

Code Result: false

Test Result: pass



Rectangle r12_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Circle c12_5 = new Circle(new Point(1, 10), 1);

<u>Test of Rectangle.intersect(LineSeg) 15</u>

Case 50: Line is completely in rectangle

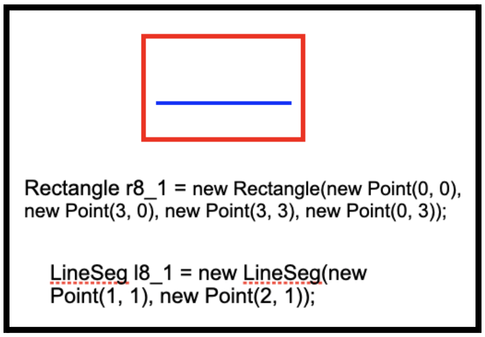
Rectangle r15_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l15_1 = new LineSeg(new Point(1, 1), new Point(2, 1));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r8_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l8_1 = new LineSeg(new Point(1, 1), new Point(2, 1));

Case 51: `Line intersects one of rectangle's lines`

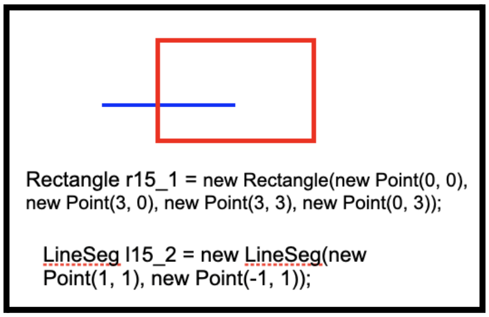
Rectangle r15_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l15_2 = new LineSeg(new Point(1, 1), new Point(-1, 1));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r15_1 = new Rectangle(new Point(0, 0),
new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l15_2 = new LineSeg(new
Point(1, 1), new Point(-1, 1));

Case 52: `Line is on one of rectangle's lines`

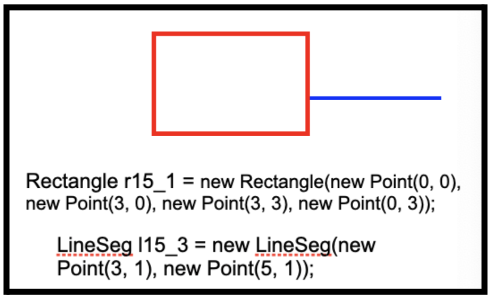
Rectangle r15_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l15_3 = new LineSeg(new Point(3, 1), new Point(5, 1));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r15_1 = new Rectangle(new Point(0, 0),
new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l15_3 = new LineSeg(new
Point(3, 1), new Point(5, 1));

Case 53: `Line intersects two lines from rectangle`

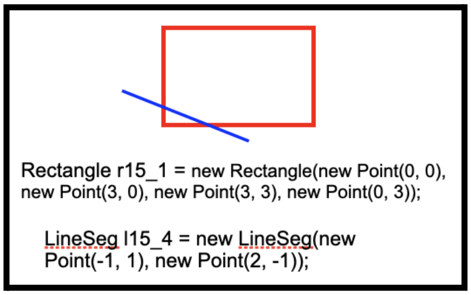
Rectangle r15_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l15_4 = new LineSeg(new Point(-1, 1), new Point(2, -1));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r15_1 = new Rectangle(new Point(0, 0),
new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l15_4 = new LineSeg(new
Point(-1, 1), new Point(2, -1));

Case 54: Line is not in rectangle

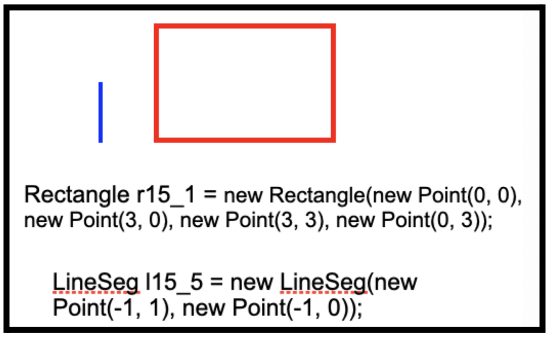
Rectangle r15_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l15_5 = new LineSeg(new Point(-1, 1), new Point(-1, 0));

Expected Result: false

Code Result: false

Test Result: pass



Rectangle r15_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

LineSeg l15_5 = new LineSeg(new Point(-1, 1), new Point(-1, 0));

Test of Rectangle.intersect(Rectangle) 16

Case 55: Rectangle is inside rectangle

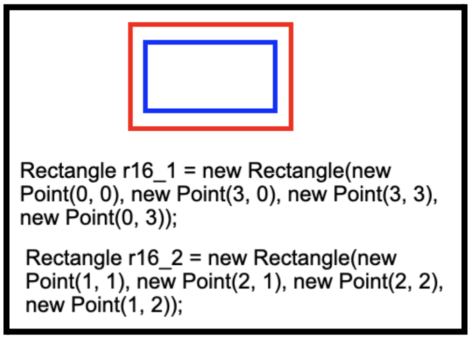
Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_2 = new Rectangle(new Point(1, 1), new Point(2, 1), new Point(2, 2), new Point(1, 2));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_2 = new Rectangle(new Point(1, 1), new Point(2, 1), new Point(2, 2), new Point(1, 2));

Case 56: Rectangle's corner is inside rectangle

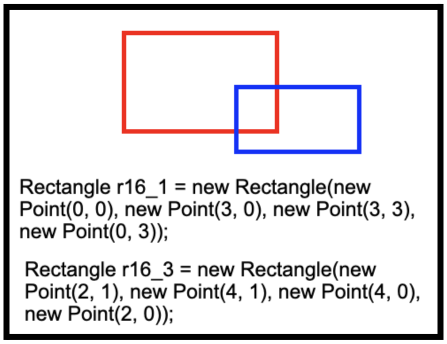
Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_3 = new Rectangle(new Point(2, 1), new Point(4, 1), new Point(4, 0), new Point(2, 0));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_3 = new Rectangle(new Point(2, 1), new Point(4, 1), new Point(4, 0), new Point(2, 0));

Case 57: Rectangle touching edge with other rectangle

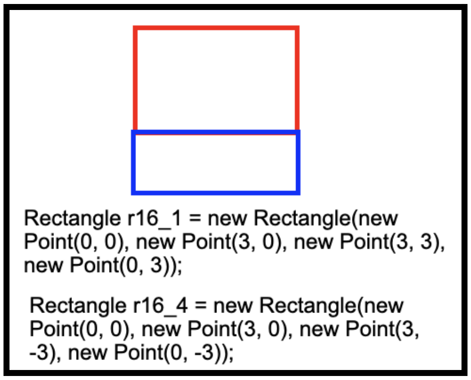
Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_4 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, -3), new Point(0, -3));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_4 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, -3), new Point(0, -3));

Case 58: Rectangle's side intersects two sides of other rectangle (slanted rectangle)

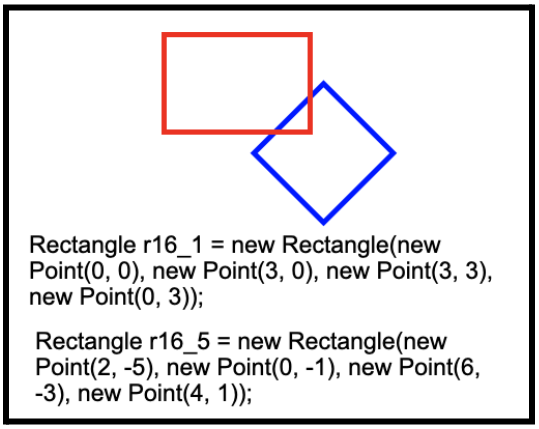
Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_5 = new Rectangle(new Point(2, -5), new Point(0, -1), new Point(6, -3), new Point(4, 1));

Expected Result: true

Code Result: true

Test Result: pass



Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_5 = new Rectangle(new Point(2, -5), new Point(0, -1), new Point(6, -3), new Point(4, 1));

Case 59: Rectangle does not intersect with other rectangle

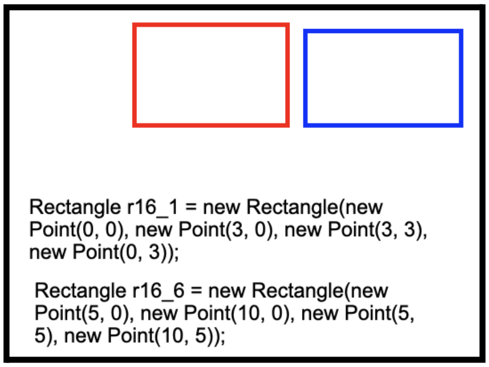
Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_6 = new Rectangle(new Point(5, 0), new Point(10, 0), new Point(5, 5), new Point(10, 5));

Expected Result: false

Code Result: false

Test Result: pass



Rectangle r16_1 = new Rectangle(new Point(0, 0), new Point(3, 0), new Point(3, 3), new Point(0, 3));

Rectangle r16_6 = new Rectangle(new Point(5, 0), new Point(10, 0), new Point(5, 5), new Point(10, 5));