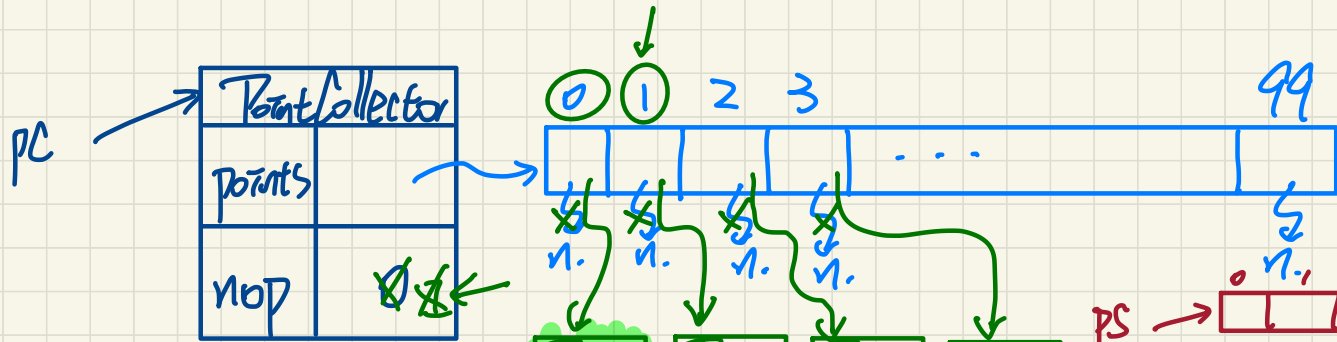


In-Lab Demo - Sep. 18

Programming Pattern

Reference-Typed Arrays
Integer Counter
Breakpoints & Debugger



count = 1

~~3~~
~~4~~
4

Point	Point	Point	Point
x 3	x -3	x -3	x 3
y 4	y 4	y 4	y -4

$(-3, 4)$	$(3, 4)$
$(-3, -4)$	$(3, -4)$

0
1
points

Programming Pattern

```
public class PointCollector {
    private Point[] points;
    private int nop; /* number of points */

    public PointCollector() {
        this.points = new Point[100];
    }

    public void addPoint(double x, double y) {
        this.points[this.nop] = new Point(x, y);
        this.nop++;
    }

    public int getNumberOfPoints() {
        return this.nop;
    }

    public Point[] getPointsInQuadrantI() {
        Point[] ps = new Point[this.nop];
        int count = 0;

        /* number of points in Quadrant I */
        for(int i = 0; i < this.nop; i++) {
            Point p = this.points[i];
            if(p.getX() > 0 && p.getY() > 0) {
                ps[count] = p; count++;
            }
        }
        Point[] q1Points = new Point[count];

        /* ps contains null if count < nop */
        for(int i = 0; i < count; i++) {
            q1Points[i] = ps[i];
        }
        return q1Points;
    }
}
```

of q1 points

count += 1

```
@Test
public void test() {
    PointCollector pc = new PointCollector();
    assertEquals(0, pc.getNumberOfPoints());

    pc.addPoint(3, 4);
    assertEquals(1, pc.getNumberOfPoints());

    pc.addPoint(-3, 4);
    assertEquals(2, pc.getNumberOfPoints());

    pc.addPoint(-3, -4);
    assertEquals(3, pc.getNumberOfPoints());

    pc.addPoint(3, -4);
    assertEquals(4, pc.getNumberOfPoints());

    Point[] ps = pc.getPointsInQuadrantI();
    assertEquals(1, ps.length);
    assertTrue(ps[0].getX() == 3 && ps[0].getY() == 4);
}
```