Compilation: javac CollisionSystem.java
Execution: java CollisionSystem N
(N random particles)
java CollisionSystem < input.txt
(from a file)

Creates N random particles and simulates their motion according
to the laws of elastic collisions.

// redraw all particles
private void redraw(double limit) {
    StdDraw.clear();
    for (int i = 0; i < particles.length; i++) {
        particles[i].draw();
    }
    StdDraw.show(20);
}

public void simulate(double limit) {
    // initialize PQ with collision events and redraw event
    pq = new MinPQ<Event>();
    new Event(0, null, null);

    // physical collision, update positions, and then simulation clock
    double dt = a.timeToHit(particles[i]);
    if (t + dt <= limit) pq.insert(new Event(t + dt, a, particles[i]));
}

...
/** An event during a particle collision simulation. Each event contains
 * the time at which it will occur (assuming no supervening actions)
 * and the particles a and b involved.
 */

private class Event implements Comparable<Event> {
    private final double time; // when event is scheduled to occur
    private final Particle a, b; // particles involved in event,
                                 // possibly null
    private final int countA, countB; // collision counts at event creation

    // create a new event to occur at time t involving a and b
    public Event(double t, Particle a, Particle b) {
        this.time = t;
        this.a    = a;
        this.b    = b;
        if (a != null) countA = a.count();
        else countA = -1;
        if (b != null) countB = b.count();
        else countB = -1;
    }

    // compare times when two events will occur
    public int compareTo(Event that) {
        if (this.time < that.time) return -1;
        else if (this.time > that.time) return +1;
        else return 0;
    }

    // has any collision occurred between when event was created and now?
    public boolean isValid() {
        if (a != null && a.count() != countA) return false;
        if (b != null && b.count() != countB) return false;
        return true;
    }

    // create collision system and simulate
    CollisionSystem system = new CollisionSystem(particles);
    system.simulate(10000);
}