```java
import java.util.Random;

public class MonteCarloPi {

private static int totalPoints = 0;
    private static int pointsInCircle = 0;

    public static void main(String[] args) throws InterruptedException {
        int numThreads = 4; // number of threads to use
        int numPoints = 1000000; // number of points to generate per thread
        Thread[] threads = new Thread[numThreads];

        // create and start threads
        for (int i = 0; i < numThreads; i++) {
            threads[i] = new Thread(new PointGenerator(numPoints));
            threads[i].start();
        }

        // wait for threads to finish
        for (int i = 0; i < numThreads; i++) {
            threads[i].join();
        }

        // calculate and output pi estimate
        double pi = 4.0 * pointsInCircle / totalPoints;
        System.out.println("Estimated value of pi: " + pi);
    }

    private static class PointGenerator implements Runnable {
        private int numPoints;
        private Random rand = new Random();

        public PointGenerator(int numPoints) {
            this.numPoints = numPoints;
        }

        public void run() {
            for (int i = 0; i < numPoints; i++) {
                double x = rand.nextDouble();
                double y = rand.nextDouble();

                if (x * x + y * y <= 1.0) {
                    pointsInCircle++;
                }
                totalPoints++;
            }
        }
    }
}
```