```
7  public class KMeans {
8    static final double EPSILON = 0.01;
9    static List<Point> points;
10   static Map<Integer, Point> centers;
11   public static void main(String[] args) {
12     points = KMeans.readFile("cluster.dat");
13     centers = randomDistinctCenters(points);
14     double convergence = 1.0;
15     while (convergence > EPSILON) {
16       Map<Integer, List<Point>> clusters = points
17               .stream()
18               .collect(
19                       Collectors.groupingBy(p -> KMeans.closestCenter(centers, p))
20               );
21       Map<Integer, Point> newCenters = clusters
22               .entrySet()
23               .stream()
24               .collect(
25                       Collectors.toMap(
26                               e -> e.getKey(),
27                               e -> Point.barycenter(e.getValue())
28                       )
29               );
30       convergence = distance(centers, newCenters);
31       centers = newCenters;
32     }
33     displayResults(clusters, centers);
34   }
```

FIGURE 17.13 Stream-based KMeans application: aggregate data.