```
9   static class RecursiveWordCountTask extends RecursiveTask<Map<String, Long>> {
10    final int THRESHOLD = ...;
11    Spliterator<String> rightSplit;
12
13    RecursiveWordCountTask(Spliterator<String> aSpliterator) {
14      rightSplit = aSpliterator;
15    }
16    protected Map<String, Long> compute() {
17      Map<String, Long> result = new HashMap<>();
18      Spliterator<String> leftSplit;
19      if (rightSplit.estimateSize() > THRESHOLD
20            && (leftSplit = rightSplit.trySplit()) != null) {
21        RecursiveWordCountTask left = new RecursiveWordCountTask(leftSplit);
22        RecursiveWordCountTask right = new RecursiveWordCountTask(rightSplit);
23        left.fork();
24        right.compute().forEach(
25              (k, v) -> result.merge(k, v, (x, y) -> x + y)
26        );
27        left.join().forEach(
28              (k, v) -> result.merge(k, v, (x, y) -> x + y)
29        );
30      } else {
31        rightSplit.forEachRemaining(
32              word -> result.merge(word, 1L, (x, y) -> x + y)
33        );
34      }
35      return result;
36    }
37  }
```

FIGURE 17.15 The RecursiveWordCountTask class.