```
1  public class MatrixMultiply {
2    static final int N = ...;
3    static double[][] matrixA;
4    static double[][] matrixB;
5    static class RowColumn {
6      int row;
7      int col;
8      RowColumn(int aRow, int aCol) {
9        row = aRow;
10       col = aCol;
11     }
12     public boolean equals(Object anObject) {
13       RowColumn other = (RowColumn) anObject;
14       return (this.row == other.row && this.col == other.col);
15     }
16   }
17   public static void main(String[] args) {
18     vector = readMatrix("matrixA.dat");
19     matrix = readMatrix("matrixB.dat");
20     MapReduce<RowColumn, RowColumn, Double, Double> mapReduce = new MapReduce<>();
21     List<RowColumn> inputList = new ArrayList<>(N * N);
22     for (int i = 0; i < N; i++) {
23       for (int j = 0; j < N; j++) {
24         inputList.add(new RowColumn(i, j));
25       }
26     }
27     mapReduce.setInput(inputList);
28     mapReduce.setMapperSupplier(MatrixMultiply.Mapper::new);
29     mapReduce.setReducerSupplier(MatrixMultiply.Reducer::new);
30     Map<RowColumn, Double> output = mapReduce.call();
31     displayOutput(output);
32   }
33   // Exercise: missing mapper and reducer classes?
34   ...
35 }
```

**FIGURE 17.17** The MatrixMultiply class used in Exercise 17.4.