```
1  class MMThread {
2    double[][] lhs, rhs, prod;
3    int n;
4    public MMThread(double[][] lhs, double[][] r
5      n = lhs.length;
6      this.lhs = lhs;
7      this.rhs = rhs;
8      this.prod = new double[n][n];
9    }
10   void multiply() {
11     Worker[][] worker = new Worker[n][n];
12     for (int row = 0; row < n; row++) {
13       for (int col = 0; col < n; col++) {
14         worker[row][col] = new Worker(row,col);
15       }
16     }
17     for (int row = 0; row < n; row++) {
18       for (int col = 0; col < n; col++) {
19         worker[row][col].start();
20       }
21     }
22     for (int row = 0; row < n; row++) {
23       for (int col = 0; col < n; col++) {
24         try {
25           worker[row][col].join();
26         } catch (InterruptedException ex) {
27         }
28       }
29     }
30   }
```

FIGURE 16.1 The MMThread task: matrix multiplication using threads.