

```

1/**
2 * A library of basic matrix operations.
3 */
4public class MatrixOps {
5    /**
6     * Returns the matrix resulting from adding the two given matrices,
7     * or null if the matrices don't have the same dimensions.
8     */
9    public static int[][] add(int[][] m1, int[][] m2) {
10        int m1Rows = m1.length;
11        int m1Columns = m1[0].length;
12        int m2Rows = m2.length;
13        int m2Columns = m2[0].length;
14        String message = " " + m1Columns + " " + m2Rows;
15        System.out.println(message);
16        if (m1Columns != m2Columns || m2Rows != m1Rows) {
17            return null;
18        }
19
20        int[][] newMatrix = new int[m1Rows][m1Columns];
21        for (int i = 0; i < m1Rows; i++) {
22            for (int j = 0; j < m1Columns; j++) {
23                newMatrix[i][j] = m1[i][j] + m2[i][j];
24            }
25        }
26        return newMatrix;
27    }
28
29    /**
30     * Returns a unit matrix of the given size.
31     * A unit matrix of size N is a square N x N matrix that contains 0's
32     * in all its cells, except that the cells in the diagonal contain 1.
33     */
34    public static int[][] unit(int n) {
35        int[][] imatrix;
36        imatrix = new int[n][n];
37        for (int row = 0; row < n; row++) {
38            for (int col = 0; col < n; col++) {
39                if (row == col) {
40                    imatrix[row][col] = 1;
41                } else {
42                    imatrix[row][col] = 0;
43                }
44            }
45        }
46        return imatrix;
47    }
48
49    /**
50     * Returns the matrix resulting from multiplying the two matrices,
51     * or null if they have incompatible dimensions.
52     */
53    public static int[][] mult(int[][] m1, int[][] m2) {
54        int m1Rows = m1.length;
55        int m1Columns = m1[0].length;
56        int m2Rows = m2.length;
57        int m2Columns = m2[0].length;
58        int[][] product = new int[m1Rows][m2Columns];
59
60        if (m1Columns != m2Rows) {
61            return null;
62        }
63
64        for (int i = 0; i < m1Rows; i++) {
65            for (int j = 0; j < m2Columns; j++) {
66                for (int k = 0; k < m1Columns; k++) {
67                    product[i][j] += m1[i][k] * m2[k][j];
68                }
69            }
70        }
71
72        return product;
73    }
74
75    /**
76     * Returns a matrix which is the transpose of the given matrix.
77     */
78    public static int[][] transpose(int[][] m) {
79        int mRows = m.length;
80        int mColumns = m[0].length;
81        int transpose[][] = new int[mColumns][mRows];
82
83        for (int i = 0; i < mColumns; i++) {
84            for (int j = 0; j < mRows; j++) {
85                transpose[i][j] = m[j][i];
86            }
87        }
88
89        return transpose;
90    }
91
92    /**
93     * Prints the given matrix, and then prints an empty line.
94     */
95    public static void println(int[][] m) {
96        for (int row = 0; row < m.length; row++) {
97            for (int col = 0; col < m[0].length; col++) {
98                System.out.print(m[row][col] + " ");
99            }
100            System.out.println();
101        }
102        System.out.println();
103    }
104
105    /**
106     * Tests all the matrix operations featured by this class.
107     */
108    public static void main(String args[]) {
109        int[][] a = { { 1, 2, 1 },
110                     { 0, 1, 1 },
111                     { 2, 0, 1 } };
112
113        int[][] b = { { 1, 0, 2 },
114                     { 1, 2, 0 },
115                     { 2, 0, 1 } };
116
117        System.out.println("Matrix A:"); println(a);
118        System.out.println("Matrix B:"); println(b);
119
120        System.out.println("A + B:"); println(add(a, b));
121        System.out.println("B + A:"); println(add(b, a));
122        System.out.println("I (a unit matrix of size 3):"); println(unit(3));
123
124        int[][] c = { { 1, 2, 3 },
125                     { 4, 5, 6 },
126                     { 2, 0, 1 } };
127        System.out.println("A * B:"); println(mult(a, b));
128        System.out.println("A * I:"); println(mult(a, unit(3)));
129
130        System.out.println("Matrix C:"); println(c);
131        System.out.println("C, transposed:"); println(transpose(c));
132    }
133}

```