```
1   import java.util.Random;
2   import java.util.Scanner;
3
4   public class MatrixArithmetic
5   {
6     public static void main (String[] args)
7     {
8       boolean run = true;
9       Scanner scan = new Scanner(System.in);
10      String[] menuOptions = {
11        "Add 2 matrices",
12        "Subtract 2 matrices",
13        "Multiply 2 matrices",
14        "Multiply matrix by a constant",
15        "Transpose matrix",
16        "Matrix trace"
17      };
18
19      do
20      {
21        int selection = menu(menuOptions);
22
23        int size;
24        int[][] matrix1, matrix2;
25        switch(selection)
26        {
27          case 0:
28            run = false;
29            break;
30
31          case 1:
32            size = getInt(scan,"Enter the size of the square matrices: ", "Not an
    Integer! Try Again!");
33            matrix1 = genMatrix(size,1,10);
34            matrix2 = genMatrix(size,1,10);
35
36            System.out.println("First matrix is:");
37            printMatrix(matrix1);
38
39            System.out.println("Second matrix is:");
40            printMatrix(matrix2);
41
42            System.out.println("The resulting matrix is:");
43            printMatrix(addMatrix(matrix1, matrix2));
44            break;
45
46          case 2:
47            size = getInt(scan,"Enter the size of the square matrices: ", "Not an
    Integer! Try Again!");
48            matrix1 = genMatrix(size,1,10);
49            matrix2 = genMatrix(size,1,10);
50
51            System.out.println("First matrix is:");
52            printMatrix(matrix1);
53
54            System.out.println("Second matrix is:");
55            printMatrix(matrix2);
56
57            System.out.println("The resulting matrix is:");
58            printMatrix(subtractMatrix(matrix1,matrix2));
59            break;
60
61          case 3:
62            size = getInt(scan,"Enter the size of the square matrices: ", "Not an
    Integer! Try Again!");
63            matrix1 = genMatrix(size,1,10);
64            matrix2 = genMatrix(size,1,10);
65
66            System.out.println("First matrix is:");
67            printMatrix(matrix1);
```

```
68
69            System.out.println("Second matrix is:");
70            printMatrix(matrix2);
71
72            System.out.println("The resulting matrix is:");
73            printMatrix(multiplyMatrix(matrix1,matrix2));
74            break;
75
76          case 4:
77            size = getInt(scan,"Enter the size of the square matrix: ", "Not an
    Integer! Try Again!");
78            matrix1 = genMatrix(size,1,10);
79            int constant = getInt(scan, "Enter the multiplication constant", "Not an
    Integer! Try Again!");
80
81            System.out.println("The matrix is:");
82            printMatrix(matrix1);
83
84            System.out.println("The original matrix multiplied by " + constant + "
    is:");
85            printMatrix(multiplyMatrix(matrix1,constant));
86            break;
87
88          case 5:
89            size = getInt(scan,"Enter the size of the square matrix: ", "Not an
    Integer! Try Again!");
90            matrix1 = genMatrix(size,1,10);
91
92            System.out.println("The matrix is:");
93            printMatrix(matrix1);
94
95            System.out.println("The transposed matrix is:");
96            printMatrix(transposeMatrix(matrix1));
97            break;
98
99          case 6:
100           size = getInt(scan,"Enter the size of the square matrix: ", "Not an
    Integer! Try Again!");
101           matrix1 = genMatrix(size,1,10);
102
103           System.out.println("The matrix is:");
104           printMatrix(matrix1);
105
106           System.out.print("The trace for this matrix is: ");
107           System.out.println(traceMatrix(matrix1));
108           break;
109       }
110       if (selection != 0)
111       {
112         System.out.printf("%64s\n" ,"Command Number " + selection + " completed.
    ");
113         System.out.println();
114       }
115     } while (run);
116     System.out.println("Testing complete");
117   }
118
119   public static int[][] genMatrix(int size, int minVal, int maxVal)
120   {
121     Random rand = new Random();
122     int[][] matrix = new int[size][size];
123
124     for (int i = 0; i < matrix.length; i++)
125     {
126       for (int j = 0; j < matrix[i].length; j++)
127       {
128         matrix[i][j] = rand.nextInt(maxVal - minVal + 1) + minVal;
129       }
130     }
131
```

```
132     return matrix;
133   }
134
135   public static int[][] addMatrix(int[][] matrix1, int[][] matrix2)
136   {
137     int[][] sumMatrix = new int[matrix1.length][matrix1.length];
138     if (matrix1.length == matrix2.length)
139     {
140       for (int i = 0; i < matrix1.length; i++)
141       {
142         for(int j = 0; j < matrix1[i].length; j++)
143         {
144           sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
145         }
146       }
147     }
148     return sumMatrix;
149   }
150
151   public static int[][] subtractMatrix(int[][] matrix1, int[][] matrix2)
152   {
153     int[][] diffMatrix = new int[matrix1.length][matrix1.length];
154     if (matrix1.length == matrix2.length)
155     {
156       for (int i = 0; i < matrix1.length; i++)
157       {
158         for(int j = 0; j < matrix1[i].length; j++)
159         {
160           diffMatrix[i][j] = matrix1[i][j] - matrix2[i][j];
161         }
162       }
163     }
164     return diffMatrix;
165   }
166
167   public static void printMatrix(int[][] matrix)
168   {
169     for(int i = 0; i < matrix.length; i++)
170     {
171       for (int j = 0; j < matrix[i].length; j++)
172       {
173         System.out.printf("%5d", matrix[i][j]);
174       }
175       System.out.println();
176     }
177   }
178
179   public static int[][] transposeMatrix(int[][] matrix1)
180   {
181     int[][] movedMatrix = new int[matrix1.length][matrix1.length];
182     for(int i = 0; i < matrix1.length; i++)
183     {
184       for(int j = 0; j < matrix1.length; j++)
185       {
186         movedMatrix[j][i] = matrix1[i][j];
187       }
188     }
189     return movedMatrix;
190   }
191
192   public static int[][] multiplyMatrix(int[][] matrix1, int[][] matrix2)
193   {
194     int[][] productMatrix = new int[matrix1.length][matrix1.length];
195     int length = matrix1.length;
196     int sum = 0;
197     for (int i = 0; i < length; i++)
198     {
199       for (int j = 0; j < length; j++)
200       {
201         for (int l = 0; l < length; l++)
```

```
202         {
203           sum += matrix1[i][l] * matrix2[l][j];
204         }
205         productMatrix[i][j] = sum;
206         sum = 0;
207       }
208     }
209     return productMatrix;
210   }
211
212   public static int traceMatrix(int[][] matrix)
213   {
214     int sum = 0;
215     for (int i = 0; i < matrix.length; i++)
216     {
217       sum += matrix[i][i];
218     }
219     return sum;
220   }
221
222   public static int[][] multiplyMatrix(int[][] matrix, int constant)
223   {
224     int[][] newMatrix = new int[matrix.length][matrix.length];
225     for (int i = 0; i < matrix.length; i++)
226     {
227       for (int j = 0; j < matrix.length; j++)
228       {
229         newMatrix[i][j] = constant * matrix[i][j];
230       }
231     }
232     return newMatrix;
233   }
234
235   public static int menu(String[] options)
236   {
237     int selection;
238     boolean validOption = false;
239     Scanner scan = new Scanner(System.in);
240     do
241     {
242       System.out.println("Your options are:");
243       System.out.println("-----------------");
244
245       for (int i = 0; i < options.length; i++)
246         System.out.printf("\t%s\n", (i + 1) + ") " + options[i]);
247       System.out.printf("\t%s\n", "0) EXIT");
248
249       selection = getInt(scan, "Please enter your option: ", "Not an integer! Try
    again! Please enter your option:");
250       if (selection > -1 && selection <= options.length)
251         validOption = true;
252       else
253         System.out.println();
254
255     } while(!validOption);
256
257     return selection;
258   }
259
260   public static int getInt(Scanner scan, String promptMsg, String invalidMsg)
261   {
262     System.out.print(promptMsg);
263     while(!scan.hasNextInt())
264     {
265       scan.next();
266       System.out.print(invalidMsg);
267     }
268
269     return scan.nextInt();
270   }
```

```
271
272
273 }
```