FloydWarshall.java

```
1 import java.util.Scanner;
 3 public class FloydWarshall {
       private int matrix[7][7];
 5
       private int numVertices;
       public static final int INFINITY = 999;
 6
 7
 8
       public FloydWarshall(int vertices) {
           matrix = new int[vertices + 1][vertices + 1];
 9
10
           this.numVertices = vertices;
11
       }
12
13
       public void floydwarshall(int mat[][]) {
14
           for (int s = 1; s <= numVertices; s++) {</pre>
15
               for (int d = 1; d <= numVertices; d++) {</pre>
16
                    matrix[s][d] = mat[s][d];
17
               }
18
           }
19
           for (int i = 1; i <= numVertices; i++) { // i = intermediate</pre>
20
               for (int s = 1; s <= numVertices; s++) {</pre>
21
                    for (int d = 1; d <= numVertices; d++) {</pre>
22
                        if (matrix[s][i] + matrix[i][d] < matrix[s][d])</pre>
23
                            matrix[s][d] = matrix[s][i] + matrix[i][d];
24
                   }
               }
25
26
           }
27
28
           for (int s = 1; s <= numVertices; s++) {</pre>
               System.out.print("\t" + s);
29
30
31
           System.out.println();
           for (int s = 1; s <= numVertices; s++) {</pre>
32
33
               System.out.print(s + "\t");
34
               for (int d = 1; d <= numVertices; d++) {</pre>
                    System.out.print(matrix[s][d] + "\t");
35
36
37
               System.out.println();
38
           }
39
       }
40
41
       public static void main(String... arg) {
```

FloydWarshall.java

```
42
           int matrix[][];
43
           int numVertices;
           Scanner scan = new Scanner(System.in);
44
45
           System.out.println("Number of vertices:");
46
           numVertices = scan.nextInt();
47
           matrix = new int[numVertices + 1][numVertices + 1];
48
           System.out.println("Weighted matrix");
49
           for (int s = 1; s <= numVertices; s++) {</pre>
50
               for (int d = 1; d <= numVertices; d++) {</pre>
51
                   matrix[s][d] = scan.nextInt();
52
                   if (s == d) {
                       matrix[s][d] = 0;
53
54
                       continue;
55
                   }
                   if (matrix[s][d] == 0) {
56
57
                       matrix[s][d] = INFINITY;
                   }
58
59
               }
60
          }
61
          System. out. println("The Transitive Closure of the Graph");
          FloydWarshall floydwarshall = new FloydWarshall
62
  (numVertices);
63
           floydwarshall.floydwarshall(matrix);
64
           scan.close();
65
      }
66 }
```