

```
1 class Graph
2 {
3
4     String[] knoten;
5     int anzahlKnoten;
6     int[][] kanten;
7
8     Graph()
9     {
10         knoten = new String[20];
11         kanten = new int[20][20];
12         anzahlKnoten = 0;
13     }
14
15     boolean neuerKnoten(String bezeichnung)
16     {
17         if (anzahlKnoten < 20) {
18             int position = anzahlKnoten;
19             knoten[position] = bezeichnung;
20
21             anzahlKnoten++;
22             return true;
23         } else {
24             return false;
25         }
26     }
27
28     boolean neueKante(String knotenA, String knotenB,
29                         int wert)
30     {
31         int a = knotenSuchen(knotenA);
32         int b = knotenSuchen(knotenB);
33
34         if (a > -1 && b > -1) {
35             kanten[a][b] = wert;
36             return true;
37         }
38
39         return false;
40     }
41 }
```

```
42 int kanteAusgeben(String knotenA, String knotenB)
43 {
44     int a = knotenSuchen(knotenA);
45     int b = knotenSuchen(knotenB);
46
47     if (a > -1 && b > -1) {
48         return kanten[a][b];
49     } else {
50         return -1;
51     }
52 }
53
54 int knotenSuchen(String bezeichnung)
55 {
56     for(int i = 0; i<anzahlKnoten; i++) {
57         if (knoten[i].equals(bezeichnung)){
58             return i;
59         }
60     }
61
62     return -1;
63 }
64
65 void ausgeben() {
66     System.out.println("Adjazenzmatrix:");
67
68     for (int x = 0; x < anzahlKnoten; x++) {
69         for (int y = 0; y < anzahlKnoten; y++) {
70             System.out.print(kanten[x][y]);
71             System.out.print(", ");
72         }
73         System.out.println();
74     }
75 }
76
77 }
```

```
1 class Graph
2 {
3
4     String[] knoten;
5     int anzahlKnoten;
6     int[][] kanten;
7
8     Graph()
9     {
10         knoten = new String[20];
11         kanten = new int[20][20];
12         anzahlKnoten = 0;
13     }
14
15     boolean neuerKnoten(String bezeichnung)
16     {
17         if (anzahlKnoten < 20) {
18             int position = anzahlKnoten;
19             knoten[position] = bezeichnung;
20
21             anzahlKnoten++;
22             return true;
23         } else {
24             return false;
25         }
26     }
27
28     boolean neueKante(String knotenA, String knotenB,
29                         int wert)
30     {
31         int a = knotenSuchen(knotenA);
32         int b = knotenSuchen(knotenB);
33
34         if (a > -1 && b > -1) {
35             kanten[a][b] = wert;
36             return true;
37         }
38
39         return false;
40     }
41 }
```

```
42 int kanteAusgeben(String knotenA, String knotenB)
43 {
44     int a = knotenSuchen(knotenA);
45     int b = knotenSuchen(knotenB);
46
47     if (a > -1 && b > -1) {
48         return kanten[a][b];
49     } else {
50         return -1;
51     }
52 }
53
54 int knotenSuchen(String bezeichnung)
55 {
56     for(int i = 0; i<anzahlKnoten; i++) {
57         if (knoten[i].equals(bezeichnung)){
58             return i;
59         }
60     }
61
62     return -1;
63 }
64
65 void ausgeben() {
66     System.out.println("Adjazenzmatrix:");
67
68     for (int x = 0; x < anzahlKnoten; x++) {
69         for (int y = 0; y < anzahlKnoten; y++) {
70             System.out.print(kanten[x][y]);
71             System.out.print(", ");
72         }
73         System.out.println();
74     }
75 }
76
77 }
```