

```

import java.util.ArrayList;
class GRAPHMATRIX{
    private ArrayList<KNOTEN> knotenfeld;
    private ArrayList<ArrayList<Integer>> matrix;

    GRAPHMATRIX() {
        knotenfeld = new ArrayList<KNOTEN>();
        matrix = new ArrayList<ArrayList<Integer>>();
    }
}

```

Lösung ZurückSetzen:

```

void zurückSetzen() {
    knotenfeld.clear();
    matrix.clear();
}

```

Lösung KnotenGeben:

Methode die einen Text erwartet, um einen Knoten zurückzugeben
 Schau dir das gesamte Array an
 Wenn der Bezeichner des Knotens gleich dem Text ist
 Gib diesen Knoten zurück
 Sonst
 Gib einen leeren Knoten zurück
 Ende Schauen
 Ende Methode

```

KNOTEN KnotenGeben(String b) {
    for(int i=0; i<knotenfeld.size();i++){
        if (knotenfeld.get(i).bezeichnerGeben().equals(b)) {
            return knotenfeld.get(i);
        }else{
            return null;
        }
    }
}

```

Oder:

```

KNOTEN KnotenGeben(String b) {
    for(KNOTEN k: knotenfeld){
        if (k.bezeichnerGeben().equals(b)) {
            return k;
        }else{
            return null;
        }
    }
}

```

Lösung KnotenNummerGeben:

Methode die einen Text erwartet, um eine Position zurückzugeben
Mache beginnend beim ersten bis zum letzten Knoten folgendes:
Wenn Bezeichner von Knoten an bestimmter Pos. gleich Text
Gebe Index dieses Knotens zurück
*Ende Wenn
Gib -1 zurück
Ende Machen
Ende Methode

```
int KnotenNummerGeben(String b){
    for(int i=0; i < knotenfeld.size(); i++){
        if (knotenfeld.get(i).BezeichnerGeben().equals(b)){
            return i;
        }
    }
    return -1;
}
```

Lösung KnotenBezeichnerGeben:

Methode die Bezeichner des Knotens an bestimmter Stelle zurückgibt
Wenn die Stelle Sinn macht (innerhalb möglicher Werte)
Gib Bezeichner des Knotens an der angegebenen Stelle zurück
Sonst
Gib leeren Text zurück
Ende Wenn
Ende Methode

```
String KnotenBezeichnerGeben(int knr){
    if((knr < knotenfeld.size()) && (knr >=0)){
        return knotenfeld.get(knr).BezeichnerGeben();
    }else{
        return "";
    }
}
```

Lösung KanteEinfügen:

Methode die Kanten von A nach B mit Gewichtung einfügt
Lass dir zu beiden Knoten dir Nr geben
Wenn es beide Knoten gibt und diese nicht identisch sind
 Trage bei Matrix(v,n) die Gewichtung ein
 Trage bei Matrix(n,v) die Gewichtung ein
Ende Wenn
Ende Methode

```
void KanteEinfügen(String vk, String nk, int gewicht){
    int vn = KnotenNummerGeben(vk);
    int nn = KnotenNummerGeben(nk);
    if((vn!=-1) && (nn!=-1) && (vn!=nn)) {
        matrix.get(vn).set(nn,gewicht);
        matrix.get(nn).set(vn,gewicht);
    }
}
```

Lösung KantenGewichtGeben:

Methode die zu Strings A+B Gewicht einer eventuellen Kante liefert
Lass dir zu beiden Knoten dir Nr geben
Wenn es beide Knoten gibt
 Gib Matrix(v,n) zurück
Sonst
 Gib -1 zurück
Ende Wenn
Ende Methode

```
int KantenGewichtGeben(String vk, String nk){
    int vn = KnotenNummerGeben(vk);
    int nn = KnotenNummerGeben(nk);
    if((vn!=-1) && (nn!=1)) {
        return matrix.get(vn).get(nn);
    }else{
        return -1;
    }
}
```

Lösung KnotenEinfügen:

| | | | | | |
|---|----|----|----|----|----|
| | 0 | 1 | 2 | 3 | 4 |
| 0 | | | | | -1 |
| 1 | | | | | -1 |
| 2 | | | | | -1 |
| 3 | | | | | -1 |
| 4 | -1 | -1 | -1 | -1 | -1 |

```
Methode die neuen Knoten mit anzugebendem Bezeichner einfügt
  Füge in die Knotenliste einen neu zu erstellenden Knoten ein
  Füge der Matrix (aus ArrayListen) eine weitere ArrayList dazu
  Mache für alle Zeilen der Matrix
    solange die Zeile kürzer ist als die Knotenliste
      füge dort -1 ein
    *Ende solange
  *Ende Mache
*Ende Methode
```

Lösung KnotenEinfügen - kurz:

```
void KnotenEinfügen(String bezeichner) {
    knoten.add(new KNOTEN(bezeichner));
    matrix.add(new ArrayList<Integer>());
    for (ArrayList<Integer> a: matrix){
        while(a.size() < knoten.size()){
            a.add(new Integer(-1));
        }
    }
}
```

Lösung KnotenEinfügen - lang:

Methode die neuen Knoten mit anzugebendem Bezeichner einfügt

Merke dir die alte Matrixgröße

Füge in die Knotenliste einen neu zu erstellenden Knoten ein

Mache für alle Zeilen der Matrix

Lass dir die Zeile geben und füge dort -1 ein

*Ende Für alle

Füge der Matrix (aus ArrayListen) eine weitere ArrayList dazu

Mache für diese letzte ArrayList in der Matrix

Füge -1er ein (einen mehr als alte Matrixgröße)

*Ende Mache

*Ende Methode

```
void KnotenEinfügen(String bezeichner){
    int anzAlt = matrix.size();
    knoten.add(new KNOTEN(bezeichner));
    for (int ix = 0; ix < anzAlt; ix++){
        matrix.get(ix).add(new Integer(-1));
    }
    matrix.add(new ArrayList<Integer>());
    for (int ix = 0; ix <= anzAlt; ix++){
        matrix.get(anzAlt).add(new Integer(-1));
    }
}
```

```
public class KNOTEN{
    private String bezeichner;

    KNOTEN(String bez){
        bezeichner = bez;
    }

    String bezeichnerGeben(){
        return bezeichner;
    }
}
```