Name:

Matrikelnummer:

Bearbeitungszeit: 90 min.

Aufgabe 1. Gegeben ist die Klasse `Schottergrube` mit den Methoden

```
public double getVerfuegbareMenge()
```

und

```
public double getPreis(),
```

die die verfügbare Menge in Tonnen und den Preis pro Tonne zurückgeben.

Schreiben Sie eine Klasse `Disponent` mit einem parameterlosen Konstruktor und der Methode

```
public Map<Schottergrube,Double> getBestellungen(
       double benoetigteMenge, Schottergrube[] schottergruben),
```

die in einer Map die bei den einzelnen Schottergruben bestellte Menge zurückgibt, sodass die benötigte Menge — evt. aufgeteilt auf mehrere Schottergruben — möglichst günstig bestellt wird. In der zurückgelieferten Map sollen nur jene Schottergruben enthalten sein, bei denen tatsächlich bestellt wird. Bei jeder Schottergrube in `schottergruben` kann nicht mehr als die verfügbare Menge bestellt werden.

Aufgabe 2. Gegeben ist die Klasse `Tourenplanung` mit dem Konstruktor

```
public Tourenplanung(double[][] dist)
```

und die Methoden

```
public double getDist(int i, int j),
public void setDist(int i, int j, double d),
public ArrayList<Integer> getRoute(int[] orte).
```

Der Konstruktor initialisiert die Tourenplanung mit der Entfernungstabelle zwischen den Orten. Die Orte sind durch die Nummern `0,...,dist.length-1` bezeichnet, und `dist[i][j]` ist die direkte Entfernung vom Ort `i` zum Ort `j`. Die Methoden `getDist(i,j)` und `setDist(i,j,d)` geben die Entfernung zwischen zwei Orten zurück bzw. verändern diese. Die Methode `getRoute(orte)` gibt die kürzeste Route zurück, die bei `orte[0]` beginnt, bei `orte[orte.length-1]` endet, und alle Orte in `orte` zumindest einmal besucht. Die berechnete Route wird als Liste der in dieser Reihenfolge zu besuchenden Orte zurückgegeben.

Weiters ist das Interface `Baustellen` mit der Methode

```
public void setGesperrt(boolean[][] gesperrt)
```

gegeben. Mit dieser Methode soll es möglich sein anzugeben (durch setzen von `gesperrt[i][j]==true`), welche direkten Verbindungen zwischen Orten `i` und `j` gesperrt sind.

Schreiben Sie eine Unterklasse `TourenplanungMitBaustellen` von `Tourenplanung`, die das Interface `Baustellen` implementiert, und über den Konstruktor

```
public TourenplanungMitBaustellen(
       double[][] dist, boolean[][] gesperrt),
```

und die Methode

```
public ArrayList<Integer> getRouteMitBaustellen(
       int[] orte, boolen[][] tempGesperrt).
```

verfügt. Der Konstruktor soll die Entfernungstabelle initialisieren und das Angeben von gesperrten direkten Verbindungen erlauben. Die Methode `getRouteMitBaustellen(orte,tempGesperrt)` soll analog zu `getRoute(orte)` die kürzeste Route (unter Berücksichtigung der gesperrten direkten Verbindungen) berechnen, wobei in `tempGesperrt` noch zusätzliche, nur für diese Berechnung temporär gesperrte direkte Verbindungen angegeben werden können.