

## Übungsaufgabe Graphen

In dieser Aufgabe soll Ihnen die grundlegende Funktionsweise von Graphen nähergebracht werden. Hierfür werden Sie Knoten und Kanten einfügen, löschen, prüfen, ob Knoten oder Kanten existieren und den Wert von mehreren Kanten aufaddieren.

Die Aufgabe baut auf dem Beispiel eines Straßennetzes auf. Die Knoten werden hierbei durch die Klasse `Crossroad` und die Kanten durch die Klasse `OneWayStreet` dargestellt. `Crossroad` hat eine Liste aller Objekte von `OneWayStreet`, welche von dieser `Crossroad` zu einer anderen führen. Die Klasse `Graph` hat dann eine Liste aller vorhandenen `Crossroad`-Objekte. Die Listen sind dabei in Form der `LinkedList` implementiert.

Machen Sie sich zunächst mit den Klassen `Crossroad` und `OneWayStreet` vertraut, auch wenn es in diesen Klassen keine Aufgaben zu bearbeiten gibt.

Alle Aufgaben befinden sich in der Klasse `Graph`.

Ihre Methoden können Sie in der Klasse `RoadSystem` testen. Natürlich können Sie sich auch eigene Testfälle schreiben.

Viel Spaß und Erfolg beim Programmieren!

### Aufgabe 1 `addCrossroad`

Die Methode `addCrossroad` soll ein neues `Crossroad`-Objekt erstellen und es der Liste `crossroads` hinzufügen. Die `id` des neuen Objektes soll dabei `globalId` sein.

Anschließend soll `globalId` um eins erhöht werden. So ist sichergestellt, dass jede `Crossroad` über eine einzigartige `id` verfügt.

### Aufgabe 2 `findCrossroad`

Die Methode `findCrossroad(int id)` soll durch die Liste `crossroads` durchgehen und das Objekt mit der korrekten `id` zurückgeben. Ist kein Objekt mit der übergebenen `id` vorhanden, soll `null` zurückgegeben werden.

### Aufgabe 3 `addOneWayStreet`

Die Methode `addOneWayStreet` soll ein neues `OneWayStreet`-Objekt erzeugen, wobei `targetCrossroadId` den Ziel Knoten angibt und `distance` die Entfernung der beiden `Crossroads`. Die `Crossroad` mit der `startCrossroadId` soll dann das neue Objekt der Liste `oneWayStreets` hinzufügen.

Falls `targetCrossroadId` oder `startCrossroadId` keinen `Crossroad`-Objekten zugeordnet werden kann, soll die Methode nichts machen.

### Aufgabe 4 findOneWayStreet

Die Methode `findOneWayStreet` soll prüfen ob zwischen den angegebenen `Crossroads` eine Einbahnstraße existiert und falls ja soll `distance` der `OneWayStreet` zurückgegeben werden. Existiert ein solches `OneWayStreet`-Objekt nicht, soll -1 zurückgegeben werden.

### Aufgabe 5 removeOneWayStreet

Die Methode `removeOneWayStreet` soll die Einbahnstraße zwischen den beiden angegebenen `Crossroads` entfernen. Hierfür muss vom Startknoten die korrekte `OneWayStreet` aus der Liste `oneWayStreets` entfernt werden.

### Aufgabe 6 removeCrossroad

Die Methode `removeCrossroad` soll das angegebene `Crossroad`-Objekt aus dem Graphen entfernen. Hierfür müssen zunächst alle `OneWayStreets`, die mit dem `Crossroad`-Objekt verbunden sind gelöscht werden (unabhängig davon, in welche Richtung die Einbahnstraßen verlaufen). Anschließend muss das `Crossroad`-Objekt aus der Liste `crossroads` gelöscht werden.

### Aufgabe 7 totalDistance

Die Methode `totalDistance` soll das Attribut `distance` jedes im Graphen enthaltenen `OneWayStreet`-Objekts addieren und anschließen zurückgeben. Es soll also die Strecke aller Straßen addiert werden.