

Übung 10 – Graphen

Vervollständigen Sie die Klasse `graph.AdjListGraph`, die Graphen mit Hilfe von Verbindungslisten verwaltet. Die Klasse `graph.AdjListGraph` implementiert das Interface `graph.Graph` und ist von der Klasse `graph.AbstractGraph` abgeleitet. Diese Basisklasse implementiert folgende Methoden:

- `toString`: Ausgabe des Graphen in einen String
- `getNofVertices`: Anzahl der Knoten
- `getNofEdges`: Anzahl der Kanten
- `isDirected`: Abfrage, ob der Graph gerichtet oder ungerichtet ist. Diese Eigenschaft wird mit dem Konstruktor gesetzt.

Die beiden Methoden `getNofVertices` und `getNofEdges` sind aber ziemlich ineffizient implementiert. Überschreiben Sie diese Methoden in Ihrer Implementierung der Klasse `graph.AdjListGraph` mit effizienteren Implementierungen.

Zum Testen Ihrer Implementation steht ein Graphen-Editor zur Verfügung. Damit dieser Editor überhaupt funktioniert muss er wissen, wie der Graph beschaffen ist. Dazu verwendet er Ihre (in dieser Übung zu erstellende) Klasse `graph.AdjListGraph`! Den Editor benötigen Sie auch für die folgenden Übungen zum Thema Graphen.

Installation:

Laden Sie sich die Datei `graph.zip` von der Kurshomepage und entpacken Sie die Dateien. Beachten Sie, dass sämtliche Java-Klassen zum Paket `graph` gehören und daher im Verzeichnis `graph` abgelegt sind. Kompilieren Sie die `.java`-Dateien

```
javac graph\*.java
```

und starten Sie den Editor mit

```
java graph.GraphEditor.
```

Es passiert noch nichts, da die Klasse `graph.AdjListGraph` noch nicht ausprogrammiert ist.

Aufgaben:

Implementieren Sie eine Graphen Klasse

- Vervollständigen Sie die Implementation von `graph.AdjListGraph`. Beginnen Sie zuerst mit der Methode `addVertex` und `getVertices`. Wenn diese richtig funktionieren, dann sollten Sie die Knoten im Editor sehen können.
- Implementieren Sie danach auch noch `addEdge` und `getAdjacentVertices`. Jetzt sollten auch die Kanten sichtbar werden. Zeichnen Sie einige Graphen und vergewissern Sie sich, dass Ihre Implementierungen korrekt sind.
- Implementieren Sie nun `removeEdge` und optimieren Sie `getNofVertices` und `getNofEdges`.
- Programmieren Sie folgenden Konstruktor:

```
AdjListGraph(AdjListGraph original);
```

Dieser Konstruktor nimmt einen bestehenden Graphen der gleichen Klasse und baut eine Kopie davon auf. Solche Konstruktoren werden auch *copy-constructor* genannt. Testen Sie diesen Konstruktor indem Sie auf den Copy-Knopf drücken. Dabei wird die Methode `clone()` aufgerufen und die Kopie wird dargestellt.

Bedienung des Editors:

An der rechten Seite sind die wichtigsten Bedienelemente angeordnet. Es gibt 4 verschiedene Modi in denen sich der Editor befinden kann.

1. *Selection*
Wenn Sie einige Knoten (und Kanten) gezeichnet haben, können Sie in diesem Modus die Knoten verschieben und selektieren. Die Selektion von Knoten ist in dieser Übung noch nicht relevant.
2. *Insert Node*
Sie können in diesem Modus Knoten durch einfachen Mausklick hinzufügen. Mit jedem Mausklick wird ein neuer Knoten erzeugt. Ist zugleich die Funktion "Auto Name" eingeschaltet, dann wird jeder Knoten automatisch einen Namen erhalten (eine fortlaufende Nummer). Schalten Sie diese Option aus, dann werden Sie nach jedem Mausklick nach einem Namen gefragt.
3. *Insert Edge*
In diesem Modus können Sie nun Kanten einfügen, indem Sie zuerst auf den Ausgangsknoten und dann auf den Zielknoten klicken. Der Ausgangsknoten wird markiert solange der Zielknoten noch nicht angewählt wurde.
4. *Remove Edge:*
Funktioniert analog zum Insert Edge Modus, nur wird hier die Verbindung zwischen den Knoten gelöscht. Es wird dabei nicht geprüft, ob eine Verbindung bestanden hat oder nicht.

Mit *Print* können Sie den Graphen auf der Konsole ausgeben lassen (dient hauptsächlich zur Überprüfung des angezeigten Graphen)

Im *File*-Menu können Sie unter *New Graph* zwischen vier verschiedenen neuen Graphenarten wählen. Um welche Art Graph es sich gerade handelt, wird auch unten rechts in der Statuszeile angezeigt.

Mit *Save...* und *Open...* können Sie einen Graphen speichern, bzw. wieder laden.

Warnung!

Beim Graphen-Editor handelt es sich um eine einfache Implementation. Viele Funktionen sind nicht "wasserdicht" implementiert, d.h. es werden nicht alle Eventualitäten abgefangen und behandelt. Besondere Vorsicht ist geboten im Umgang mit dem "Auto Name". Es wird ein simpler Zähler hochgezählt. Nach dem Selberbenennen kann es zu Duplikaten kommen (die von Ihrer Graph-Klasse ignoriert werden sollten). Bei der Eingabe von Kantengewichten wird nicht überprüft ob es sich tatsächlich um Zahlen handelt.