

## Übungsblatt 1

Abgabe: 9. Mai 2007

### Zugang zum SVN-Repository:

Server: <https://i11svn.iti.uni-karlsruhe.de/praktika/>

Benutzer und Passwort bitte über Marco Gaertler oder Martin Holzer einrichten lassen. Kann auch per Email geschehen, in dem die Ausgabe des folgenden (Linux-)Kommandos an den Betreuer geschickt wird:

```
htpasswd2 -n <USERNAME>
```

wobei <USERNAME> für den Benutzernamen steht.

Checken Sie die Projekte `Libraries`, `examples` und `PriorityClustering` aus. Grundlegende Informationen, die für das weitere Projekt relevant sind, können aus dem Skript zur Netzwerkanalyse entnommen werden (eingescheckt).

### Einführung in die Graphenstruktur

Wir nutzen im Folgenden die Bibliothek *y-Files*. Achtung, diese Bibliothek ist noch Java 1.4 kompatibel und nutzt keine Generics! Die Basisklasse, welche Graphen repräsentiert, ist `y.base.Graph`. Die Erweiterung `y.view.Graph2D` fügt graphische Attribute hinzu. Knoten- und Kantenattribute können mittels den Klassen `y.base.NodeMap` und `y.base.EdgeMap` gespeichert werden. Zur Speicherung von Graphen wollen wir GraphML nutzen, welches mittels `yext.graphml.graph2D.GraphMLIOHandler` eingelesen bzw. geschrieben werden kann.

Schreiben Sie eine Klasse, welche Breitensuche (BFS) implementiert (Pseudo-Code im Netzwerkanalyse-Skript S. 46).

### Einführung in die Clusterungsstruktur

Wir nutzen die eigenentwickelte Bibliothek, die in `clusteringTools-fat.jar` zur Verfügung gestellt wird (Javadoc befindet sich ebenfalls im Projekt `Libraries`). Eine Clustering (`de.uka.algo.clustering.Clustering`) ist eine Partition der Knotenmenge in disjunkte, nicht-leere Teilmengen sogenannte Cluster (`de.uka.algo.clustering.Cluster`). Eine Clustering ist stets mit einem Graphen assoziiert und muß vor Benutzung explizit initialisiert werden, welches

entweder durch `resetSingletons()` (jeder Knoten bildet sein eigenes Cluster) oder `resetLargeCluster()` (alle Knoten sind im gleichen Cluster) geschieht. Schreiben Sie ein Programm, welches einen zufälligen Graphen mit zufälliger Clusterung erzeugt. Optimieren Sie anschließend die Clusterung bezüglich der Anzahl der Kanten innerhalb der Cluster, d.h. verschieben sie Knoten von einem Cluster in einen anderen, um die Anzahl der Intracluster-Kanten zu erhöhen ohne die Anzahl der Cluster zu verringern.