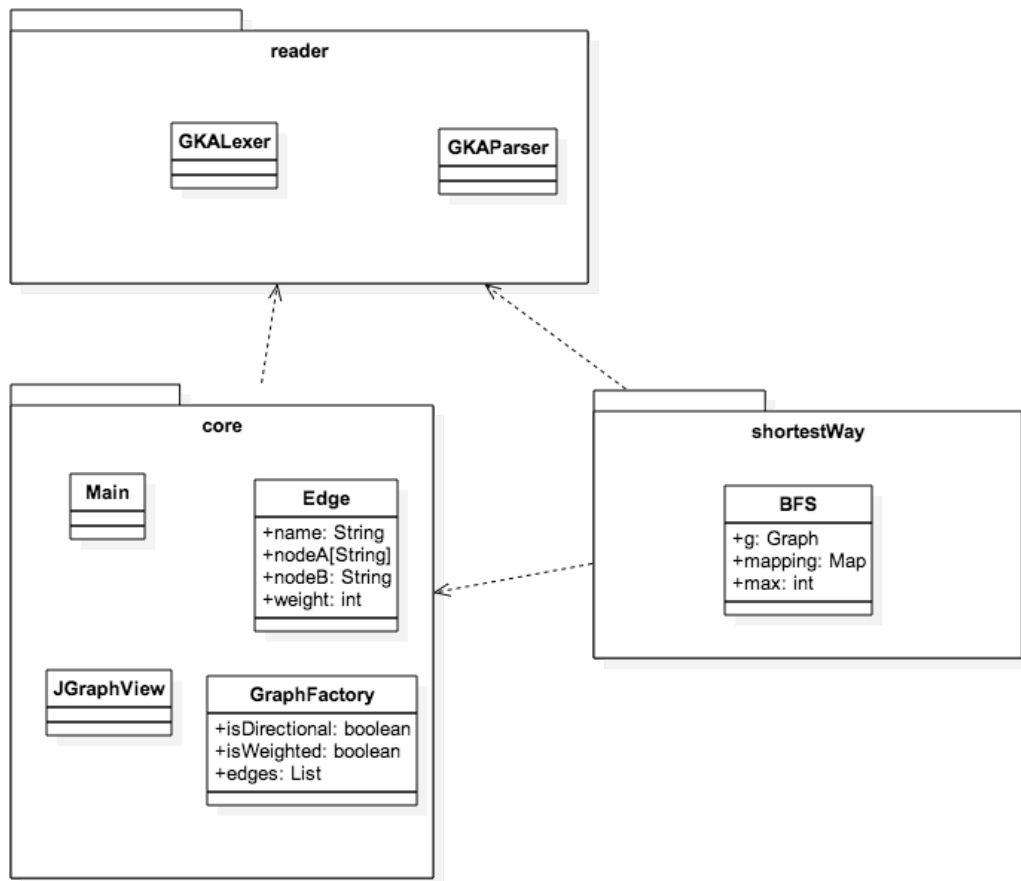


# Praktikum 1

Abgabe von : Moritz Spindelhirn



## Quellenangaben:

Die ANTLR Grammatik wurde mit Hilfe der offiziellen Dokumentation erstellt. Daraus wurde z.B. der Hinweis über die Verarbeitung für Leerzeichen übernommen.

Für die Erstellung der grafischen Oberfläche mit jGraphx wurden als Basis ein Beispiel aus dem jGraph Repository genutzt.

- <https://theantlrguy.atlassian.net/wiki/display/ANTLR4/ANTLR+4+Documentation>
- <https://github.com/jgrapht/jgrapht>

## Bearbeitungszeitraum:

---

5 Stunden + 3 Stunden + 2 Stunden

## Einlesen des Graphen

---

Die Graphdatei wird mit Hilfe eines über ANTLR2 erstellten Parser eingelesen. Die zu Grunde liegende Grammatik ist im Quellcode-Paket unter *antlr/gka.g4* zu finden.

Mit Hilfe von ANTLR2 wird das komplette Package *de.futjikato.gka.reader* automatisch generiert.

Folgende Regeln gelten für Graphdateien:

- Sobald eine Zeile eine Richtung angibt ist der gesamte Graph gerichtet.  
**Wichtig:** Für nicht gerichtete Kante werden zwei Kanten im Grapg erstellt. Jeweils eine für die Hin- und die Rückrichtung.
- Sobald eine Gewichtung in der HGraphdatei steht ist der gesamte Grapg gerichtet. Als Standartwert für die nicht gewichteten Zeilen wird in dem Fall "0" angenommen.

Folgende Klassen werden für Graphen genutzt :

Klasse	Gerichtet	Gewichtet
Pseudograph	-	-
DirectedPseudograph	x	-
WeightedPseudograph	-	x
DirectedWeightedPseudograph	x	x

## Darstellung von Graphen

---

Die Graphen werden in einem simplen JFrame mit Hilfe der jGraphX dargestellt. Aktuell wird für alle Graphen das *mxCircleLayout* verwendet.

Das Userinterface beinhaltet aktuell keine Möglichkeit eine Graphdatei auszuwählen. Diese wird über ein Programm Argument angegeben.

## Schreiben von Graphen

---

Die *GraphWriter* Klasse bietet die Methode *write* welche einen Dateinamen und einen Graphen als Parameter entgegennimmt und den Graphen im gka Dateiformat schreibt. Um alle Vertices zu schreiben werden erst alle Vertices geschrieben. Darauf folgen alle Edges. Auf Basis der Grapgklasse wird entschieden, ob die Graphen gerichtet sind.

## BFS

---

Der BFS Algorithmus ist als eigene ausführbare Klasse gebaut. Man kann diese Klasse mit 3 Parametern starten. Als erstes gibt man den Pfad zur .gka Datei an. Darauf folgen der Startknoten und der Zielknoten.

Als Resultat wird auf der Konsole das erstellte Mapping und der gefunde Pfad ausgegeben. In der grafischen Darstellung wird jeder Knoten in rot eingefärbt, um den Pfad zu visualisieren.