Codedokumentation des Spanning-Tree Projektes

Aufgabenstellung

Die Aufgabe des Labor Projektes Netztechnik ist es einen Spanning Tree auf Ebene 2 eines Netzwerkes zu simulieren. Hierfür wird ein Netzwerk über einen Input-File definiert. Dieses Netzwerk berechnet selbständig in der Simulation den besten Spanning Tree. Der berechnete Spanning Tree wird nach der Berechnung auf minimale Weise im Terminal angezeigt.

Import-File

Der Import File richtet sich nach den Vorgaben aus der Vereinbarung.
Dabei wird durch //Node-IDs die Deklaration der Knoten gestartet und durch //Connections die Deklaration der Verbindungen gestartet. Das sieht bspw. so aus:

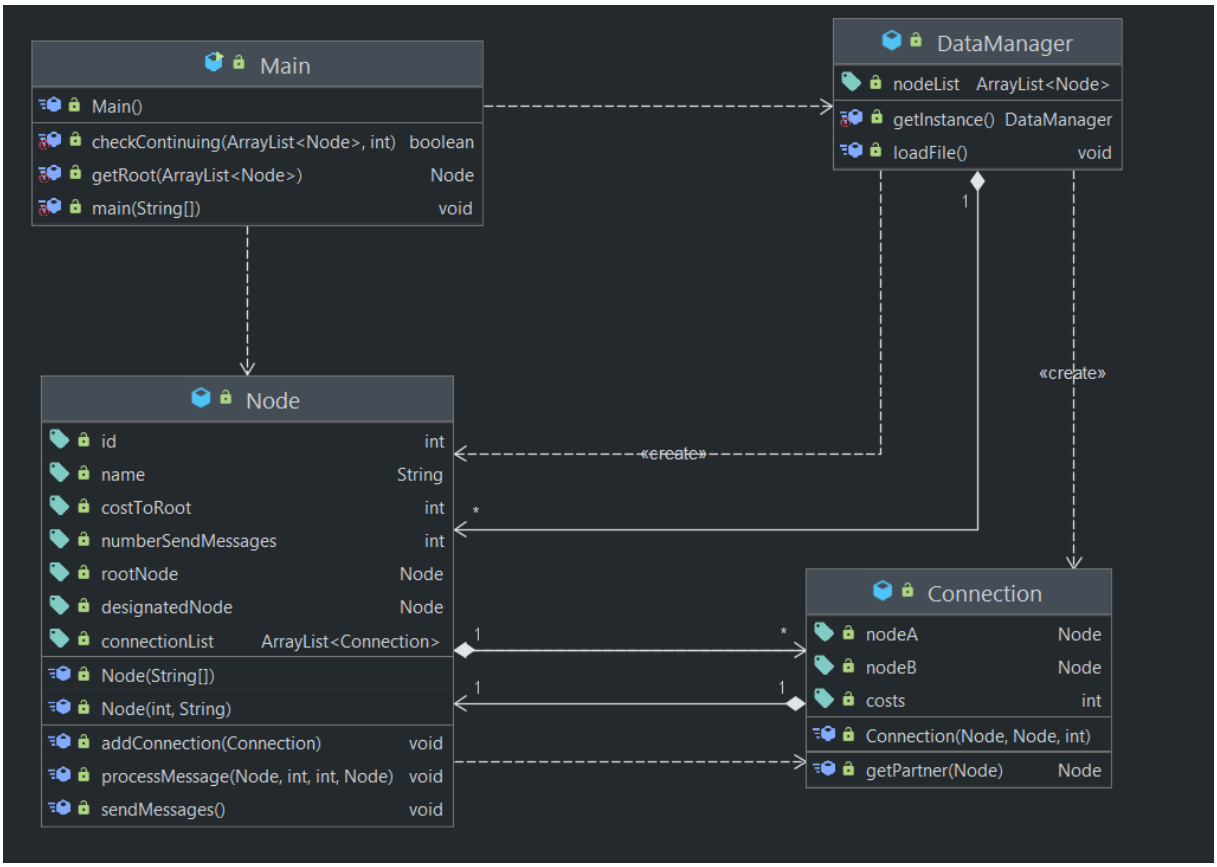
```
//Node-IDs
   A,5
   B,1
   C,3
   D,7
   E,6
   F,4
//Connections
   A,B,10
   A,C,10
   B,D,15
   B,E,10
   C,D,3
   C,E,10
   D,E,2
   D,F,10
   E,F,2
```

Damit der File eingelesen wird muss er import.txt heißen und im gleichen Ordner wie der src Ordner liegen.

Klassendiagramm

Klassen:

- Main
  Die Aufgabe der Main ist die Steuerung des Gesamt-Prozesses und der Ausgabe an den Nutzer
- DataManager
  Der DataManager kümmert sich um die Datenhaltung und den Import des Import Files
- Node
  Die Node Klasse spiegelt einen Netzwerkknoten da. Ein Netzwerk Knoten hat eine ID, einen Weg zum Root (Next Hop und Kosten) und mehrere verbundene Knoten
- Connection
  Eine Connection verbindet einen Knoten A mit einem Knoten B mit den Kosten x



Einlesen

Das Einlesen des Import Files wird durch den DataManager übernommen. Dabei wird ein FileReader und BufferReader eingesetzt. Diese lesen den File zeilenweise ein. Es wird für jede Zeile geschaut was der Inhalt ist. Hat eine Zeile „//Node-IDs“ dann wird diese Zeile nicht weiterverarbeitet und folgende Zeilen werden als Knoten verarbeitet. Das gleiche gilt analog für den „//Connections“ Tag für Connection. Des Weiteren werden alle „//“ Zeilen als Kommentar gesehen und nicht verarbeitet.

Berechnung

Die Berechnung beginnt mit einer Loop, die so lange sich wiederholt, bis jeder Knoten eine bestimmte Anzahl an Nachrichten gesendet hat. In dieser Loop wird ein Knoten zufällig ausgewählt. Mit diesem Knoten wird die Funktion sendMessages() aufgerufen. Diese sendet eine Nachricht an alle angebundene Knoten indem die Funktion die processMessage() aufruft. Diese Funktion überprüft, ob der sendende Knoten einen besseren Root bzw. besseren Weg zum Root hat. Sollte dies der Fall sein wird der next Hop auf den senden Knoten gesetzt und die Wegkosten und der Root geändert.

Ausgabe

Der Spanning Tree wird nach der Berechnung ausgegeben. Die Ausgabe orientiert sich an der Ausgabe des gegebenen Beispiels. Die Ausgabe sieht wie folgt aus:

```
Root: B
A -> B (10)
C -> D (15)
D -> E (12)
E -> B (10)
F -> E (12)
```