

**Projekt AGB-10**

# **Entwurfsbeschreibung des Prototyps**

Verantwortlicher: Marcel Pfütze

Gruppe: SWP10-9

10. Mai 2010

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>3</b>
<b>2</b>	<b>Produktübersicht</b>	<b>3</b>
<b>3</b>	<b>Grundsätzliche Struktur und Entwurfsprinzipien für das Gesamtsystem</b>	<b>3</b>
3.1	Das Swing Interface . . . . .	3
<b>4</b>	<b>Grundsätzliche Struktur und Entwurfsprinzipien der einzelnen Pakete</b>	<b>4</b>
4.1	Die Klassen Edges und Nodes . . . . .	4
4.2	Die Klasse WeightedGraph . . . . .	4
4.3	Die Klasse TabbedReader . . . . .	5
4.4	Die Klasse GraphViewer . . . . .	5
4.5	Die Klasse NodeList . . . . .	5
4.6	Die Klasse MainWindow . . . . .	5

## 1 Allgemeines

Die Softwarestudie fordert die Entwicklung einer auf Prefuse basierenden Desktop-Anwendung zur visuellen Darstellung eines Graphen. Der Prototyp soll in der Lage sein einen Graph im Tab-separierten Format zu laden und daraus einen Graphen zu erstellen. Jede Zeile der INPUT-Datei enthält hierbei die Information über genau zwei adjazente Knoten und das dazugehörige Kantengewicht in der Form:

Knoten1 <tab>Knoten2 <tab>Kantengewicht

Zusätzlich zum Graphen wird eine Liste aller Knoten angezeigt. Wenn der Nutzer auf einen Knoten in der Liste oder im Graphen klickt, so setzt er den Fokus auf den ausgewählten Knoten und dieser verschiebt sich ins Zentrum des Graphen.

## 2 Produktübersicht

Der Prototyp bietet dem Nutzer zwei Möglichkeiten mit dem Programm zu interagieren. Der Nutzer kann zum einen über einen Button eine Datei hochladen, in der die Informationen über einen Graphen in der Form Knoten1 <tab>Knoten2 <tab>Kantengewicht gegeben ist. Auf Grundlage dieser Daten wird von der Anwendung in einem Panel ein Graph im radial view erzeugt. In einem Panel auf der linken Seite wird zusätzlich eine Liste aller Knoten erstellt. In dem erstellten Graphen hat der Nutzer die Möglichkeit einen Knoten zu fokussieren, indem er den entsprechenden Knoten entweder im Graphen oder in der lexikographisch geordneten Liste anklickt. Die Gewichte der Kanten werden als Label an den Kanten dargestellt.

## 3 Grundsätzliche Struktur und Entwurfsprinzipien für das Gesamtsystem

### 3.1 Das Swing Interface

Swing ist ein Interface und eine Grafikbibliothek von Sun Microsystems zum Programmieren von grafischen Oberflächen. Es ist seit Java Version 1.2 Bestandteil der Java-Runtime. Die Komponenten der mit Swing erstellten Oberflächen werden direkt von Java gerendert und sind somit nicht vom Betriebssystem abhängig. Wir benutzen Swing für die grafische Oberfläche unserer Anwendung.

## 4 Grundsätzliche Struktur und Entwurfsprinzipien der einzelnen Pakete

### 4.1 Die Klassen Edges und Nodes

Da die Prefuse Library keine Unterstützung für gewichtete Kanten bietet, gibt es eigene, neu implementierte Klassen, die diese Funktionalität anbieten. Diese nennen wir Edges und Nodes.

### 4.2 Die Klasse WeightedGraph

Objekte der Klasse WeightedGraph stellen einen gewichteten Graph dar. Der Konstruktor erzeugt dabei zunächst einen leeren WeightedGraph. Knoten und Kanten lassen sich über dafür vorgesehene Methoden hinzufügen. Über diverse GetterMethoden kann man auf die einzelnen Knoten und Kanten des Graphen zugreifen. Die Klasse bietet zudem eine Methode um den WeightedGraph in ein Objekt der Klasse Graph aus der Prefuse Library zu überführen. Hierfür wird eine interne private Klasse namens GraphParser benutzt.

Methoden zum Aufbau des Graphen:

**WeightedGraph()** Konstruktor der Klasse. Er erstellt lediglich leere Container für die Knoten und Kanten.

**Nodes addNode(Nodes node)** Methode, die ein übergebenes Objekt der Klasse Nodes zum Graphen hinzufügt. Wenn der übergebene Knoten bereits existiert, wird der Methodenaufruf ignoriert. Rückgabe ist der übergebene Knoten.

**Edges addEdge(Edges edge)** Das Kantenäquivalent zur addNode Methode. Auch hier wird der Aufruf ignoriert, falls die übergebene Kante bereits im Graphen existiert.

**Edges addEdge(Nodes n1, Nodes n2, int weight)** Man kann dem Graphen auch eine Kante hinzufügen, indem man die beteiligten Knoten und das dazugehörige Kantengewicht übergibt. Das Objekt der Klasse Edges wird dann intern erzeugt.

### 4.3 Die Klasse TabbedReader

Die Klasse TabbedReader bietet Methoden, mit denen sich gewichtete Graphen aus Textdateien in das Programm einlesen lassen. Der Graph muss dabei in folgendem Format vorliegen:

Knoten1 <tab>Knoten2<tab>Kantengewicht

Methoden zum Einlesen des Graphen:

**WeightedGraph readGraph(InputStream is)** Der Graph wird aus einem InputStream ausgelesen. Rückgabe ist ein Objekt der Klasse WeightedGraph.

**WeightedGraph readGraph(String path)** Der Pfad zu der Datei wird als Stringobjekt übergeben. Rückgabe ist wieder das den Graphen repräsentierende Objekt der Klasse WeightedGraph.

### 4.4 Die Klasse GraphViewer

Diese Klasse dient zur Darstellung des Graphen und ist von der Klasse Display aus dem Swing Interface abgeleitet. An den Konstruktor wird dabei ein WeightedGraph Objekt übergeben.

### 4.5 Die Klasse NodeList

Die Klasse NodeList ist eine von der Klasse JList abgeleitete Liste, in der man die einzelnen im Graph vorhandenen Knoten auswählen und anzeigen kann. Dem Konstruktor muss dabei eine LinkedList Datenstruktur übergeben werden, welche die einzelnen Objekte der Klasse Nodes enthält, die im Graphen vorkommen.

### 4.6 Die Klasse MainWindow

Diese Klasse stellt das Hauptfenster, in dem das Programm abläuft, dar. Dem Konstruktor werden zwei Strings übergeben. Der eine enthält den Titelnamen und der andere den Pfad der Textdatei, in welchem sich die Graphen befinden. Die Klasse ist von JFrame abgeleitet und basiert somit auch auf dem Swing Interface.