IMS
Institut für Mikroelektronische Systeme
Leibniz Universität Hannover

Programmierpraktikum Technische Informatik (C++)
Aufgabe 08

Hinweise
Abgabe: Stand des Git-Repositories am 28.6.2022 um 9 Uhr.
Die Dateien zur Bearbeitung dieser Aufgabe erhalten Sie, indem Sie die neue Aufgabe aus dem Aufgabenrepository in Ihr lokales mergen. Dies geschieht mit `git pull common main` innerhalb Ihres Repositories. Die Lösungen committen Sie bitte in Ihr lokales Repository und pushen sie in Ihr Repository auf den Gitlab-Server.

In dieser Aufgabe sollen Sie von einer gegebenen abstrakten Klasse (`Graph`) für gerichtete und gewichtete Graphen zwei neue Klassen (`AdjacentGraph`, `ClassicGraph`) ableiten und implementieren. Ein Graph besitzt Knoten (Nodes) und Kanten (Edges).

Teilaufgabe 1 (2.5 Punkte)

Erstellen Sie zunächst die Klasse `AdjacentGraph`, die von `Graph` abgeleitet werden soll. Die Klasse `AdjacentGraph` soll einen Graphen durch eine Adjazenzmatrix darstellen. Die `edge_id` entspricht der Position in der Matrix bei fortlaufender Nummerierung. Die Gewichte der Edges entsprechen den Einträgen in der Matrix. Eine `0` bedeutet, dass keine Edge vorhanden ist.

Hinweise:

- Die Dokumentation der zu implementierenden Memberfunktionen von `Graph` finden Sie in `graph.h`. Sie können sich auch mit `make doc` automatisch eine HTML-Dokumentation in `html/index.html` erzeugen.
- Eine Matrix lässt sich durch einen `std::vector<std::vector<T>>` darstellen.

Teilaufgabe 2 (2.5 Punkte)

Erstellen Sie nun die Klasse `ClassicGraph`. Diese speichert Nodes und Edges in eigenen Objekten. Hierzu sollen Sie für beide jeweils eine private Klasse innerhalb von `ClassicGraph` erzeugen. Die Klasse `ClassicGraph::Edge` speichert das Gewicht und die IDs der Nodes, mit welchen sie verbunden ist. Die Klasse `ClassicGraph::Node` speichert die IDs der ausgehenden und eingehenden Kanten in je einem `std::vector`. Die einzelnen Objekte

Appelstraße 4 | 30167 Hannover
fon 0511 762-19691 | fax 0511 762-19694
www.ims.uni-hannover.de | info@ims.uni-hannover.de

sollen im `ClassicGraph` in zwei Vektoren gespeichert werden.

Bonusaufgabe (2 Punkte)

Implementieren Sie Template-Versionen der Klassen, so dass in den Edges die Gewichte mit beliebigem Datentyp abgelegt werden können. Diese Klassen sollen `GraphT`, `AdjacentGrapT` und `ClassicGraphT` heißen.