

C23 – Fortgeschrittene Algorithmen und Programmierung

Klausur Wintersemester 2017/2018 – 1. PZR, Gruppe 1

Name _____

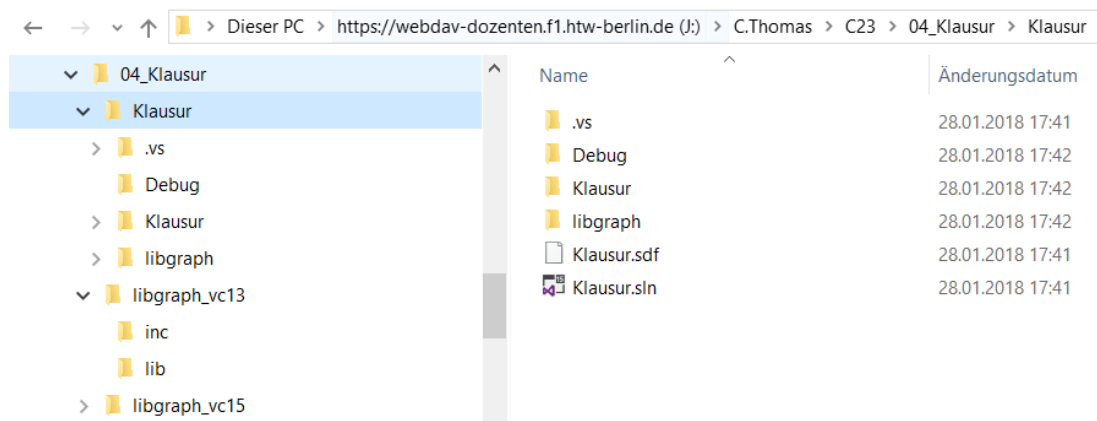
Matrikelnr. _____

Randbedingungen

- Die Klausuraufgabe ist durch jeden Klausurteilnehmer einzeln zu bearbeiten.
- Für die Klausur stehen 3 Zeitstunden zu Verfügung. Vorbereitungen (siehe unten) sind vor Beginn der Klausur abzuschließen.
- Die Klausuraufgabe ist auf den HTW-Rechnern im Laborraum zu bearbeiten. Das Ergebnis (Quelltexte, Projektdatei, UML-Diagramm und Testdokumentation) ist als .zip-Datei auf den Belegserver in das Verzeichnis „C.Thomas/C23/04_Klausur“ zu kopieren.

Vorbereitung

1. Kopieren Sie den Inhalt des Verzeichnisses „C.Thomas/C23/04_Klausur“ aus dem Dozentenserver in Ihr Arbeitsverzeichnis.
2. Starten Sie Visual Studio. Öffnen Sie das Projekt „Klausur“ im kopierten Verzeichnis.
3. Prüfen Sie, ob das Projekt „Klausur“ problemfrei übersetzt und gelinkt wird.
4. Sollte ein „_MSC_VER“ Linkerfehler auftreten, kopieren Sie je nach verfügbarer Visual Studio Version die .lib Datei aus dem Verzeichnis libgraph_vc15/lib oder libgraph_vc13/lib in das Verzeichnis Klausur/libgraph/lib.
5. Sollten weiterhin Fehler auftreten, kontaktieren Sie bitte den Dozenten.



Kontext – Schiffstransport

In Abbildung 1 ist das Netz der Häfen und Binnenwasserstraßen in Brandenburg (vereinfacht) abgebildet. Zwischen den Häfen verkehren Binnenschiffe, die unter anderem Schüttgüter (Sand, Kies, Schotter, Kohle) transportieren. Die Schiffe haben unterschiedliche Durchschnittsgeschwindigkeiten und Ladevermögen (siehe Tabelle 1).

Tabelle 1: Schiffstypen und Daten

Schiffstyp	Ladekapazität	Geschwindigkeit	Transportkosten
Klasse I	300 t	14,5 km/h	0,06 Euro / Tonnenkilometer
Klasse III	650 t	12 km/h	0,05 Euro / Tonnenkilometer
Klasse IV	1.350 t	10 km/h	0,03 Euro/ Tonnenkilometer

Auf einzelnen Strecken müssen Schleusen passiert werden. Das Passieren der Schleusen erfordert zusätzliche Fahrtzeit (siehe Abbildung 1) und kostet 10 Euro pro Durchfahrt, unabhängig vom Schiffstyp.

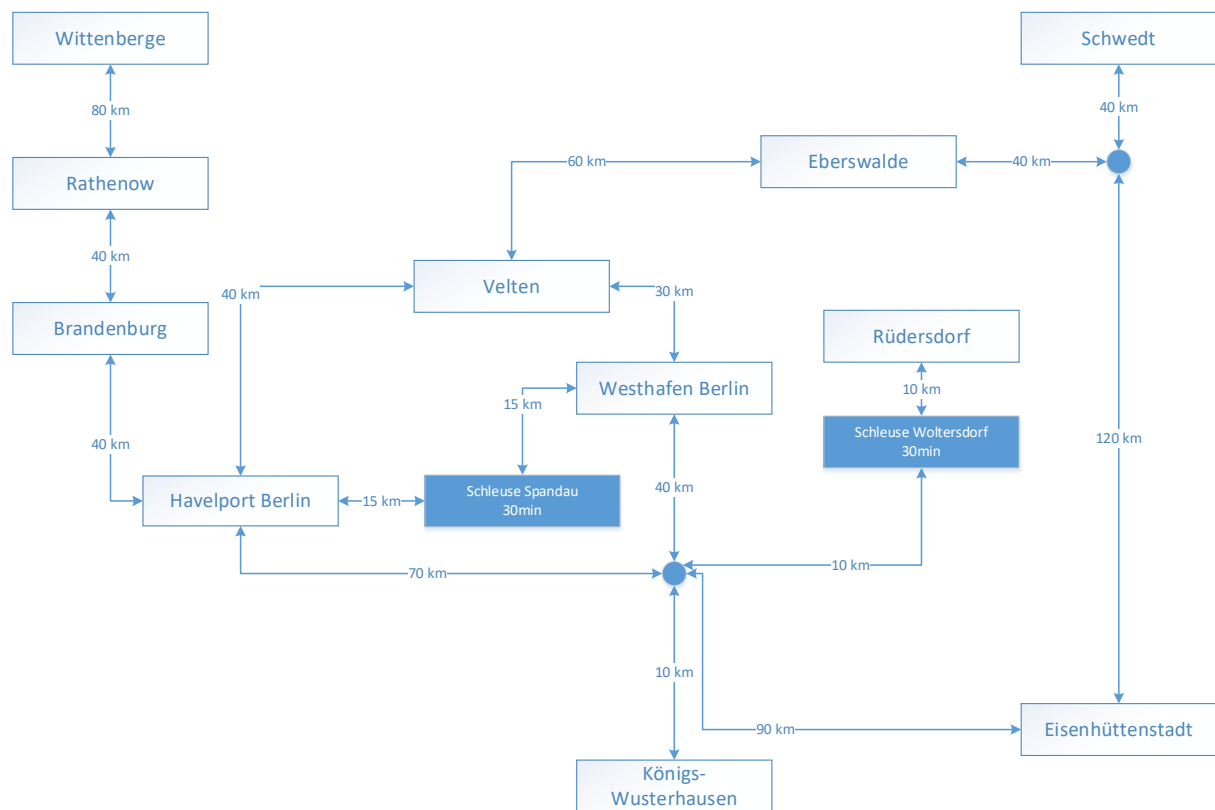


Abbildung 1: Netz der Binnenschiffahrtsstraße in Brandenburg, mit Häfen und Schleusen (vereinfacht)

Es gibt insgesamt 5 Schiffe, die in verschiedenen Häfen festgemacht haben (siehe Tabelle 2).

Tabelle 2: Verfügbare Schiffe

Schiffsname	Schiffstyp	Position
Berlin	Klasse III	Eisenhüttenstadt
Cottbus	Klasse III	Schwedt
Sachsen	Klasse IV	Velten
Donau	Klasse I	Westhafen Berlin
Elbe	Klasse I	Rüdersdorf

Anforderungen

Folgende Funktionalitäten sollen über ein Nutzermenü verfügbar sein:

- (1) Schiffahrtswegenetz ausgeben:
 - a. Liste der Häfen und Schleusen thematisch und alphabetisch sortiert ausgeben.
 - b. Zu jedem Hafen / jeder Schleuse die Verbindungen zu benachbarten Häfen / Schleusen ausgeben, mit Angabe der Streckenlänge
- (2) Transportauftrag erfassen:
 - a. Durch Benutzer Starthafen und Zielhafen auswählen lassen
 - b. Durch Benutzer Transportmenge angeben lassen (zwischen 100 Tonnen und 1.350 Tonnen)
- (3) Transport planen und ausführen:
 - a. Geeignetes Transportmittel auswählen
 - b. Zeitoptimale Strecke ermitteln und ausgeben (Achtung: Es kann sein, dass das Schiff erst leer zum Starthafen fahren muss, um anschließend beladen zum Zielhafen zu fahren)
 - c. Gesamtzeitbedarf ausgeben (inklusive Leerfahrt, sofern zutreffend)
 - d. Kosten für Transport (inklusive Leerfahrt, sofern zutreffend)
- (4) Gesamtliste der ausgeführten Transporte ausgeben:
 - a. Start- und Zielhafen, Kosten und Zeitbedarf sowie Namen und Klasse des ausführenden Schiffes ausgeben
- (5) Positionen der Schiffe ausgeben
 - a. Name des Schiffes, sowie Klasse und Position (Name des Hafens) ausgeben
- (6) Programm beenden.

Aufgabenstellung

- Implementieren Sie eine C++ Anwendung, welche die beschriebene Anwendung als Konsolenanwendung umsetzt. Implementieren Sie das Netzwerk der Häfen / Schleusen und Schifffahrtswege nach Abbildung 1 als Graph und erfüllen sie die oben genannten Anforderungen.

Nutzen Sie dabei die Graph-Library und das Projekt „Klausur“:
„<Dozentenserver>/C.Thomas/C23 /04_Klausur“.

- Verändern Sie die Klassen in der libgraph-Library nicht, sondern arbeiten Sie mit abgeleiteten Klassen, falls nötig!

- Erstellen Sie ein Klassendiagramm (UML) für Ihre Anwendung, bevor Sie mit der Implementierung beginnen. Es soll die benutzten Klassen mit den wichtigsten Membervariablen / Funktionen, Assoziationen und den Vererbungsbeziehungen darstellen.
- Testen Sie die Anwendung, indem Sie die Anwendung starten und:
 1. Das Schifffahrtswegenetz ausgeben
 2. Die Positionen der Schiffe ausgeben
 3. Die Gesamtliste der Transportaufträge ausgeben
 4. Die Transportaufträge nach Tabelle 3 ausführen
 5. Die Positionen der Schiffe ausgeben
 6. Die Gesamtliste der Transportaufträge ausgeben
 7. Das Programm beenden

Tabelle 3: Transportaufträge

Starthafen	Zielhafen	Ladung
Velten	Westhafen Berlin	1200 Tonnen
Westhafen Berlin	Eisenhüttenstadt	250 Tonnen
Wittenberge	Rüdersdorf	500 Tonnen

Kopieren Sie zur Dokumentation der Testergebnisse die Ein- und Ausgaben ihres Programms von der Konsole in eine Textdatei und geben Sie diese mit ab.

Viel Erfolg!

Bewertung

	nicht	teilweise	vollständig
Entwurf			
Klassenentwurf / Vererbung etc. richtig definiert	0	5	10
Daten / Funktionen richtig definiert	0	5	10
Implementierung			
Anforderungen vollständig umgesetzt	0	5...25	30
Kontrollfluss fehlerfrei	0	5...10	15
Stil & Dokumentation			
Coding-Richtlinien eingehalten	0	0	5
Programm lesbar und kommentiert	0	0	5
Test			
Programm funktionsfähig wie gefordert	0	5...15	20
Testergebnisse dokumentiert	0	0	5