

Logistik Management System - LogManSys

Situationbeschreibung:

Sie organisieren in Ihrem Logistikunternehmen den Frachtverkehr zwischen den Ballungsgebieten in Deutschland. Der Verkehr wird mit LKWs im Linienbetrieb durchgeführt. Das bedeutet, dass alle angebotenen Verbindungen täglich befahren werden, unabhängig von deren Auslastung.

Folgende Abkürzungen werden verwendet:

Hamburg (HH)
Hannover (H)
Frankfurt (F)
Berlin (B)
Stuttgart (S)
Leipzig (L)
München (M)

Jede in einer Liste eingetragene Route wird von genau einem LKW befahren. Diese Route kann aus mehreren Teilstrecken bestehen.

Allgemeine Struktur:

Startpunkt – Zwischenpunkt₁ Zwischenpunkt_i Zwischenpunkt_n – Zielpunkt

Listenbeispiel:

F, S, M
HH, H, B, L, M, S, F
HH, H, F, H, HH
M, L, B, HH
S, M
S, M
H, F, S
S, F, H
HH, H, B, L, M, S, F, L, B
HH, B, L, M

Ein letzte Beispieleintrag **HH, B, L, M** bedeutet also eine Route von Hamburg nach München mit erstem Zwischenhalt in Berlin und zweitem Zwischenhalt in Leipzig.

Jede Zeile der Liste stellt eine Route dar. Beachten Sie, dass die Länge der Zeilen unterschiedlich sein können. Keine Zeile wird jedoch mehr als 20 Einträge (also Start-, Ziel- und max. 18 Zwischeneinträge) enthalten

Wichtige Werte für jede Route sind das noch zur Verfügung stehende Frachtgewicht (freies Frachtgewicht) und das zu Verfügung stehende Frachtvolumen (freies Frachtvolumen) beschrieben. Die Werte für das freie Frachtvolumen und das freie Frachtgewicht sollen für die gesamte Route als konstant angesehen werden und sich nicht von Teilstrecke zu Teilstrecke ändern.

Beispiel: Ein LKW mit dem maximalen Zuladungsgewicht von 25 Tonnen und dem maximalen Ladevolumen von 36 m³ befährt eine Route.

	Unbeladen	mit Ladung 4,5t; 26m ³
-----	-----	-----
freies Frachtgewicht:	25t	20,5t
freies Frachtvolumen:	36m ³	10m ³

Aufgabe:

Für die erste SW-Version soll eine Klasse **Route** konzipiert werden, die die einzelnen Routen verwaltet. Das bedeutet, die Gesamtroutenliste ist ein Array bzw. eine Liste bestehend aus Objekten der Klasse **Route**.

a. Erstellen Sie diese Klasse in JAVA mit den Attributen zum Streckenverlauf und den Frachtkapazitäten. Versehen Sie diese Klasse mit einem Konstruktor mit dem sich die Attribute initialisieren lassen.

b. Erweitern Sie die Klasse um folgende getter-Methoden:

```
Point: getStart()  
Point: getZiel()  
double: getAvailableVolume()  
double: getAvailableWeight()
```

c. Erweitern Sie die Klasse um folgende Methoden oder erstellen Sie entsprechende Functions.

```
// wird der entsprechende Punkt auf dieser Route angefahren?  
boolean: isConnected ( Point )
```

```
// exist. für die beiden Punkte eine direkte Verbindung?  
boolean: existAsDirectLink ( Point1; Point2 )
```

```
// exist. für die beiden Punkte eine direkte oder indirekte Verbindung?  
boolean: existAsLink      ( Point1; Point2 )
```

d. Ein Kunde fragt an, ob er Frachtkapazität (Volumen und Gewicht) vom Startpunkt A zum Zielpunkt B buchen kann. Erstellen Sie eine Methode bzw. Funktion die die Liste durchsucht und entscheidet, ob sich diese Anfrage positiv beantworten lässt.

Erstellen Sie eine weitere Methode / Funktion, die die Buchung eines Transportes vom Startpunkt A zum Zielpunkt B mit dem gebuchten Volumen und Gewicht in der Routenliste berücksichtigt.

Zusatzanforderung:

Es gilt als geschickt, diese Buchung in der Route mit den wenigsten Teilstrecken zu berücksichtigen.

Hinweis: Die Notation kann als JAVA-Quelltext, Struktogramm oder als UML-Aktivitätsdiagramm erfolgen.

e. Ihr Logistikunternehmen plant den Aufbau einer zusätzlichen Umlademöglichkeit an einer der angefahrenen Stationen. Es ist sinnvoll die Umlademöglichkeit an einer solchen Station zu errichten, die von den meisten Routen/LKWs angefahren wird.

Erstellen Sie eine Methode / Funktion, die herausfindet welche Station von den LKW am häufigsten angefahren wird.

Hinweis: Die Notation kann als JAVA-Quelltext, Struktogramm oder als UML-Aktivitätsdiagramm erfolgen.