

Bachelor-Studiengang Informatik

Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2009/2010

Übungsblatt 9: diverse Themen

Ausgabe am: 16.12.2010
freiwillige Abgabe am: 10.1.2011

Zusatzaufgabe 1

8 + 7 + 3 + 6 + 6 = 30 Punkte

Das S-Bahn-Netz der Stadt Byteburg ist regelmäßig aufgebaut:

- Es gibt fünf Hauptlinien ("1" bis "5") und eine Ringlinie.
- Die Stationen entlang der Hauptlinien sind von innen nach außen mit 1 bis 6 durchnummeriert. Im Zentrum liegt Station "00".
- Die Ringlinie verbindet reihum die Stationen 3 der Hauptlinien.
- Die Stationen sind mit einem zweistelligen Code benannt: Die Zehnerstelle nennt die Linie, die Einerstelle die Stationsnummer. In der Skizze unten ist das für zwei Linien eingetragen.
- Linie "6" befindet sich (inklusive Station 63) im Bau und wird noch nicht benutzt.

Im Innenraum (in der Skizze blau hinterlegt) liegen das Zentrum und alle Stationen bis einschließlich der Ringlinie. Die übrigen Stationen liegen im Außenraum. Schreiben Sie ein Programm "ByteburgTarif", das die Fahrpreise nach den folgenden Regeln berechnet:

- Eine Fahrt kostet 2 Euro (Beispiel: 11 nach 13: 2 Euro).
- Jede überquerte Zonengrenze kostet zusätzlich 1 Euro (Beispiel: 11 nach 14: 3 Euro).
- Jede benutzte Endstation kostet zusätzlich 1 Euro (Beispiel: 16 nach 11: 4 Euro).
- Eine Fahrt zwischen zwei benachbarten Stationen kostet immer 1 Euro, auch wenn eine Zonengrenze dazwischen liegt oder eine Endstation benutzt wird (Beispiele: 13 nach 14, 15 nach 16 und 13 nach 53: jeweils 1 Euro).

Auch nach Bau von Linie "6" bleiben die Hauptlinien im Uhrzeigersinn fortlaufend durchnummeriert. Ihr Programm soll mit möglichst wenig Aufwand für das erweiterte Streckennetz anpassbar sein.

Aufgabenstellung:

- Erstellen Sie eine Klasse *ByteburgTarifRechner*, welche die Fahrpreise nach den oben angegebenen Regeln für fünf Linien berechnet. Implementieren Sie in dieser Klasse eine Methode

```
int ermittleFahrkosten( int startStation, int zielStation )
```

welche die Fahrkosten von Station *startStation* nach Station *zielStation* in Euro berechnet.
- Erstellen Sie in einer weiteren Klasse *ByteburgTarif* eine *main*-Methode, welche die Codes von zwei Stationen als ganze Zahlen von der Konsole liest und die Fahrkosten ausgibt. Die eingegebene Stationsnummer soll auf Gültigkeit geprüft werden.
- Überlegen Sie genau, was Sie an Ihrem Programm ändern müssen, um die Tarife auch für die neue Linie berechnen zu können.

d) Erweitern Sie Ihr Programm so, dass Sie dem Konstruktor der Klasse *ByteburgTarifRechner* die Anzahl der Linien von Byteburg als Parameter übergeben können. Erstellen Sie an jeder Stelle, die Sie gegenüber Teilaufgabe a ändern, einen kurzen Kommentar.

e) Diskutieren Sie:

- Wäre in Ihrer Lösung ein Interface sinnvoll einzusetzen (Sie dürften Ihre Lösung dazu abwandeln)? Wenn ja, welche Klasse(n) würde(n) dieses Interface benutzen, welche würde(n) es implementieren und welche Methoden wären enthalten?
- Wäre in Ihrer Lösung Vererbung sinnvoll einzusetzen? Wenn ja, welche Klassen würde es geben, was würde von welcher Klasse geerbt und erweitert werden?

Von Ihren Lösungen zu den Teilaufgaben a und b sind Listings abzugeben, *keine* Testläufe. Wenn Sie Teilaufgabe d bearbeiten, geben Sie nur deren Listings ab, nicht diejenigen von a und b.

Wenn Sie eine Lösung für dieses freiwillige Übungsblatt abgeben möchten, dann senden Sie Ihre Lösung bis spätestens zum 10. Januar 2011 um 10 Uhr an Ihren Tutor für dieses Übungsblatt. Erfragen Sie zuvor *in einer Übungsstunde*, wer sie betreuen wird.

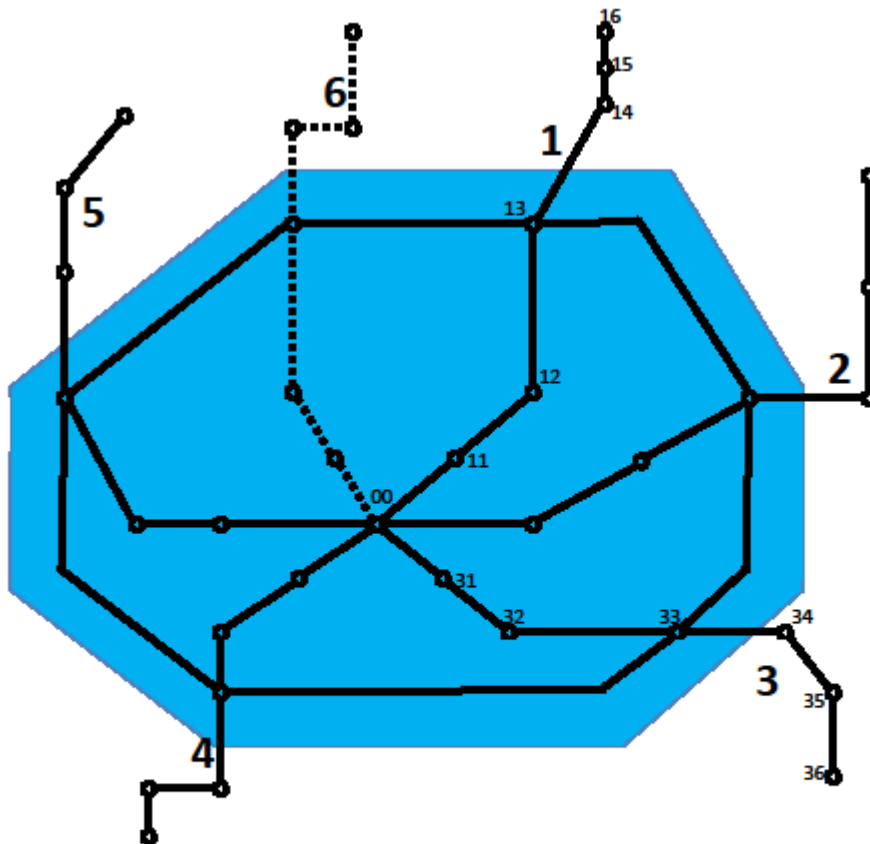
Hinweis zum Zusatzblatt

Die Aufgaben sind in Eclipse zu bearbeiten.

Legen Sie für die Bearbeitung dieses Übungsblattes ein Paket (engl. Package) namens *uebung09* an, in dem Sie alle Klassen ablegen.

Vergessen Sie nicht die Kommentare...!

Skizze zur Aufgabe:



Vergrößerte Darstellung:

Der blaue Bereich zeigt die Innenzone bzw. den Verlauf der Grenze von der Innen- zur Außenzone, die Ringlinie als Verbindung der jeweiligen Station 3 hat nichts mit dem Zonenverlauf zu tun.