

# Rechnernetze und verteilte Systeme

## Übungsblatt 10

**Ausgabe:** 17. Dezember, **Abgabe:** 10. Januar 2014, 20:00 Uhr

### Programmieraufgabe 10.1 (20 Punkte)

Implementieren Sie in Java 7 eine einfache Verwaltung einer Routing-Tabelle. Auf der Übungsseite finden Sie entsprechende Vorlagen (Flags.java, IRouter.java, Main.java, Router.java, HWPort.java).

Die Anwendung soll folgende Eigenschaften erfüllen:

- Halten Sie sich an die Spezifikationen, wie sie in IRouter.java vorgegeben sind. Halten Sie sich insbesondere an die weiteren Vorgaben, die Sie in den Kommentaren finden.
- Sie müssen lediglich die Klasse Router.java implementieren. Weitere Dateien sind nicht abzugeben.
- Die Klasse Router soll eine Routing-Tabelle verwalten. Neue Routen können hinzugefügt und alte gelöscht werden. Es soll die passende Route gefunden werden und die Routingtabelle ausgedruckt werden können.
- Sie können davon ausgehen, dass eine Destination-IP-Adresse nur einmal eingetragen werden kann. Sollte versucht werden, eine bereits vorhandene Destination-IP-Adresse erneut hinzuzufügen, muss `routeAdd(...)` `false` zurückgeben und eine Fehlermeldung ausgegeben. Einträge in der Routing-Tabelle können somit eindeutig an der Destination-IP-Adresse identifiziert werden.

Hinweise:

- Verwenden Sie bitte keine Umlaute und dergleichen in Ihren Kommentaren (Kommentare in Englisch sind erlaubt)! Achten Sie bitte auf sinnvolle Bezeichner und lesbare Formatierung!
- Beginnen Sie Ihre Arbeit am besten damit javadoc zu benutzen. Es empfiehlt sich, dann und wann in die Java API zu schauen.
- Die Methode `System.out.format` bietet die Möglichkeit, Ausgaben komfortabel zu formatieren. Sie können dies nutzen um z.B. Routingtabellen auszugeben.
- In Java ist ein Integer genau 32 Bit lang. Dies nutzen wir zur Speicherung von IP-Adressen. Dass Integer Dezimalzahlen in Zweierkomplementdarstellung repräsentieren, ist für uns nicht von Bedeutung, da alle benötigten Operationen bitweise sind. Welche Dezimalzahl repräsentiert wird, ist also zunächst unwichtig.
- Sie werden bitweise Operatoren benötigen. Informieren Sie sich über diese.
- Falls Sie komplexere Datenstrukturen zur Verwaltung ihrer Tabelle verwenden möchten, können Sie sich mit inneren Klassen / nested classes behelfen. Verwenden Sie bitte eine angemessene, dynamische Datenstruktur, um Ihre Einträge zu speichern (LinkedList, ArrayList, Maps etc. ). Eine Möglichkeit wäre, eine innere Klasse zu schreiben, die jedes Attribut einer Route als Klassenvariable speichert. Objekte dieser Klasse könnten Sie dann zum Beispiel in einer ArrayList speichern.
- Fehlermeldungen müssen auf der Standardfehlerausgabe, dem Error Stream, ausgegeben werden.

- Abgaben, die nicht kompilierbar sind, werden mit 0 Punkten bewertet.
- Achten Sie bitte darauf, dass Sie Ihre Abgaben *pünktlich* hochladen. Sehen Sie für eventuelle Probleme ein großzügiges Zeitfenster vor dem Termin vor! Nicht pünktlich abgegebene Arbeiten werden mit 0 Punkten bewertet. Nachträgliche Korrekturen nach dem Abgabetermin sind nicht möglich. Bewertet wird dann das, was Sie zuletzt hochgeladen haben.

### **Achtung!**

**Bitte nur selbst erstellte Lösungen abgeben. Mehrfache Abgaben derselben Lösung (Plagiate) werden mit 0 Punkten gewertet und führen zum Nichtbestehen der Studienleistung!**

Die Abgabe der Aufgabe erfolgt über ASSESS unter <http://ess.cs.uni-dortmund.de/ASSESS/>.

**Beachten Sie bitte auch Einträge im Forum:**

<http://ls4-www.cs.tu-dortmund.de/phpbb3/viewforum.php?f=11>

**Antworten der Moderatoren in dieser Gruppe sind verbindlich.**

**Die Abgabe muss in 2er bis 4er Gruppen erfolgen.**