

Die Hausaufgabe kann in Gruppen von bis zu vier Studierenden abgegeben werden. Bitte pro Gruppe nur eine PDF-Datei und eine Java-Datei hochladen! Bitte laden Sie alle Dateien mit demselben Benutzerkonto hoch und vermerken in jeder (!) Datei Namen und Matrikelnummern aller Gruppenmitglieder!

Aufgabe 1.1 – DNS und HTTP

Sie wollen von Ihrem Rechner im Hochschulnetz aus mittels HTTP ein Objekt vom Server *www.google.com* abrufen. Berechnen Sie die Zeit, die vom Starten der Anfrage in Ihrem Browser bis zur Anzeige des Objektes vergeht unter den folgenden Annahmen:

- Die Zeit für die Erzeugung der Anfrage und das Darstellen des Objektes auf Ihrem Rechner kann vernachlässigt werden.
- Die Übertragungszeit für das angefragte Objekt beträgt 201ms.
- Die IP-Adresse von *www.google.com* ist in keinem DNS-Server zwischengespeichert.
- Die Auflösung des Hostnamens *www.google.com* erfolgt mit Hilfe des lokalen DNS-Servers der Hochschule über eine iterative DNS-Anfrage.
- Die Verarbeitungszeit für die DNS-Anfrage auf dem lokalen DNS-Server der Hochschule und allen weiteren DNS-Servern kann vernachlässigt werden.
- Die Round-Trip-Zeiten zwischen den beteiligten Rechnern sind in der folgenden Tabelle dargestellt:

	Ihr Rechner	Lokaler DNS-Server	Root-DNS-Server	TDL-DNS-Server COM	Autoritativer DNS-Server google.com	www.google.com
Ihr Rechner	X	411	32	255	88	11
Lokaler DNS-Server	411	X	40	270	90	13
Root-DNS-Server	32	40	X	21	66	9
TDL-DNS-Server COM	255	270	21	X	16	14
Autoritativer DNS-Server google.com	88	90	66	16	X	12
www.google.com	11	13	9	14	12	X

Aufgabe 1.2 – HTTP und Socket-Programmierung

Entwickeln Sie einen einfachen HTTP-Client in Java, ausgehend vom in der Vorlesung vorgestellten *EchoClient*. Dieser Client soll sich mit dem Befehl

```
java HTTPClient
```

starten lassen.

Wenn der Client gestartet wird, soll er nacheinander die folgenden Aktionen durchführen:

1. Eine URL wird vom Benutzer erfragt.
2. Bei leerer URL soll das Programm beendet werden, sonst weiter bei Schritt 3.
3. Ein TCP-Socket zum in der URL angegebenen Server wird geöffnet.
4. Eine HTTP-Anfrage für das in der URL angegebene Objekt wird über das Socket gesendet.
5. Die Antwort des Servers wird empfangen und auf dem Bildschirm ausgegeben.
6. Das Socket wird geschlossen.
7. Der Ablauf wird bei 1 fortgesetzt.

Ergibt sich z.B. durch falsche Server-Namen, Netzfehler, ... ein Ablauf, der vom im Skript beschriebenen „normalen“ Ablauf abweicht, soll eine Fehlermeldung ausgegeben und der Vorgang bei Schritt 1 fortgesetzt werden.

Geben Sie die Datei *HTTPClient.java* in Ilias ab.