

Da in einem bestimmten Netzwerksegment (194.94.2.xx) Probleme mit stabilen Verbindungen und Übertragungsraten bestehen, soll mit einem Programm untersucht werden, unter welchen IP-Adressen in diesem Netzbereich Rechner aktiv sind. Hierzu wird jede IP-Adresse in diesem Bereich (194.94.2.1 - 194.94.2.128) mit dem ping-Befehl (weiter unten gibt es hierzu Beispielausgaben) angesprochen und eine Liste aller Adressen erstellt, von denen eine Antwort kommt.

Um das reale Netzwerk nicht unnötig zu belasten, soll für die ersten Schritte eine Simulation verwendet werden. Hierfür kann die Klasse Net verwendet werden, die auch eine Simulation des ping-Befehls liefert - Adressen werden hierbei als String übergeben.

Die Klasse Net wird uns zum Glück fertig übergeben und auch von anderen Teams gepflegt – da brauchen (und dürfen) wir also auch nichts dran verändern. Unser Programm erstellen wir daher in einer neuen, eigenen Klasse.

Das eigentliche Programm soll wieder in kleinen, aufeinander aufbauenden Schritten entstehen:

- Schreiben Sie eine Applikation, die zu Testzwecken den Befehl ping(...) aus der Klasse Net aufruft, und das Ergebnis auf dem Bildschirm ausgibt. Unter den Adressen 194.94.2.14 und 194.94.2.20 sollten unterschiedliche Ergebnisse zu sehen sein.
Erweitern Sie ihre Applikation, in dem Sie die Antwort vor der Ausgabe in einer Variablen zwischenspeichern.
- (Bonus: Nur für die Nerds und nur, wenn noch Zeit bleibt) Binden Sie die Klasse Net zur Verwendung des ping-Kommandos als statischen Import ein.
- Erstellen Sie eine Methode, die überprüft, ob in einem übergebenen String das Wort „Zeitüberschreitung“ vorkommt.
Genauso wichtig ist es, zu erkennen, ob in diesem String eine Antwortzeit vorkommt. (welche Buchstabenkombination ist hier ein sicheres Erkennungszeichen?)
Als Antworttyp für diese Methoden bietet sich ein boolean an – wer eine andere Idee verfolgen möchte, darf aber auch gerne kreativ werden.
- Überprüfen Sie mit einer eigenen Methode, ob eine Adresse auf einen ping-Befehl antwortet, also erreichbar ist. Die Adresse soll dazu als Parameter übergeben werden.
- Nun soll mit einer eigenen Methode in dem gesamten Netzsegment (194.94.2.1 - 194.94.2.128) gezählt werden, wie viele Adressen nicht antworten, also noch vergeben werden können. Die Anzahl soll in der main-Methode ausgegeben werden.
- Die antwortenden Adressen sollen jetzt in einer geeigneten Datenstruktur gespeichert werden. Das könnte ein Array oder auch eine geeignete Collection sein – das entscheiden Sie selber nach Vorliebe und Fähigkeit.
- Für neue Rechner, die noch in das Netzwerk eingebunden werden können, sollen die wichtigsten Informationen (wem sie zugeteilt wird, die Adresse, was für ein Gerät angeschlossen wird, also Computer, Netzwerkdrucker, ...) in Objekten abgespeichert werden. Erstellen Sie hierfür eine geeignete Klasse mit den entsprechenden get/set-Methoden, geeignetem Konstruktor und einer toString-Methode

- (Bonus: Nur für die Nerds und nur, wenn noch Zeit bleibt)

Jetzt ist es eigentlich an der Zeit, ins reale Netz zu gehen: Mit der Kommandosequenz

```
Runtime.getRuntime().exec( <Befehl> );
```

ist man unter JAVA in der Lage, externe Kommandos und Programme auszuführen.

Verändern Sie Ihr Programm so, dass es anstelle des simulierten ping-Befehls nun mit dem realen ping in der Konsole arbeiten kann.

Beispiel für eine ping-Antwort für eine nicht erreichbare Adresse:

Ping wird ausgeführt für 192.168.10.100 mit 32 Bytes Daten:

Zeitüberschreitung der Anforderung.

Zeitüberschreitung der Anforderung.

Zeitüberschreitung der Anforderung.

Zeitüberschreitung der Anforderung.

Ping-Statistik für 192.168.10.100:

Pakete: Gesendet = 4, Empfangen = 0, Verloren = 4
(100% Verlust),

Beispiel für eine erfolgreich ausgeführte Abfrage:

Ping wird ausgeführt für 194.94.2.1 mit 32 Bytes Daten:

Antwort von 194.94.2.1: Bytes=32 Zeit<1ms TTL=255

Antwort von 194.94.2.1: Bytes=32 Zeit<1ms TTL=255

Antwort von 194.94.2.1: Bytes=32 Zeit<1ms TTL=255

Antwort von 194.94.2.1: Bytes=32 Zeit<1ms TTL=255

Ping-Statistik für 194.94.2.1:

Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
(0% Verlust),

Ca. Zeitangaben in Millisek.:

Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms