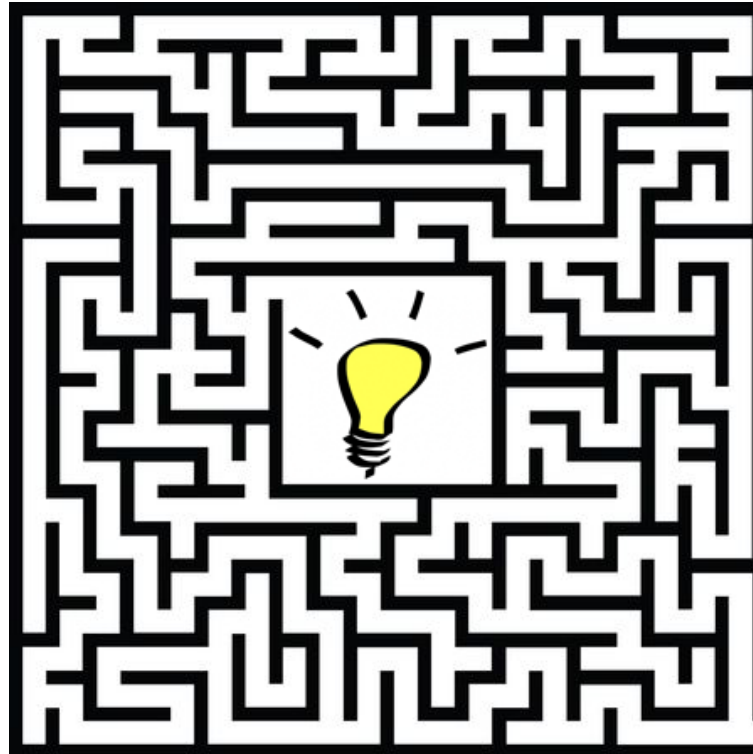


Prof. Dr. H. Pagnia

Advanced IT
(Testate Aufgaben)



Herbst 2021

DHBW Mannheim

Testat Aufgabe 2**Testat Aufgabe 2** (max. 5 Punkte)

Entwerfen Sie mit TCP einen Server, der Nachrichten speichert und zur Abfrage über das Netz bereit hält.¹
Zum Ablegen einer Nachricht auf dem Server sendet ein Client einen String mit dem folgenden Format an den Server:

SAVE *beliebig langer String mit abschließendem Zeilenende*

Der Server generiert nach dem Empfang einen neuen geeigneten eindeutigen Schlüssel (als String) und speichert die Nachricht in einer Datei, wobei der Schlüssel als Dateiname verwendet wird. Danach sendet der Server den Schlüssel zurück an den Client:

KEY *schluessel*

Alle Dateien sollen auf dem Server auf dem Desktop im Verzeichnis "Messages/" abgespeichert werden, das Sie vorher schon anlegen sollten.

¹Einen ähnlichen Dienst bieten pastebin.com bzw. ZeroBin (<https://sebsauvage.net/wiki/doku.php?id=php%3Azerobin>)

Zum Abrufen einer Nachricht sendet ein Client einen String:

GET *schluessel*

an den Server, der daraufhin überprüft, ob eine entsprechende Datei existiert.

Falls ja, sendet er den Inhalt der Datei an den Client:

OK *dateiinhalt*

Anderenfalls sendet er:

FAILED

Implementieren Sie den Server auf Port 7777 sowie einen Client zum Testen.

Tipps

- ◇ Implementieren Sie den Server als *Non-Persistent Server* und passen Sie den Client entsprechend an.
- ◇ Verwenden Sie die Filter-Streams `PrintWriter` und `BufferedReader`.
- ◇ Verwenden Sie *split* aus der Klasse `String` zum Zerlegen der Nachrichten.
- ◇ Verwenden Sie die Klassen `FileReader` und `FileWriter` zum Zugriff auf die Dateien:
 - ▷ `new BufferedReader(new FileReader(filename))`
 - ▷ `new PrintWriter(new FileWriter(filename))`
- ◇ Erzeugen Sie den eindeutigen Schlüssel mithilfe eines Zufallszahlengenerators.

Die Bearbeitung erfolgt einzeln oder in Zweiergruppen. Für Client und Server sind geeignete Testfälle zu dokumentieren. Einzureichen ist bis **spätestens zum So. 24.10.2021 23:59 Uhr** ein ZIP-File mit

- (1) einem `src` Verzeichnis mit dem kompilierbaren (und kommentierten!) Java-Code,
- (2) einem `class` Verzeichnis mit dem lauffähigen Code,
- (3) einem `test` Verzeichnis mit der Testdokumentation,
- (4) einer von der/dem/den Bearbeitenden unterschriebenen Ehrenwörtliche Erklärung mit der Zusage, dass das Programm eigenständig entwickelt und implementiert wurde.