

Übungsblatt 17

Aufgabe 1: (Streams, Sockets)

- a) Schreiben Sie einen TCP Server, der wiederholt Textzeilen von einem Client empfangen kann. Der Server sendet dem Client nach jeder empfangenen Zeile ein OK zurück.
- b) Schreiben Sie einen TCP Client, der wiederholt von der Standardeingabe Textzeilen liest und diese an den Server aus Aufgabenteil a) sendet.
- c) Verwenden Sie das Gerüst aus den Aufgabenteilen a) und b) um das Spiel Galgenmännchen über das Netzwerk umzusetzen. Der Server soll sich den Begriff ausdenken. Der Client soll wiederholt versuchen, den Begriff zu erraten. Die Regeln sind aus Programmieren I bekannt!

Aufgabe 2: (Streams, Object Serialization, Sockets)

- a) Ändern Sie den TCP Server aus Aufgabe 1a) so, dass er wiederholt ein Objekt der Klasse Person (mit Vorname, Nachname und Telefonnummer) entgegennehmen kann. Jedes entgegengenommene Objekt soll er fortlaufend hintereinander in die gleiche Datei schreiben.
- b) Ändern Sie den TCP Client aus Aufgabe 1b) dahingehend, dass er vom Nutzer wiederholt Vorname, Nachname und Telefonnummer einer Person einliest, daraus ein Personenobjekt erstellt und dieses Personenobjekt an den TCP Server sendet.

Aufgabe 3: (Generics, Stack)

Auf Übungsblatt 13 Aufgabe 3 haben sie mit Hilfe einer Liste einen Stack implementiert. Löschen Sie die Annotation `@SuppressWarnings` aus dem Quellcode und verwenden Sie Generics so, dass Sie keine Warnungen mehr bekommen. Bei der Deklaration einer Referenz auf den Stack sowie bei der Instanziierung soll es möglich sein, eine Typangabe mitzugeben, bspw. (wenn ihre Klasse `GenericStack` heißt):

```
GenericStack<String> s = new GenericStack<>();
```

Folgende main-Methode müsste bei Ausführung mit Ihrer Klasse, „Welt“ und „Hallo“ ausgeben:

```
public static void main(String[] args) {  
    GenericStack<String> s = new GenericStack<>();  
    s.push("Hallo");  
    s.push("Welt");  
    System.out.println(s.pop());  
    System.out.println(s.pop());  
}
```

Aufgabe 4: (JUnit Test)

- a) Formulieren Sie die vorgegebene main-Methode aus Aufgabe 3 in einen JUnit Test um!
- b) Testen Sie ferner das Verhalten der Methode `pop()` auf einen leeren Stack. Wenn ein Fehler auftritt, was wäre eine gute Abhilfe? Setzen Sie die Maßnahme um!