

Übung zu Nachrichtenformate

Aufgabe 1

Betrachten sie das Zug-Beispiel aus der Übung 2.

Wir nehmen an, dass das Programm auf mehrere Rechner verteilt wird:

- Auf einem Rechner (Server) ist die ‚Strecke‘ implementiert.
- Auf jedem Zug ist auf einem Rechner die Klasse Zugfahrt implementiert.
- Auf einem weiteren Rechner ist die Klasse ‚Streckenwächter‘ implementiert.

Untersuchen Sie das Beispiel dahingehend, welche Informationen in diesem System dann über Nachrichten ausgetauscht werden müssen.

Aufgabe 2

Entwerfen Sie ein Nachrichtenformat, das zum Austausch der Informationen geeignet ist.

Berücksichtigen Sie in ihrem Beispiel die zusätzlichen Informationen:

Datum/Uhrzeit	Datum/Uhrzeit, zu der eine Nachricht gesendet werde
Rechnerkennung	Sender und Empfänger einer Nachricht (z.B. die Zugnummer)
Sequenznummer	fortlaufende Nummer, die je Rechner vergeben wird

Bedenken Sie, dass nicht nur Anfragen gestellt werden müssen, sondern auch Antworten zu übertagen sind sowie mögliche Fehler (z.B. Ausnahmen).

Aufgabe 3

Implementieren Sie die Methoden zur Erzeugung ihrer Nachrichten. Binden sie dies so in ihr Programm ein, dass anstelle einer `send()/receive()`-Methode nur die zusätzliche Protokollierung der Kommunikation in ihrem Nachrichtenformat erfolgt (z.B. über `System.out.println()`-Befehle).

Aufgabe 4

Wir beantworten die nachstehenden Fragen gemeinsam:

- Welche Eigenschaften hat das von Ihnen gewählte Datenformat?
(wohldefiniert, vollständig, maschinell analysierbar (Parsable), erweiterbar, dokumentiert, effizient, robust)
- Welche Vereinbarungen müssen getroffen werden, damit ihr Schnittstellenformat von allen Anwendungen gleichermaßen verarbeitet werden kann?