

Erwartete Lösung	I	II	III
Quellen: Hartmut Meyer			
Aufgabe 1 (30%): a) Eine Aufteilung der Datensätze in kleinere – jeweils über eine Prüfsumme gesicherte – Pakete, würde den Schaden bei einem Übertragungsfehler minimieren: Nicht alle, sondern nur ein Messwert geht verloren. Demgegenüber steht ein größerer Kommunikationsoverhead (ungünstigeres Verhältnis von Nutzdaten zu Gesamtdaten), was bei der stark begrenzten Bandbreite ein Problem ist. Je höher die Fehlerrate bei der Übertragung ist, desto sinnvoller würde eine Aufteilung der Datenpakete. b) TCP würde die Korrektheit der übertragenen Daten garantieren. Wegen der begrenzten Bandbreite und der grundsätzlich unsicheren Übertragungsweise, könnte das automatische Nachfordern von nicht korrekt übertragenen Daten allerdings schnell zu einem Lockup führen. c) TCP und UDP sind beides Protokolle, die auf der Transport-Schicht des OSI-Modells angesiedelt sind. Sie transportieren die Daten in Paketen vom Rechner des Absenders zum Zielrechner. Das TCP-Protokoll nummeriert die Pakete und setzt sie beim Empfänger wieder in der richtigen Reihenfolgen zusammen. Außerdem sorgt es selbstständig dafür, dass verloren gegangene Pakete vom Absender erneut übertragen werden.	10	5	5
Aufgabe 2 (70%): Erwarteter Programmiercode für die Aufgabe siehe Dateien im Verzeichnis „ProgrammLösung“ auf der CD. (a) Benutzeroberfläche nach Vorgabe erstellen. Server auf Socket 11111 warten lassen. Bei jedem eingehenden Verbindungswunsch einen eigenen Client-Thread starten, dem der Socket übergeben wird. Input-/Output-Streams erzeugen Eingehende Kommandos vom Client in einer Schleife einlesen. Client-Thread beenden, sobald der Client die Verbindung unterbricht. Stationskennung aus erstem Datensatz ermitteln. Empfangene Datensätze ins Datenmodell der JList-Komponente schreiben. (b) Prüfsumme der empfangenen Daten berechnen. Empfangene Prüfsumme aus Datensatz extrahieren. Empfangene und berechnete Prüfsumme miteinander vergleichen. Prüfsummenfehler zählen und in JList-Komponente anzeigen. Zähler für Prüfsummenfehler zurücksetzen. (c) Reset Befehl an Client senden, wenn bei zwei aufeinanderfolgenden Datensätzen ein Prüfsummenfehler festgestellt wurde. Zähler für Prüfsummenfehler zurücksetzen.	3 3 3 2	3 6 1 4 4 5 6 5 5 5 3 5 2	5
Summe der Punkte	21	59	20
Prozentualer Anteil der Anforderungsbereiche	21%	59%	20%

Unterrichtliche Voraussetzungen

Aufgabe 1:

(a) Anspruchsvolle Aufgabe. Die Schüler haben mit Funkübertragung und Übertragungsverlusten bisher keine Erfahrung. Hier kommt es für die Schüler darauf an, das gegebene Szenario (Übertragungsfehler, stark beschränkte Übertragungsrate) in Gedanken mit den beiden angebotenen Protokollvarianten durchzugehen und zu bewerten.

(b) Zwar kennen die Schüler das TCP Protokoll – bisher aber eben als Garant für eine verlässliche (wenn auch langsamere) Kommunikation. Auch hier müssen die Schüler ihr Wissen auf für sie unbekanntem Terrain anwenden.

(c) Reproduktionsaufgabe.

Aufgabe 2:

Die Programmieraufgabe setzt gute Kenntnisse der Unterrichtsinhalte aus den letzten beiden Schuljahren voraus.

Leistungsschwächere Schüler werden Schwierigkeiten mit den komplexen Dialogelementen (JComboBox und JList) und den dazugehörigen Datenmodellen haben. Typische Probleme in diesem Zusammenhang sind, dass die Komponenten im WindowBuilder lokal im Konstruktor bzw. in einer Methode angelegt werden, aber von Hand in Attribute der Klasse umgewandelt werden müssen bzw. zumindest sichergestellt sein muss, dass sie wirksam mit den (globalen) Datenmodell-Objekten verknüpft werden.

Erfahrungsgemäß stellt es relativ viele Schüler vor Probleme, Strings zu zerlegen um an einzelne Informationen zu gelangen.

Die Prüfsummenbildung wird einige Schüler vor Schwierigkeiten stellen, weil sie es nicht gewohnt sind mit Buchstaben zu „rechnen“. Manch einer wird die eigentlich einfache Lösung deshalb nicht sehen.

Wegen der aufeinander aufbauenden Teilaufgaben, habe ich an einigen Stellen „Abkürzungen“ angeboten.