

→ verwende `BinaryTransporter.java` aus `de.htw.ds.util` um den content der data connection zwischen server & client zu transportieren.

Verteilte Systeme

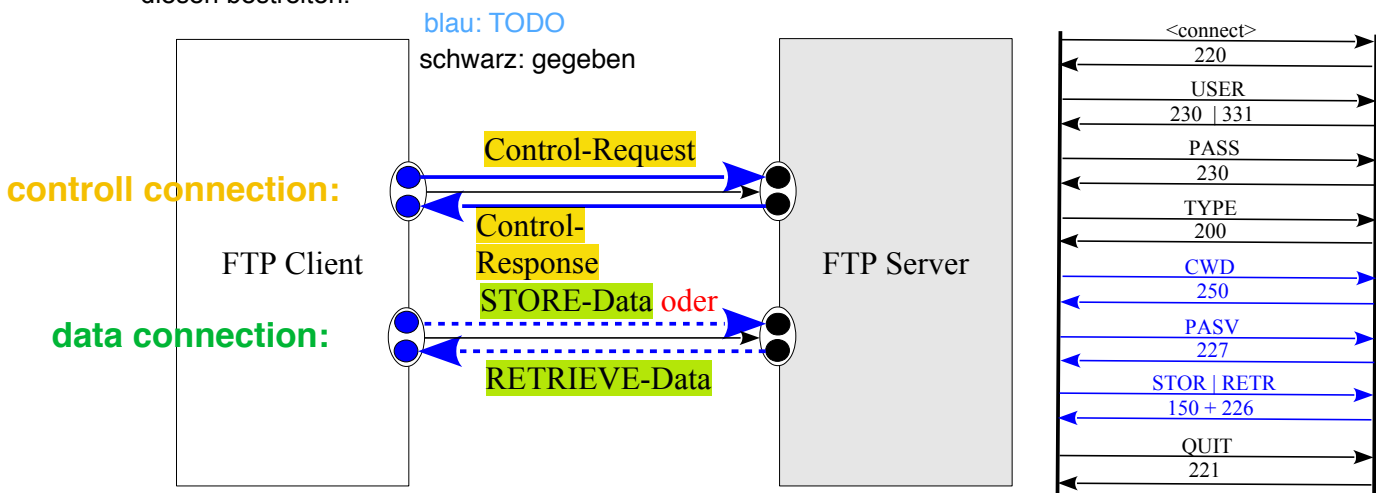
Übung B1 - Einzelarbeit

Bearbeitungszeit 2 Wochen

Die Übung adressiert den Umgang mit TCP Socket Verbindungen, Streams, und dem Java Logging API. Weiterführende Informationen zum FTP-Protokoll gibt es unter <http://en.wikipedia.org/wiki/FTP>, sowie dem im Package vorhandenen RFC 959.

Aufgabe: Automatisierter FTP Client

Ein FTP-Client ist ein Werkzeug mit dessen Hilfe sich Dateien mit einem FTP-Server austauschen lassen. Normalerweise ist ein FTP-Client ein interaktives Werkzeug bei dem ein Benutzer nach und nach die benötigten Informationen bereitstellt. Dieser einfache FTP-Client soll dagegen ähnlich wie **wget** unter Linux nicht-interaktiv, also rein durch Aufruf-Parameter parametrisiert werden, und eine gesamte FTP-Session mit diesen bestreiten.



Kopiert dazu die Klasse `FtpClientSkeleton` nach `FtpClient`. Implementiert die zwei Methoden `retrieve()` und `store()` nach den dort unter TODO hinterlegten Anweisungen. Startet Euren Ftp-Client mit den folgenden Parametern:

- Control Socket-Adresse des FTP-Servers, z.B. `uranus.f4.htw-berlin.de:21`
- User-ID
- Passwort
- Modus für Übertragung (true für binär, false für ASCII)
- Modus für Betrieb (RETRIEVE für Datei herunterladen, STORE für Datei hochladen)
- Quelldatei (absoluter Pfad für Client-Datei, relativer Pfad für Server-Datei)
- Zieldatei (absoluter Pfad für Client-Datei, relativer Pfad für Server-Datei)

Die Implementierung wurde gegen den HTW FTP-Server getestet und funktioniert mit diesem einwandfrei. Da das FTP-Protokoll ziemlich viele Freiheitsgrade bietet muss das mit anderen FTP-Servern nicht der Fall sein, also testet ihr dort auf eigene Gefahr. Beachtet dass Pfadangaben welche die FTP-Serverseite adressieren immer relativ sein müssen (da sie relativ zum Context-Verzeichnis des Servers interpretiert werden – also ohne führenden Schrägstrich bzw. ohne führende Laufwerksangabe!

FTP-Kommunikation betrifft normalerweise immer zwei Port-Verbindungen. Über eine Kontroll-Verbindung wird das eigentliche FTP-Protokoll kommuniziert. Für Datei-Transfers wird eine eigene Daten-Verbindung verwendet, über die protokollfrei Nutzdaten (die zu transferierende Datei) kommuniziert werden. Das Kontroll-Protokoll ist rein auf Text basiert: Auf eine Anfrage-Nachricht (eine Zeile) folgen eine oder mehrere Antwort-Zeilen, je nachdem ob die Antwort aus einem oder mehreren Teilen besteht, oder aus mehreren Zeilen zusammengesetzt ist. Die Klasse `FtpResponse` bietet bereits eine Methode zum Parsen der Antworten.

Der Skeleton alleine ist bereits lauffähig und behandelt alle Standard-Teile des FTP-Protokolls (Anmelden, Modus setzen, abmelden). Für die Initiierung eines Datei-Transfers muss eine Sequenz von je drei Kommandos implementiert werden. Tip: Verwendet den **TcpMonitor** aus Aufgabe B2 zur Testunterstützung.