

Aufgabe 2: Sockets (13 Punkte)

a) Ordnen Sie die folgenden Beschreibungen einer oder mehreren Socket API-Primitiven (und/oder einer Socket-API-Methode in java.net) zu:

Beschreibung	API-Primitive und/oder Java-Methode
Erstellen und Übermitteln einer Nachricht an den Kommunikationspartner (in Form eines Bytestroms)	PrintWriter getOutputStream send (Send) ✓
Blockierendes Warten und anschliessendes Annehmen eines eingehenden Connection Requests (Antrag auf Verbindungsaufbau) des Clients auf der Serverseite	wenn socket bereits offen: .accept() (Accept) ✓
Blockierendes Warten auf eine Nachricht, die vom Kommunikationspartner als Bytestrom gesendet wurde, und anschliessendes Lesen dieses Bytestroms	.accept() (Receive) BufferedReader getInputStream ✓
Beantragen eines Verbindungsaufbaus mit dem serverseitigen Kommunikationspartner (erster API-Call auf der Client-Seite)	new socket (Connect) (bind connect) ✓
Kontaktaufnahme mit dem lokalen Netzwerkinterface (also der verwendeten TCP/IP-Library) als erster Call auf der Serverseite und Bekanntgabe der Socket-Adresse (IP-Adresse und Portnummer)	new ServerSocket (listen, bind) (Socket) ✓

b) Beantworten Sie die folgenden Fragen zum Java Socket API:

1. Welchen Architekturstil realisieren TCP/IP-Sockets inhärent? Sind weitere Stile möglich?

Antwort:

Client - Server, nein es wird immer 1 socket
server geben und die clients dazu. ✓

2. Welche(s) Message Exchange Pattern(s) gibt das Java Socket API vor?

Antwort:

~~Point-to-point, publish~~
Synchron and asynchron Point-to-Point
Publish-Subscribe ?

3. Wer legt fest, wie die zu übertragenden Daten kodiert werden, a) Programmierer(in), b) TCP/IP Implementierung im Betriebssystem, c) die Java Spezifikation (JSR) des Socket APIs?

Antwort:

~~as Bytestream~~ a) ✓ Messaging Format muss
selbst spezifiziert werden (Byteströme schlussendlich)

4. Welche wichtigen Timeouts gibt es in der Socketprogrammierung?

Antwort:

Socket Timeout (Timeout bevor accept aufgerufen wird)
geler ✓