

## Chat-Server

### Engine Alpha

Die [Engine Alpha](#) (ea) ist eine Java-Bibliothek, welche eine Reihe vorprogrammierter Klassen und Methoden bereitstellt. Für unser Projekt nutzen wir die Klassen [Server](#) und [Client](#) sowie die Interface [Empfänger](#) und [SenderInterface](#).

**Empfänger bzw. SenderInterface:** Ermöglicht die Kommunikation mit allen anderen Engine-Alpha-Instanzen. Durch die zu implementierenden Methoden wird sichergestellt, dass ausgehende Nachrichten ankommen und eingehende verarbeitet werden können.

**Server:** Lauscht auf Verbindungen eines Clients und sendet an alle verbundenen Clients, d.h. er funktioniert als Broadcast. Implementiert [Empfänger](#) und [SenderInterface](#).

**Client:** Verbindet sich mit einem Server. Implementiert [Empfänger](#) und [SenderInterface](#).

### Verbindungsaufbau

Die Verbindung wird im Konstruktor der beiden Klassen [Server](#) und [Client](#) aufgebaut.

```
public Server(int port)
public Client(String ipAdresse,int port)
```

Der Server wird auf einen Port (> 1024, [da 1-1024 bereits reserviert sind](#)) festgelegt.

Der Client verbindet sich mit einem Server, indem er die IP-Adresse und den Port des Servers übergeben bekommt.

### Senden und Empfangen

Durch die Interface [Empfänger](#) und [SenderInterface](#) haben Server und Client alle möglichen Methoden zum Senden und Empfangen. Wie z.B.

```
public void sendeString(String nachricht)
public void sendeInt(int zahl)
public void empfangeString(String nachricht)
```

Diese Methoden senden an und empfangen von allen verbundenen Clients bzw. dem Server.

Wird im Client zum Beispiel die Methode [sendeString\(„test“\)](#) aufgerufen, erfolgt beim Server (nach erfolgreicher Kommunikation über das Internet – diesen Teil übernimmt die Engine) ein Aufruf der Methode [empfangeString\(„test“\)](#) mit dem entsprechenden Parameter.

## Aufgabe

Versuche nun, einen einfachen Chat-Server zu implementieren, bei dem alle Nachrichten, die dem jeder Client alle Nachrichten auf dem Bildschirm ausgibt, die an den Server gesendet werden.

### Hinweise:

- 1) Um die Klassen der Engine Alpha nutzen zu können, musst du dir den Ordner *+libs* herunterladen und in dein BlueJ-Projekt kopieren. Die Klassen der Engine Alpha sind nur verfügbar, wenn du das entsprechende Package importierst: Dazu musst du in jeden Klassen-Code `import ea.*;` in die erste Zeile deines Codes (noch vor `public class ...` !) schreiben: Der Compiler weiß somit, dass du die Klassen benutzen möchtest.
- 2) Die IP-Adresse des Servers ist die deines Computers. (die du mit „ipconfig /all“ in der Windows-Konsole herausfinden kannst: Sie steht neben „IPv4-Adresse...“). Als Abkürzung kannst du auch „[localhost](#)“ oder „[127.0.0.1](#)“ verwenden, diese Adressen führen dich auch zu deinem eigenen Computer.
- 3) Du kannst die Definitionen der Klassen der Engine Alpha jederzeit in der [Wiki](#) oder in den [docs](#) nachlesen. Außerdem gibt es in der [wiki eine extra Seite](#) über Server-Client-Kommunikation.
- 4) Falls du gar nicht weißt, wo du anfangen sollst und wie das ganze zu programmieren ist, findest du im Anhang auch noch einen kleinschrittigen Leitfaden. Du kannst diesen Leitfaden auch zum Überprüfen deiner Lösungen benutzen.

### Extra-Aufgabe für erfahrene Programmierer

- Jeder Client soll vor seine Nachricht noch (s)einen Namen schreiben.
- Mit der Engine Alpha kannst du leicht eine verbesserte Bildschirmausgabe und „Live“-Texteingabe basteln. Dazu brauchst du: die Klassen `ea.Game` und `ea.Text` sowie [die Kamera](#). In der Klasse Game gibt es die Methode `eingabeFordern(String aufforderung)`.