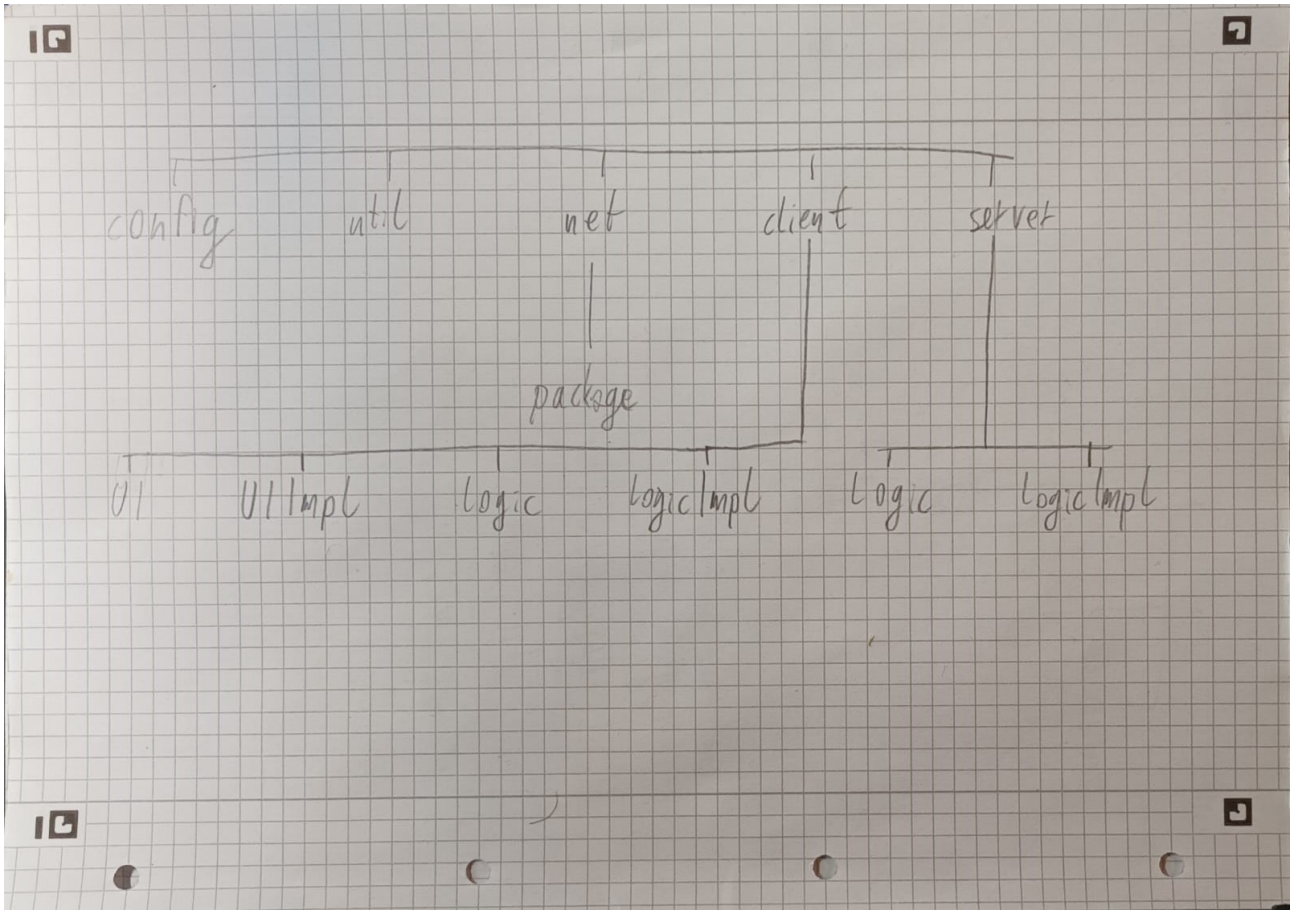
Localchat – Struktur



config

<u>Abhängigkeiten:</u>
keine
Unabhängig von:
allem
<u>Rolle:</u>
Klassen die statische Werte bereitstellen, die das verhalten in anderen Teilen der Projekts kontrollieren (Portnummern, Maximalwerte, usw.)

util

<u>Abhängigkeiten:</u>
config
<u>Unabhängig von:</u>
net, packet, client, server
<u>Rolle:</u>
Klassen und Funktionen die in allen Teilen des Projekts von nutzen sein können.

Keine Netzwerkfunktionen. Keine UI Funktionen. Keine Spezialisierungen für einzelne Anwendungsfälle.

net

Abhängigkeiten:
config, util
Unabhängig von:
packet, client, server
Rolle:
Klassen und Funktionen zur Übertragung von Daten über Netzwerke. Klassen die wiederkehrende Aufgaben vereinfachen.

Keine UI Funktionen. Keine Spezialisierungen für einzelne Anwendungsfälle.

packet

Abhängigkeiten:
config, util, net
Unabhängig von:
client, server
Rolle:
Paketklassen, die das serialisieren, de-serialisieren, validieren und normalisieren von Daten koordinieren die zwischen dem Client und Server ausgetauscht werden. Alle Pakettypen, auch solche, die für spezifische Anwendungsfälle ausgelegt sind, sollen hier definiert und implementiert werden.

Keine UI Funktionen.

client

Abhängigkeiten:
config, util, net, packet
Unabhängig von:
server
Rolle:
Enthält untergeordnete Pakete, die alle Funktionalitäten definieren und implementieren die mit denen Benutzer interagieren und für ihn mit dem Server kommunizieren.

Kein Code außerhalb der Unterpakete.
Die Client Anwendung ist vollständig unabhängig vom Server; selbst wenn der Client der Host ist. Die einzige Verbindung findet über das Netzwerk statt (eventuell localhost). Obwohl der Client sich nicht darauf verlassen darf, ist es nicht ausgeschlossen, dass der Server Teil des selben Prozesses wie der Client ist.

client.UI

Abhängigkeiten:
config, util
Unabhängig von:
net, packet, server, client.UIImpl, client.logic, client.logicImpl
Inhalt:
UI (Abstrakte Klasse): Methoden zum übergeben von Informationen an die UI

Rolle:
Enthält ein Interface, dass Methoden für alle Events enthält, die an die UI gesendet werden können.

Erklärung:
# (verwendet Dependency Injection)
ui = ClientUIImpl()
logic = ClientLogicImpl()
ui.set_logic(logic)
logic.set_ui(ui)
ui.start() # startet intern Logik

Nun kann die UI über das Client-Logic-Interface die Methoden der Implementierung dieser nutzen, ohne die Implementierung tatsächlich zu kennen, und die Client-Logik kann Informationen an die UI senden ohne ihre Implementierung zu kennen.

Kommunikation zwischen UI und Logik
UI → sende Nachricht: „hi" → Logic
Logic → Nachricht empfangen: „hallo" → UI

Aufgaben innerhalb der Implementierungen
Logic: Keep-Alive-Signal an Server senden (UI ist nicht involviert)
UI: Nächsten Frame in GUI rendern (Logik ist nicht involviert)

client.UIImpl
Abhängigkeiten:
config, util, client.UI, client.logic
Unabhängig von:
net, packet, server, client.logicImpl
Rolle:
Enthält alle Implementierungen des Client-UI-Interfaces.

client.logic
Abhängigkeiten:
config, util, net, packet
Unabhängig von:
server, client.UI, client.UIImpl, client.logicImpl
Inhalt:
logic (Abstrakte Klasse): Methoden zum senden von Befehlen und Abfragen an die Client-Logik
Rolle:
Enthält ein Interface, dass Methoden für alle Befehle und Abfragen enthält, die an die Client-Logik gesendet werden können.

client.logicImpl

Abhängigkeiten:
config, util, net, packet, client.logic
Unabhängig von:
server, client.UIImpl
Rolle:
Enthält alle Implementierungen des Client-Logik-Interfaces.

server

Abhängigkeiten:
config, util, net, packet
Unabhängig von:
clien
Rolle:
Enthält untergeordnete Pakete, die alle Funktionalitäten implementieren, die der Server hat.

Kein Code außerhalb der Unterpakete.
Die Server Anwendung ist vollständig unabhängig von allen Clients, selbst vom Host. Verbindungen finden ausschließlich das Netzwerk statt (localhost eingeschlossen). Obwohl der Server sich nicht darauf verlassen darf, ist es nicht ausgeschlossen, dass der Host-Client Teil des selben Prozesses wie der Server ist.

server.logic

Abhängigkeiten:
config, util, net, packet
Unabhängig von:
client, server.logicImpl
Inhalt:
logic (Abstrakte Klasse): Methoden zum starten und Stoppen des Servers, und zum festlegen des Host-Client-Ports (localhost)
Rolle:
Enthält ein Interface für die Steuerung des Servers.

server.logicImpl

Abhängigkeiten:
config, util, net, packet, server.logic
Unabhängig von:
client
Rolle:
Enthält alle Implementierungen des Server-Logik-Interfaces.