

«java.io.PrintWriter»
PrintWriter

«java.io.InputStreamReader»
InputStreamReader

«java.io.BufferedReader»
BufferedReader

«java.io.OutputStreamReader»
OutputStreamReader

«java.io.Socket»
Socket

«Client»
ChattyChatChatClient

-portNumber: int
-serverAddress: String
-message: String
-nickname: String
-output: OutputStreamReader
-input: InputStreamReader
-clientHandler<ClientHandler>: ArrayList

+main(args: String[]): int
+ChattyChatChatClient(host: String)
+startClient(): void
-connectServer(): void
-setupStreams(): void
-ableToType(tof: boolean): void
-sendMessageAll(message: String): void
-sendMessagePrivate(message: String, name: String): void
+showMessage(text: String): void
-setName(name: String): void
+getName(): String
+endClient(): void

«Server»
ChattyChatChatServer

-portNumber: int
-output: OutputStreamReader
-input: InputStreamReader
-server: ServerSocket
-connection: Socket
-nickname: String
-clientHandler<ClientHandler>: ArrayList

+ChattyChatChatServer(listen: ServerSocket)
+getConnections(): void
+addHandler(handler: ClientHandler): void
+removeHandler(handler: ClientHandler): void
+startServer(port: int): void
-connectServer(): void
-setupStreams(): void
+sendMessageAll(from: ClientHandler, message: String): void
+sendMessagePrivate(from: ClientHandler, message: String, receiver: String): void
-ableToType(tof: boolean): void
+main(args: String[]):int
+endServer():void
-setName(name: String): void
+getName(): String

«Handler»
ClientHandler

-sock: Socket
-server: ChattyChatChatServer
-out: PrintWriter

+ClientHandler(sock: Socket, server: ChattyChatChatServer)
+run(): void

«java.lang.Thread»

