

```

# GUI_ChatServer_Multi.py
# threading 모듈을 이용한 TCP 멀티 채팅 서버 프로그램

from socket import *
from threading import *

class MultiChatServer:
    #소켓을 생성하고 연결되면 accept_client() 호출
    def __init__(self):
        #접속된 클라이언트 소켓 목록
        self.clients = []
        #최종 수신 메시지
        self.final_received_message = ""

        #소켓을 생성하고 연결 요청 수락
        self.s_sock = socket(AF_INET, SOCK_STREAM)
        self.ip = "
        self.port = 2500
        self.s_sock.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
        self.s_sock.bind((self.ip, self.port))
        print("클라이언트 대기 중 ...")
        self.s_sock.listen(100)
        self.accept_client()

        #클라이언트 소켓을 목록에 추가하고 스레드를 생성하여 데이터를 수신한다
    def accept_client(self):
        while True:
            #연결 수락
            client = c_socket, (ip, port) = self.s_sock.accept()

            #소켓 목록에 없으면 접속된 소켓을 목록에 추가
            if client not in self.clients:
                self.clients.append(client)
                print (ip , ':' , str(port), ' 가 연결되었습니다')

            #접속 소켓을 위한 스레드 실행
            cth = Thread(target=self.receive_messages, args=(c_socket,)) #수신 스레드

            cth.start() #스레드 시작

        #데이터를 수신하여 모든 클라이언트에게 전송한다

```

```

def receive_messages(self, c_socket):
    while True:
        try:
            incoming_message = c_socket.recv(256)
            if not incoming_message: #연결이 종료됨
                break
        except:
            continue
        else:
            self.final_received_message = incoming_message.decode('utf-8')
            print(self.final_received_message)
            #발신자를 제외한 클라이언트에게 전송
            self.send_all_clients(c_socket)
    c_socket.close()

#송신 클라이언트를 제외한 모든 클라이언트에게 메시지 전송
def send_all_clients(self, senders_socket):
    for client in self.clients: #목록에 있는 클라이언트
        socket, (ip, port) = client

        #송신 클라이언트는 제외하고 전송
        if socket is not senders_socket:
            try:
                socket.sendall(self.final_received_message.encode('utf-8'))

            #전송 실패면 연결이 종료된 것이므로 목록에서 제거
            except:
                self.clients.remove(client) #소켓 제거
                print("{} , {} 연결이 종료되었습니다".format(ip, port))

if __name__ == "__main__":
    MultiChatServer()

```