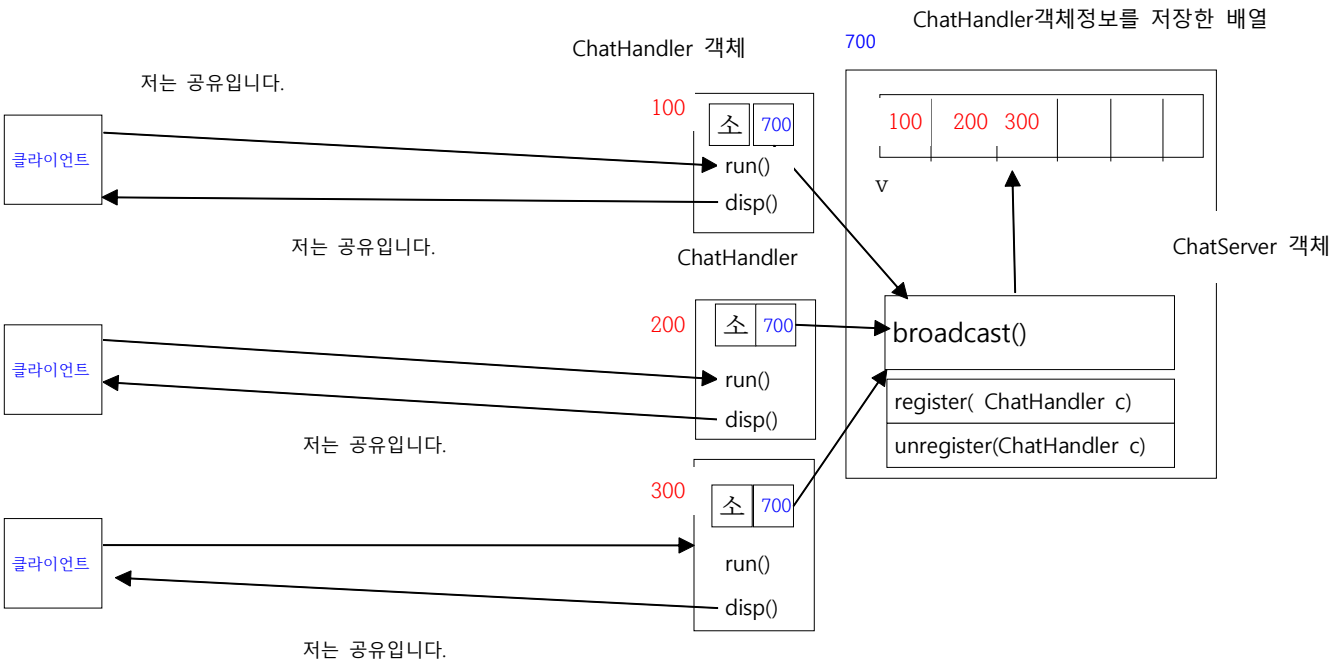
```
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Vector;

public class ChatServer {
	Vector<ChatHandler> v;

	public ChatServer( ) {
		try {
			ServerSocket ss = new ServerSocket(5000);
			v= new Vector<>();
			System.out.println("서버 준비 완료");
			while(true) {
				Socket s = ss.accept();
				ChatHandler c= new ChatHandler(this, s);
				System.out.println("클라이언트");
				c.start();
			}

		}catch(Exception e) {
			e.printStackTrace();
		}
	}

	public void register(ChatHandler c) {
		v.add(c);
	}

	public void unregister(ChatHandler c) {
		v.remove(c);
	}

	public synchronized  void broadcast(String message) {

			for(int i=0; i<   v.size(); i++) {
				ChatHandler c = v.get(i);
				try {
					c.disp(message);
				}catch(Exception e) {
					e.printStackTrace();
				}
			}
	}


	public static void main(String[] args) {
		new ChatServer( );
	}
}
```

```java
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;



public class ChatHandler  extends Thread{

	Socket s;
	ChatServer cs;
	DataInputStream in;
	DataOutputStream out;


	public ChatHandler(ChatServer cs, Socket s) throws IOException {
		this.s = s;
		this.cs =cs;
		in= new DataInputStream(s.getInputStream());
		out = new DataOutputStream(s.getOutputStream());
	}

	public void run() {
		String name="";
		try {
			name=in.readUTF();
			cs.register(this);    //생성된 객체를(chatHadler 객체)서버의 배열(벡터)에 담기
			cs.broadcast(name + "님이 방문하셨습니다");

			while(true) {
				String msg =in.readUTF();
				cs.broadcast(name + "님의 말:" + msg);
			}

		}catch(Exception e) {
			System.out.println("나감");
		}

		cs.unregister(this); //생성된 객체를(chatHadler 객체)서버의 배열(벡터)에서 제거하기
		cs.broadcast(name + "님이 나가셨습니다");

		//쓰레드 종료전에 모든 스트림 종료
		try {
			in.close();
			out.close();
			s.close();
		}catch(IOException e) {
			//e.printStackTrace();
		}

	}

	public void disp(String message) throws IOException {
		out.writeUTF(message);
	}
}
```


ChatHandler객체정보를 저장한 배열
700
ChatHandler 객체
저는 공유입니다.
100
소
700
run()
disp()
클라이언트
100 200 300
v
저는 공유입니다.
ChatHandler
ChatServer 객체
200
소
700
broadcast()
run()
disp()
클라이언트
register(ChatHandler c)
unregister(ChatHandler c)
저는 공유입니다.
300
소
700
run()
disp()
클라이언트
저는 공유입니다.